TAXONOMY OF SUSTAINABILITY INDICATORS FOR SOFTWARE DEVELOPMENT

By

MUHAMMAD USMAN SALEEM

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

March, 2023

TAXONOMY OF SUSTAINABILITY INDICATORS FOR SOFTWARE DEVELOPMENT

By

MUHAMMAD USMAN SALEEM

BSSE, National University of Modern Languages, Islamabad

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

IN **Software Engineering**

To

FACULTY OF ENGINEERING & COMPUTER SCIENCES

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

March, 2023

NATIONAL UNIUVERSITY OF MODERN LANGUAGES FACULTY OF ENGINEERIING & COMPUTER SCIENCE

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defence, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

Thesis Title: **Taxonomy of Sustainability Indicators for Software Development**

Submitted By: Muhammad Usman Saleem **Registration #:** NUML-F19-20892

Master of Science in Software Engineering

Software Engineering
Name of Discipline

Dr. Raheel Zafar
Name of Research Supervisor

Signature of Research Supervisor

Dr. Basit Shahzad
Name of Dean (FE&CS)

Signature of Dean (FE&CS)

Brig Syed Nadir Ali
Name of Director General

Signature of Director General

March 15th , 2023

Date

AUTHOR'S DECLARATION

I Muhammad Usman Saleem

Son of Muhammad Saleem Akhtar

Registration # NUML-F19-20892

Discipline Software Engineering

Candidate of **Master of Science in Software Engineering (MSSE)** at the National University of Modern Languages do hereby declare that the thesis **Taxonomy of Sustainability Indicators for Software Development** submitted by me in partial fulfillment of MSSE degree, is my original work, and has not been submitted or published earlier. I also solemnly declare that it shall not, in future, be submitted by me for obtaining any other degreefrom this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be cancelled and the degree revoked.

Signature of Candidate

Muhammad Usman Saleem

Name of Candidate

Date

ABSTRACT

The rise in environmental and social issues has led to increase the awareness of the sustainability of human activities. The software development industry is no exception, and there has been a recent surge in the number of studies on sustainable software development (SSD). However, still, there is not enough understanding of what constitutes SSD and how it can be achieved. The problem being addressed in this study is the lack of a comprehensive taxonomy of sustainability indicators for software development. Systematic literature review was performed to identify existing SSD indicators. The results were analyzed and classified into various dimensions of sustainability. The taxonomy can be used by the practitioners to select appropriate indicators for sustainable development of their software projects. A quantitative approach is adopted by the researcher as the following research design aids in collecting primary data that has the ability to afford results with high precisions along with appropriate statistics. The method of survey questionnaire has been used in the current research in which the information has been collected from 80 respondents. The focus of the present research is on the primary quantitative method. Therefore, the data has been analysed using Statisical Package for Social Sciences (SPSS) software through frequency analysis, regression and correlation analysis. It has been found that the heightened awareness of the degree to which human activities can be maintained has come about as a direct consequence of the rise of environmental and social concerns. The industry of software development is not an exception, and there has recently been an increase in the number of studies that focus on environmentally responsible practices in the software development process. It is recommended that significant value and consideration should be given to the paradigm of green software system as it has mainly assist the corporations to develop eco-friendly system, which is cost-effective as well.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SDG-Sustainable Development Goals

OSSMM- Open Source Software Sustainability Maturity Model

TBL- Triple Bottom Line

EPA- Environmental protection agency

EIA- Environmental Impact Assessment

LCA-life cycle assessment

IOA-input-output analysis

ACKNOWLEDGMENT

First of all, I wish to express my gratitude and deep appreciation to Almighty Allah, who made this study possible and successful. This study would not be accomplished unless the honest espousal that was extended from several sources for which I would like to express my sincere thankfulness and gratitude. Yet, there were significant contributors for my attained success and I cannot forget their input, especially my research supervisors, Asst. Prof. Dr. Raheel Zafar and my Co-Supervisor Prof. Dr. Noman Ahmad, who did not leave any stone unturned to guide me during my research journey.

I shall also acknowledge the extended assistance from the administrations of Department of Software Engineering who supported me all through my research experience and simplified the challenges I faced. For all whom I did not mention but I shall not neglect their significant contribution, thanks for everything.

DEDICATION

This thesis work is dedicated to my parents and my teachers throughout my education career who have not only loved me unconditionally but whose good examples have taught me to work hard for the things that I aspire to achieve. I do also want to dedicate this thesis to my wife and my son from whom I always got motivated.

CHAPTER 1

INTRODUCTION

1.1 Overview

Sustainability is increasingly used in the software development community to describe various environmental, social, and economic objectives. This use of the term has led to the development of a range of different sustainability indicators for software development. Moreover, sustainability has received increased attention in recent years, both in academia and the industry [1]. The definition of sustainability is still a matter of debate. Still, there is a consensus that it involves meeting the needs of the present generation and preserving the needs of future generations [6]. The software development industry is not immune to the challenges of sustainability. The industry has been criticised for its high rates of software development failures, which can be detrimental to the environment and society. Despite the importance of sustainability in software development, there is no consensus on what indicators should be used to measure it [2]. This lack of consensus makes it difficult to compare the sustainability performance of different software development projects and organisations. However, a comprehensive taxonomy of sustainability indicators for software development would be a valuable resource for practitioners and academics. Such a taxonomy would provide a common language for discussing sustainability in software development and allow for comparing and contrasting different indicators [3]. However, this study proposes a taxonomy of sustainability indicators for software development. The taxonomy is based on the triple bottom line approach. The taxonomy is organised into three main categories: economic indicators, environmental indicators, and social indicators [4]. Each category is further divided into subcategories. Software development practitioners can use the taxonomy to select appropriate indicators for their projects and organisations. Researchers can also use it to identify gaps in the existing body of knowledge and guide future research in this area [2]. Thus, the following research study is aimed to contribute to existing knowledge in the area of sustainability indicators for software development.

1.2 Research Background

In the contemporary digital domain, the Internet of Things, cloud computing, machine learning, and other smart computing technologies are quickly increasing. One of the major concerns is energy conservation. According to [7], the majority of research relating to computers and energy efficiency has been on lower levels of software and hardware technology stacks. These lower-level techniques, however, do not capture the complete picture in terms of energy use. It appears that rather than using energy directly, software systems influence how hardware is used, resulting in indirect energy consumption. The authors of this research also investigated the existing tools and approaches for developing energy-efficient software systems. They suggest that a lack of tools and expertise now prevents developers from designing energy-efficient software systems. According to [8] the programme is viewed as a single entity, which fails to provide precise insight into energy consumption behaviour. The researchers suggest an energy consumption viewpoint on software architecture in the study to allow for the investigation of architectural features as the primary drivers of energy consumption. The potential of the energy-saving technique is 67% as a consequence of using commercial software and software quality characteristic sustainability [9]. In this regard, it has become critical to investigate the domain of energy consumption in the modern software development process, which is characterised by leading-edge technologies such as AI, Big Data, and voluminous data.

In recent years, both academics and industry have paid considerable attention to the notion of sustainability. The phrase "sustainability" refers to a system's, processes, or resources' ability to perform at a specific level throughout time. Sustainability in software development has been widely defined to encompass elements such as a project's long-term viability, the capacity to maintain a given degree of quality, or the flexibility to adapt to changing needs [5]. In the literature, there is no single agreed-upon definition of sustainability. As a result, there is no standard set of metrics for measuring sustainability. Because there is no universal notion of sustainability, it is impossible to compare and contrast the findings of various research and discover patterns.

The concept of sustainability has received increased attention in recent years, both in academia and the industry. The term "sustainability" is typically the ability of a system, process, or resource to continue functioning at a certain level over time. In software development, sustainability has been variously defined to include factors such as the long-term viability of a project, the ability to maintain a certain level of quality, or the ability to adapt to changing requirements [5]. There is no single agreed-upon definition of sustainability in the literature. As a result, no standard set of sustainability indicators can be used to measure it. This lack of a shared understanding of sustainability makes it difficult to compare and contrast the results of different studies and identify trends and best practices.

The taxonomy of the indicators is essential for developing an understanding of sustainability in software development and for providing a shared language that can be used by researchers and practitioners alike. Various researchers have proposed different models and frameworks for classifying sustainability indicators, but there is no consensus on which one is the most appropriate [6]. There are different indicator classification schemes, each with its advantages and disadvantages. These are described in more detail below.

A taxonomy is a classification scheme that organises elements into categories based on shared characteristics. The purpose of a taxonomy is to provide a framework for understanding the relationships between different elements. In sustainability indicators, a taxonomy can be used to classify indicators according to their purpose, type, or other criteria. A sustainability indicator taxonomy helps to evaluate different indicators proposed in the literature and identify relationships between them [7]. It can also be used to develop new indicators or select existing ones for use in a particular context. For example, the analysis of current sustainability strategies, such as application framework, sustainability metrics, resource planning, resource management, configuration management, thermoelectric planning, cooling maintenance, sustainable sources, and heat recovery allocation for CDCs, is done using a taxonomy of sustainable cloud technology [7].

During discussions about the sustainability of software, the objective is to develop and apply efficient algorithmic solutions to minimise software's direct carbon footprint (for example, electricity consumption from CPU cycles) as well as its indirect effects on sustainability (such as the effects based on how the system is used in the particular operational context) [10]. Even while software consumes no energy, it has a significant influence on how much physical equipment consumes. Software, therefore, indirectly contributes to energy usage. The fundamental concern, in this case, is that the methods used to design, manage, operate, and maintain software may have an environmental impact. Because a process affects the organisational style of operation in a certain region, different processes are answerable for different environmental impacts. In this sense, a sustainable software process is one that accomplishes its (realistic) sustainability goals, which are articulated in terms of how its conception and execution directly and indirectly effect the economy, society, people, and environment [10]. The method in which the many activities that comprise the software process as a whole are carried out may have a substantial influence on the lifetime of the software process. A similar technique for software longevity was recently proposed in

A taxonomy is a classification scheme that organises elements into categories based on shared characteristics. The purpose of a taxonomy is to provide a framework for understanding the relationships between different elements. In sustainability indicators, a taxonomy can be used to classify indicators according to their purpose, type, or other criteria. A sustainability indicator taxonomy helps to evaluate different indicators proposed in the literature and identify relationships between them [7]. It can also be used to develop new indicators or select existing ones for use in a particular context. The most often used approach for measuring social sustainability is the life cycle assessment (LCA). This is a "cradle-to-grave" strategy to assessing a product's inputs, outputs, and environmental impacts across its whole life cycle [11]. Examples of this include municipal trash management, land usage, and environmental emissions [12]. The LCA has been updated to cover social problems such as labour force participation, community living standards, cultural heritage, freedom, safety, equity, and poverty alleviation [12]. Meanwhile, [22] employed a Social Impact Indicator (SII). SII is used to calculate social consequences like as stakeholder engagement and human resources based on LCA [13]. When the LCA and the Economic Input and Output Analysis approach (EIO) are merged in, an economic input-output-based life cycle assessment (EIO-LCA) is formed [14]. The EIO-LCA was used to assess the immediate and indirect supportability effects of US development firms (for example, indirect injuries during work).

Durmanov et al. (2019) used vulnerability assessment methodologies to examine the social impacts of urban redevelopment programmes (VATs). This was accomplished by identifying the most susceptible individuals and evaluating their negative social repercussions [23]. This technique can tell policymakers a lot about how to reduce the negative social impact of the project [23]. [15] developed a Social Network Analysis (SNA)-based methodology for assessing the social performance of infrastructure projects. The SNA was used to identify project stakeholders as participants, as well as their level of influence (relationships between actors) and unique societal needs.

Several different approaches can be used to develop a taxonomy of sustainability indicators. One approach is to use an existing classification scheme, such as the United Nations' Sustainable Development Goals (SDGs). SDGs guide global efforts to achieve sustainable development [17]. Each goal has a corresponding set of targets and indicators. The indicators can be used to measure progress towards the goals.

Another approach is to develop a taxonomy based on existing frameworks and models for sustainability in software development. One such framework is the Open-Source Software Sustainability Maturity Model (OSSMM), developed by a team of researchers at the University of Innsbruck. The OSSMM is a framework for assessing the sustainability of open-source software projects [18]. It consists of indicators organised into five dimensions: project governance, development process, code quality, community engagement, and user satisfaction. A third approach is to develop a taxonomy from the literature review. This approach has the advantage of being able to take into account the full range of sustainability indicators that have been proposed in the literature. However, it has the disadvantage of being time-consuming and difficult to update.

The taxonomy presented is based on a review of the literature. It includes 100 indicators, organised into 11 categories: project management, software development process, code quality, documentation, user satisfaction, community engagement, business model, licensing, infrastructure, and governance [19]. The taxonomy is intended to be a living document that can be updated as new indicators are proposed in the literature.

1.3 Research Problem

Although the literature has become voluminous on various methods to achieve sustainability in software development, a lack of industry standards points towards the inefficacy of these novel methods for real-time application. Meanwhile, the software development industry continuous to exploit natural and social resources in an unsustainable manner. The situation has been further aggravated in recent years due to an exponential expansion of the global software market [20]. Meanwhile, a significant proportion of this market has also shifted to low- and middle-income countries, where substandard labour practices, lack of standardisation, and data theft and breaches jeopardise the entire concept of sustainability in this industry. Moreover, apart from a few studies such as [8], most of the recent literature fails to answer the question of how the ever-increasing burden of energy consumption could be reduced through sustainable software development. In this regard, research has shown some progress over the last four years as the concept of ‘green IT’ expanded to include the energy component of technological/digital development. However, a significant gap exists in the knowledge of the software component of this domain, because most of the scientific investigation is focused on the hardware component. For this, an augmented concept of ‘green software development’ has emerged. Environmentally friendly software development has received little attention in the past. One such endeavour is Green Project Management [21], which focuses on environmental issues of project management. Individual parts of the software development lifecycle, on the other hand, might benefit from environmentally friendly solutions. On the other hand, much study has gone into evaluating many elements like as power usage by computers, displays etc. [22]. Moreover, the fundamental idea behind ‘sustainability’ is the prudent usage of not only energy (environmental part) but the impact on society and economy as well (the triple bottom line of sustainability). Even though the current literature discusses the energy (environmental) component of software development, analysis of the societal impact and economic implications are nearly absent [8]. Hence, a clear understanding of relevant sustainability dimensions is required to guide the current wave of sustainability in software development.

A third challenge is the lack of tools and methods for measuring sustainability in software development. Many existing tools and methods are designed for other industries and do not take into account the unique characteristics of software development [10]. This makes it difficult to accurately assess the sustainability of software development processes. The aim of this research is to address these challenges by developing a taxonomy of sustainability indicators for software development. The taxonomy will provide a shared understanding of what constitutes a sustainable software development process. It will also provide a way to compare and contrast different approaches to sustainability. Finally, the taxonomy will provide a tool for selecting appropriate indicators for measuring sustainability in software development.

1.4 Research Rationale

The problem of developing sustainable software is becoming increasingly relevant as the industry grows. There are a number of reasons for this. First, the software development industry is growing rapidly, and with this growth comes an increased demand for sustainability. Secondly, sustainability is becoming increasingly important to the general public, and software development is no exception [11]. Thirdly, how software is developed is changing, and sustainability is becoming more critical. Finally, there are a number of challenges associated with sustainable software development, and it is essential to understand these challenges to address them effectively [12]. This study will address the problem of developing sustainable software by providing a taxonomy of sustainability indicators for software development. This taxonomy will provide a framework for developers to identify, select, and use indicators to assess the sustainability of their software development processes. In doing so, it is hoped that this study will play a significant part in developing more sustainable software development practices.

1.5 Research Aim and Objectives

The aim of this study is to develop a taxonomy of sustainability indicators for software development. The specific objectives of the study are as follows:

1. To identify the aspects of software sustainability considerations
2. To decipher the various factors determining the sustainability of software
3. To analyse the domain of energy consumption in the software development process
4. To develop a taxonomy of sustainability indicators for software development.

1.6 Research Questions

The following are the research questions:

1. What are the current considerations regarding software sustainability?
2. What are the factors that determine the final sustainability of software?
3. How does a focus on lowering energy consumption affect software development?
4. How does a focus on dimensions of sustainability, as a whole, impact the process of software development?
5. What are the main sustainability indicators for software development?

1.7 Research Significance

The taxonomy of sustainability indicators for software development is significant because it provides a way to measure and compare the sustainability of different software development processes. The taxonomy can be used to identify best practices and make recommendations for improving the sustainability of software development. That is why this research study is highly significant in several ways. Firstly, the taxonomy can help organisations understand how sustainable their software development processes are. Secondly, the taxonomy can help organisations to improve the sustainability of their software development processes. Thirdly, the taxonomy can help organisations compare the sustainability of different software development processes [13]. Finally, this research study is also significant for policymakers because it can help them to develop policies that encourage the sustainable development of software.

1.8 Scope of the Research Work

The scope of this research work is limited to the development of a taxonomy for sustainable software development. The taxonomy will be developed using a systematic literature review of the existing body of knowledge on sustainability in software development. The taxonomy will not be used to assess the sustainability of specific software development processes or practices.

1.9 Thesis Structure

Chapter 1 is based on Introduction. It introduces the topic of sustainable software development and summarises the research problem. Additionally, it defines the scope of the research work and outlines the research aims and objectives. Finally, this chapter presents an overview of the dissertation structure. Chapter 2 is based on Literature Review. It presents a literature review of the existing knowledge on sustainability in software development. The literature review is organised around different themes. Chapter 3 is based on Methodology. The research methodology of the study is discussed in this chapter. The research methodology includes a systematic literature review. The systematic literature review is used to develop the taxonomy of sustainability indicators for software development. Chapter 4 is based on Results Analysis. It presents the results of the systematic literature review. The results from the systematic literature review are discussed and analysed. Chapter 5 is based on Discussion of Results. It presents the critical discussion of the findings in the light of literature review. It is the basis for the conclusion chapter. Chapter 6 is based on Conclusion. It presents conclusions at the end. The conclusions are drawn from a systematic literature review. Recommendations for future research are also presented in this chapter. Moreover, references and appendices have been given at the end of the dissertation.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter overviews the literature review of the existing knowledge body on software development sustainability. This includes a review of the most commonly used definitions and frameworks for sustainability indicators in software development. In addition, it also provides an analysis of the methods used to measure and assess sustainability indicators. Moreover, the literature from various researchers on the application of sustainability indicators in software development projects is reviewed and analysed. At the end of the chapter, a literature gap has been identified based on the extensive review of existing studies on the subject topic identifying various areas where further research is required by future researchers.

Moreover, this chapter has also analysed the top ten venue studies of 2022 on the subject of sustainability indicators for software development. These studies include a broad range of topics such as the application of lean and agile software development methods for sustainable software development, assessment of sustainability indicators, and the use of sustainability indicators for performance measurement. These studies that has been referenced (but not limited to) in the chapter are as follows:

1. Olteanu, Y. and Fichter, K., 2022. Startups as sustainability transformers: A new empirically derived taxonomy and its policy implications. *Business Strategy and the Environment*.
2. Poth, A. and Nunweiler, E., 2022, January. Develop Sustainable Software with a Lean ISO 14001 Setup Facilitated by the efiS® Framework. In *International Conference on Lean and Agile Software Development* (pp. 96-115). Springer, Cham.
3. Beerbaum, D.O., 2022. Development of a Sustainability Taxonomy for investor decision usefulness–Reflection of SEC and ISSB climate-related disclosures. *Available at SSRN*.

4. Verdejo, Á., Espinilla, M., López, J.L. and Melguizo, F.J., 2022. Assessment of sustainable development objectives in Smart Labs: technology and sustainability at the service of society. *Sustainable Cities and Society*, *77*, p.103559.

5. Bharany, S., Badotra, S., Sharma, S., Rani, S., Alazab, M., Jhaveri, R.H. and Gadekallu, T.R., 2022. Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy. *Sustainable Energy Technologies and Assessments*, *53*, p.102613.

6. Zada, I., Shahzad, S., Ali, S. and Mehmood, R.M., 2022. OntoSuSD: Software engineering approaches integration ontology for sustainable software development. *Software: Practice and Experience*.

7. Sharifi, A. and Allam, Z., 2022. On the taxonomy of smart city indicators and their alignment with sustainability and resilience. *Environment and Planning B: Urban Analytics and City Science*, *49*(5), pp.1536-1555.

8. Gill, S.S., Kumar, A., Singh, H., Singh, M., Kaur, K., Usman, M. and Buyya, R., 2022. Quantum computing: A taxonomy, systematic review and future directions. *Software: Practice and Experience*, *52*(1), pp.66-114.

9. Park, S.M. and Kim, Y.G., 2022. A Metaverse: Taxonomy, components, applications, and open challenges. *Ieee Access*, *10*, pp.4209-425

10. Hamdi, M., 2022. Towards a classification of sustainable software development process using manifold machine learning techniques. *Journal of Intelligent & Fuzzy Systems*, (Preprint), pp.1-12.

2.2 Taxonomy and Sustainability in Software Development

Before discussing the concept of taxonomy indicators for software development, it is important to understand the application of sustainability in software development. Sustainability is about finding a balance between environmental, social, and economic factors for managing the current human needs without compromising resources for future generations [14]. The concept of sustainability indicators has been developed and provides a means of measuring progress toward sustainable development (SD). There are a number of different frameworks that have been proposed for achieving sustainability in software development. The most commonly used framework is the triple bottom line (TBL) framework [15]. The environmental dimension of sustainability in software development refers to the impact of software development activities on the environment. This includes the impact of software development on climate change, energy use, water use, and waste generation. While the social dimension of sustainability in software development focuses on the impact of software development activities on society, this includes the impact of software development on social cohesion, social inclusion, and human rights.

Moreover, the economic dimension of sustainability in software development refers to the impact of software development activities on the economy [16]. This includes the impact of software development on economic growth, employment, and poverty alleviation. It is important to note that businesses are focusing on the development of green software to enhance the sustainability and for that reason, there are multiple taxonomy indictors needed for sustainable or green software development [14]. The taxonomy indicators for sustainable software development require the focus of the entire software engineering domain on certain standards, practices and guides that are necessary for software development. Taxonomy in software development is basically the planning of developing software in such a way that it enables software to meet the requirement and standards [74]. For instance, the development of green or eco-software' taxonomy indicators not only provides a complete set of practices for translating this idea into practical shape but also draw attention towards making software capable of meeting requirements.

2.2.1 Machine learning in sustainable development

The current classification and choice of the ten articles have considered the need for machine learning as an essential consideration in the sustainable development process. It has added value to the role of sustainable development with the claim of the machine learning process as an aid and support for the sustainable development process efficiency. The most commonly used framework for sustainability functionality with the help of equal use of machine efficiency in the progression with machine learning [15]. The machine support may boost the level of working condition and also adds more value to the interest of the work plans. This is an initiative that may work in the tools motivation and supporting the need for a sustainable balance. The ten articles choose for the current study hence show the active use of machine-based knowledge and the reference to the need of the machine as an active tool for gaining of knowledge in the right manner. It is evaluated that more use of sustainable development indicators has been employed in a chosen article [12]. The second article is about ISO support for setup facilitation in sustainable development, which may apply the need for the use of machine learning [13]. Third article is about the sustainability taxonomy for investor decision support in machine learning and sustainable growth [14]. Fourth article is about the machine with the help of machine knowledge sustainable development objectives in Smart Labs, which shows a direct impact of machine learning [15]. Fifth article is about the energy efficiency and green support with the cloud computing which is not possible without machine learning [16]. Sixth article is about the Software engineering approaches with machine aid [17]. The seventh article has employed the use of division for the smart city indicators and their alignment with sustainability with machine support [18]. Eighth article is about Quantum computing which is possible with the help of machine language use and learning [19]. Ninth article is about the components, applications, and it is under the support of the machine language [20]. Last article has based on the use of the sustainable software development process using machine learning techniques that is possible with the machine learning only [21].

2.3 How sustainable software development started

The study for sustainable and green development was introduced in 1987 for the first time. It was presented at the platform of the World Convention Environment and Development. At this platform, it was defined that the green software processes are those that meet the demands of the present as they are designed in such a way. Also, they do not compromise the resources of the future generations required to meet their demands. After this, the concept heralded a new era for software organisations [18]. Initially, the concept of green software was quite complex to comprehend. In the initial phase, the research work started on software security and safety, but soon the focus shifted to reducing waste and raw materials to save the environment [63]. Many publications have been done on the data centres' sustainability that covers the hardware's aspects. The concept of Green IT also covers the aspects of the hardware.

The study for sustainable and green development was introduced in the year of 1987 for the first time. It was introduced at the platform of the World Convention Environment and Development. At this platform, it was defined that the green software processes are those that meet the demands of the present as they are designed in such a way. Also, they do not compromise the ability of future generations to meet the demands on their own. After this, the concept heralded a new era for software organisations [18]. In the beginning, the concept of green software was quite complex to comprehend. In the initial phase, the research work started on software security and safety, but soon the focus shifted to the reduction of waste and raw materials to save the environment [4]. Many publications have been done on the sustainability of the data centres that cover the aspects of the hardware. The aspects of hardware are also covered by the concept of Green IT. For example, for the efficiency of the data centre, the relegation of code of conduct [48]. There is nearly no presence of executioner models or software tools that are capable of estimating the consumption of energy in the design stages of the initial or early period [52]. Such mixed roles have put much pressure on technological organisations due to conflicts. On the internal front, they are considered eco-friendly, while on the external front, they were expected to design new products capable of improving societal sustainability at large [49]. In terms of quality, the software industry, in the aspects of commoditisation, has come under much pressure to make and provide huge volumes of higher quality services and products that also lie within the defined cost. Also, the constraints for the schedule are also strict than ever before [52]. According to McInnes et al. 2017, there are different advice and activities regarding the exercises for the development of green software that is capable of using lower energy or that is energy efficient and also produces lower waste at the same time. According to him, there are three ways for software engineering to be as green as possible such as producing green software, producing support that is capable of supporting the consciousness regarding the environment and producing lower waste during the development of this software [54]. According to Alves, there are basic right ways for the engineers to go green regarding the software systems, their usage, and development procedures, and in this way, they can ensure that the aspects of both positive and negative nature of the software are being monitored consistently. This will help in evaluating the results that can lead towards further optimisation of software over its life cycle to be greener or environment friendly than ever before [43]. There are ways to improve the software energy efficiency on the systems of multiple cores. According to him, for that purpose, motivations must be driven by the huge influence of information communication technology on the global carbon dioxide emissions, which encapsulates the 2 per cent [62]. Green software is those that can consume a lower amount of energy, and they can run on this minimum amount of energy. Such software is also capable of producing less or little waste during the development phase. Broadly, the researchers focus on green software to make it more energy-efficient.

For example, for the efficiency of the data center, the relegation of code of conduct [13]. There is nearly no presence of executioner models or software tools that are capable of estimating the consumption of energy in the design stages of the initial or early period [58]. Such mixed roles have put much pressure on technological organisations due to conflicts. On the internal front, they are considered eco-friendly, while on the external front, they are expected to design new products capable of improving the societal sustainability at large [49]. In terms of quality, the software industry, in commoditization, has come under much pressure to make and provide vast volumes of higher-quality services and products that also lie within the defined cost. The schedule constraints are also stricter than ever before [56]. There are different advice and activities regarding the exercises for the development of green software that is capable of using lower energy or that is energy efficient and also produces lower waste simultaneously [54]. According to him, there are three ways for software engineering to be as green as possible such as producing green software, producing support that is capable of supporting the consciousness regarding the environment and producing lower waste during the development of this software [54]. There are basic right ways for the engineers to go green regarding software systems, their usage, and development procedures. In this way, they can ensure that the aspects of the software's positive and negative nature are monitored consistently. This will help evaluate the results that can lead towards further optimization of software to be greener or environmentally friendly than ever [43]. There are ways to improve the software energy efficiency on the systems of multiple cores. For that purpose, motivations must be influenced by the information communication technology on the global carbon dioxide emissions, which encapsulates the 2 per cent [62]. According to Fitzgerald, green software is those that can consume a lower amount of energy, and they can run on this minimum amount of energy. According to him, such software can also produce less or little waste during the development phase. The researchers focus on green software development to make it more energy-efficient.

2.3.1 Sustainability perception in software systems

The perception of sustainability is considered a way of creating balance in the lives of humans. It is perceived as a way of utilising products, services and other resources to have a minimum negative impact on the living beings, whether they are plants, animals or humans [46]. This explicitly explains that the software systems are required to build such an endurable system that is not harmful to an ecosystem in the condition of present and the conditions of future as well. Meanwhile, they are also required to be able to cater to or satisfy the needs or demands of today's users and those who are going to use it tomorrow with the least negative impacts on the environment. They must also be able to support business growth and society's values simultaneously [52]. The dimensions of software sustainability are further categorized into five categories such as economic, social, individual, technical and environmental. However, no clear perception regarding the said categories has been found in software engineering. The evolution of sustainability of the software in today's world is perceived in the following forms: sustainability in the development of software, software for sustainability, green software systems, sustainable software ecosystems etc. [59]. Sustainable software development refers to the involvement of processes in software development. At the same time, the software for sustainability refers to how software is used to assist sustainability. For instance, the software inside a refrigerator minimizes energy waste [47]. The green software systems can be perceived in terms of their impact as these software systems use lower energy resources and promote policies that assist in creating awareness regarding green software. Lastly, the sustainability of the software ecosystems can be perceived by checking their impact on the entire software ecosystem [50]. The perception of software sustainability has gathered the attention of various researchers in the past few years. However, amongst them, the most focused perceptions for advanced research are mainly green software systems and suitability for software development.

2.3.2 Software sustainability considerations

The software sustainability consideration related to safety appeared on the surface in 2010, which explains that safety is an emergent property when it interacts with the environment. This required supporting sustainability as it was known that stakeholders would be the critical challenging or succeeding factor behind all projects of sustainability and also for the green software [45]. Much of the focus was put on the effort to fight against the issues of pollution, especially the efforts of the European Union, as the EU has limited the amount of greenhouse gas emission for each country, and it can be traded in the form of emission permits. Installation must be done in a way that they can hold the credit but not able to exceed the limit of the cap set by the EU. If the installations can result in the emission of more gasses, then they have to pay to get the credit. That is why it is important for the manufacturers to closely monitor their carbon and environmental footprint to avoid any hazardous substances emitted into the environment [53]. This method must be set for all the software developing industries. The reduction in the consumption of energy and the impacts that it can have on the entire environment of data centers is going to be the top research topic in the future years. The development of software plays a specific role in the creation of rebounding effects. Usually, the response of the software engineers is to increment the power of processing and capacity of storage available at the given price to gain more of the same [6].

2.3.3 Software ecosystems' sustainability

In today's world, software systems are considered the backbone of the economy. The creation of the software ecosystem is considered to be one of the biggest systems created by humans. Software ecosystem is a set of various actors functioning together as a unit and interacting via a shared market for the services and software. They also have a relationship [18]. A common technological platform or market frequently underpins these relationships, and they operate through exchanging information, resources and artefacts. So, the sustainability of the global system of a software system involves the sustainability of the software ecosystems, and they cover the different aspects such as various sub-systems from the biggest interconnected system and all the interactions [51]. The different components are also included in it, like hardware, software and network that is utilised for the resolution of complexity in relations amidst different companies and organisations operating in various sectors or industries. The sustainability of the ecosystem's software includes how the system of software systems can bear with the evolving requirements of the users and usage over time with lower negative effects on the environment, society and humans as well [60]. This means the ability of the software ecosystem to continue to function and evolve irrespective of any glitch is some part of the ecosystem and should continuously fulfil users' needs.

2.3.4 Perspective of design

In software engineering, the focus must be shifted to the design pattern of the software system to make it reusable, which will put an end to most of the problems such as wasting material that happens after the development or during the development of every software system. This work calls for advances in software engineering systems by considering the requirements of gee nans sustainable design processes. The focus must be shifted on the making of algorithm efficiency as well while making things run fast with the presence of less hardware [62]. But this is amplified and driven by pricing schemas of cloud resources and cost-saving: the need to write efficient software that can do more with limited resources. This will translate into power efficiency when there is an increase in the amount of work done per CPU Cycle. It is generally believed that green software engineering builds software in an efficient manner that they are capable of consuming a lower amount of energy. For that purpose, it must be made a global standard that practitioners must promote or else it would be too hard for the relevant people to promote it or raise awareness regarding this [38]. Back in the year 2007 environmental protection agency (EPA) data center reported to the conger of the United States of America that by the year 2011, the intense load would be shifted on the grid by the data centers located in the said country alone. This means that it would be close to 12 GW, which is nearly equal to the output from the 25 different base load plants of power where the community of the research need to focus efficiently and effectively [39]. For the purpose of chip manufacturing, it is explained that the resources' amount and energy consumed are measured as a ratio against the final products' weight (chip), which is said to be the highest among all the industries of the manufacturing domain. This shows that the impacts on the environment are associated with the industries working in the manufacturing domain, and this must be considered while analysis of the ecological impact of a personal computer as well [13]. It has been seen that in many developing countries, there is no sufficient presence of recycling facilities for the purpose of packaging and shipping personal computers [55]. For this purpose of disposal, on the other hand, developed countries have made and incorporated many laws for the recycling of e-waste due to the high cost of operations, while the developing countries do not speak of laws and when not, even proper recycling facilities exist for the e-waste [45].

2.3.5 Role of green metrics

There are various diverse approaches related to the software's green metrics. The term green metrics raises research questions for the green software engineering in the literature study as well. There are a few questions that are asked frequently, such as what kind of green metrics are proposed in software engineering, what the classification of the green metrics is and how they are used in the software engineering literature. Green factors indicate that software must fulfil its requirements or properties. The advancement in the software engineering systems of quality assurance such techniques is being built that are able to fulfil the requirements of the future research in the establishment of sustainability metrics for the development of green software as well as the techniques that could assess them [40]. The model related to green soft has the ability to demonstrate the categories of criteria for sustainability and metrics for the products of software. They are considered to be the common criteria of quality and metrics, and these criteria are direct and indirect metrics in nature [68]. In the development phase, the classification of quality properties such as modifiability and reusability take effect, whereas, in the usage phase, usability and accessibility take effect [33]. The model of green metrics for the sustainability of software engineering energy awareness in the software systems can be attained and calculated via green metrics, and they are known as green performance indicators (KPIs). This GPI can further be classified into four different classes Information technology resource usage GPIs, compute resource usage, life cycle applications' Key performance indicators and reconfiguration [12].

2.4 Methods for Measuring and Assessing Sustainability Indicators

There are various methods for measuring and assessing sustainability indicators in software development. The most commonly used methods are environmental impact assessment (EIA), life cycle assessment (LCA), and input-output analysis (IOA). EIA is typically conducted at the project planning stage and includes the identification of environmental impacts, the prediction of environmental impacts, and the evaluation of environmental impacts [17]. At the same time, LCA is used to assess the impact of a product or service on the environment. Finally, IOA is a process that is used to assess the economic impact of a proposed project or development [18]. IOA includes the identification of inputs and outputs, their quantification, and the analysis of inputs and outputs.

2.5 Measuring Software Sustainability

Software Sustainability is measured based on several quality attributes of software systems, which are necessary for their long-term performance. A detailed description of the sustainable software quality attributes has been given below:

1. **Functional Suitability:** The functional suitability of software indicates its ability to perform the required functions. It is necessary for software to be able to perform all the required functions efficiently in order to be considered sustainable [18]. Moreover, functional suitability also considers the software’s ability to be easily upgraded and modified to meet changing needs.
2. **Usability:** The usability of software determines how easy it is for users to learn and use the software. Sustainable software is easy to use and does not require extensive training for

users to be able to use it effectively. It is also important for software to have a consistent and intuitive user interface that users can easily navigate.

3. **Reliability:** The reliability of software indicates its ability to function correctly and without errors under specified conditions. Reliable software is essential for sustainable software systems as it ensures that the software will continue to perform its required functions correctly even as it undergoes updates and modifications [19].
4. **Efficiency:** The efficiency of the software indicates its ability to use resources efficiently. Sustainable software is designed to use resources such as memory and processing power efficiently in order to minimise the impact on the system as a whole.
5. **Flexibility:** The flexibility of software indicates its ability to easily be modified to meet changing needs. Sustainable software is made flexible so that it can be modified and upgraded as new requirements arise. This allows software systems to be easily adapted to changing requirements without any major overhaul.
6. **Manageability:** Software system manageability indicates its ability to be effectively monitored and controlled. Sustainable software is designed for easy manageability so that it can be effectively monitored and controlled [20]. This helps ensure that software systems remain reliable and efficient over time.

2.6 Systematic Literature Review (SLR)

2.6.1 Systematic Literature Review Protocol

A review process offers a detailed road map for carrying out literature reviews, including systematic reviews, scoping evaluations, and meta-analyses. To ensure that the review procedure is transparent and uniform overall, the review group must create the protocol prior conducting the literature review [70]. Precisely, the protocol should provide detailed instructions on how to find and select pertinent articles for assessment, as well as describe the review procedures during the whole procedure. A review procedure is crucial for minimising review process biases and duplication of previous assessments [71].

Additionally, it offers a review process framework that aids in planning and foreseeing any difficulties. The methodology can assist the review group or future scientists in using the similar procedure to modify the review of literature whenever recent research becomes accessible after the review is finalised [72]. As a result, this research follows the aforementioned protocol for conducting in-depth systematic review of relevant literature. The key processes involved in this research's SLR protocol are presenting keywords and search strategy, highlighting inclusion and exclusion criteria, discussing quality assessment criteria and exploring data extraction approach.

2.6.2 Keywords

Lack of relevant and adequate keywords could result in systematic reviews having a one-dimensional picture of the study, making it hard to produce precise results founded on scientific knowledge [70]. Owing to this, specific terms were assigned to the keywords in this study's targets before they were used in conducting search process for the literature. "Software systems," "software sustainability concerns," "software sustainability perception," "software ecosystems sustainability," "Sustainability Indicators," "Software Sustainability," and "taxonomy of sustainability indicators for software development." were the keywords utilised.

2.6.3 Search Strategy and Process

Google Scholar and Science Direct were the two data sources that were deemed crucial when searching data for this research in order to explain outcomes that might improve the evaluation, achieve the goal of the study, and discover the research gap. The decision to use Google Scholar was made because it is an open-ended, unlimited source with a large portion of data pertaining to numerous journals. Accordingly, using the right keywords and Boolean operators, data was searched in this database, and adequate filters like year spectrum were used to refine the search results.

7,400 items relating to taxonomy of sustainability metrics for software development were found in the initial search. Nevertheless, 5 of the most pertinent studies were ultimately chosen for dissemination of data and critical evaluation of outcomes following more focused searches and consideration of inclusion and exclusion criteria. Similar to that, Science Direct, an open-source repository of papers from peer-reviewed journals, was also chosen to assemble research through the use of search techniques utilising the chosen keywords. 39 studies were found in this database during the preliminary search; nevertheless, 36 studies were excluded from the database once the appropriate filters were applied and the inclusion and exclusion criteria were taken into account. The search tactics for the two databases are shown in the table underneath.

Table 2.1: Search Strategy

Source: [Self-made]

	Concept 1		Concept 2		Concept 3
Strategy 1	Influence	**AND**	sustainability indicators	**AND**	software development
Strategy 2	Assessing	**AND**	Sustainability perception	**AND**	Software sustainability considerations

2.6.4 PRISMA Flowchart

The PRISMA structure below illustrates the procedures for selecting papers for systematic reviews. The eight most suitable articles were selected for investigation after the relevance of the publications was assessed using inclusion and exclusion criteria as well as a consistency evaluation.

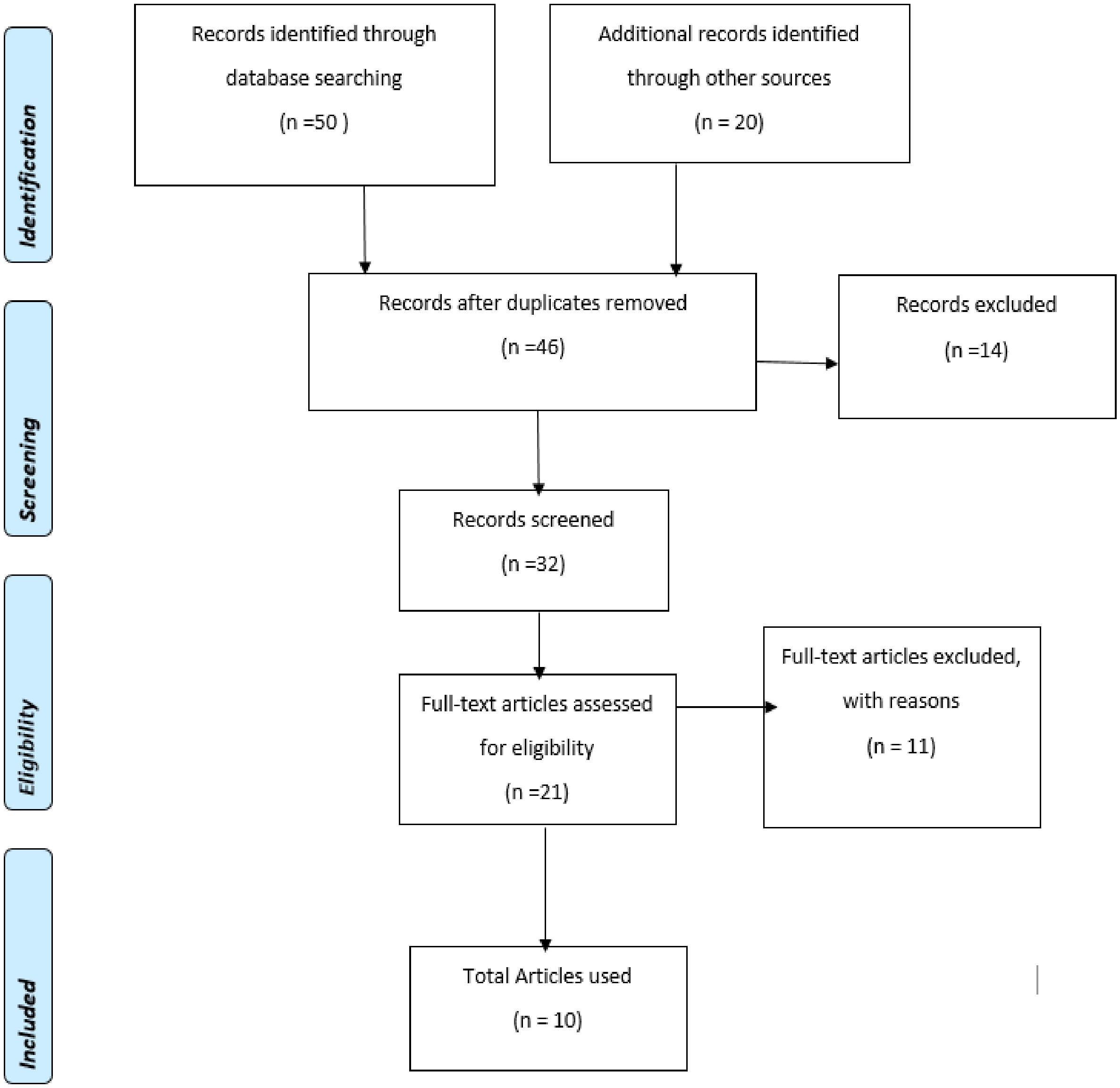

Figure 2.1: PRISMA Flowchart

Source: [Self-made]

2.6.5 Inclusion and Exclusion Criteria

The methodological structure's inclusion and exclusion parameters assess the choice of researchers and the investigations in relation to the study's goal, purpose, and reason. Additionally, the inclusion criterion has been regarded as the characteristic and rational part that might function as the study's foundation. On the contrary, the exclusion criteria refer to the traits that are not taken into account by the research design [73].

2.6.5.1 Inclusion Criteria

It has been argued that research predicated on more than 2500 research publications does not lead to reasonable and appealing results if a specified strategy of inclusion and exclusion is not followed [71]. In order to include only those researches that are focused on sustainability indicators, particularly in software development, the inclusion criteria for this study were maintained. In order to reduce the number of papers, the criteria for inclusion and exclusion have also been designed so that redundant information and technical articles are not included in the research.

The inclusion took about ten articles in the end which were chosen based on the mentioned criteria in the PRISMA chart in figure 1. It has 50 articles in the start and only 10 are chosen in the end of sorting process. The screening shows that most of the articles are being removed for the reason of duplication and irrelevancy due to the limit below the years in selection. The screening has removed about 14 articles from 32 and finally, more 11 articles were removed. The ten are chosen on the basis of the relevant information and catering of the data in the correct manner. It is in the limit of the defined years and the range of keywords.

2.6.5.2 Exclusion Criteria

In terms of exclusion criteria, articles that did not emphasise any sustainability metrics for software development were discarded. In the same manner, the literature search backed away from the effects of sustainability awareness and software sustainability concerns on software development. Studies from other countries and locations besides the UK, notably outside of Europe, were also omitted from this investigation. By applying these inclusion and exclusion criteria, a prolonged inquiry may be conducted, circumventing the issues with secondary studies [72].

Most of the articles are removed due to the duplication of the points. It is evaluated that a maximum number of the articles are removed, as mentioned in the image. This proves that the irrelevant ones and the ones below the limit of the year are already removed as they may not be any more for the sorting process. In the first go, about 14 articles were removed, leading to 11 more to be removed in the second go before finalising the articles as ten only.

2.6.6 Quality Assessment Criteria

It is as crucial to evaluate the robustness of the information in a systematic review as it is to examine the evidence. Biases resulting from the research methods might affect the findings of an inadequately performed research; therefore, they should be evaluated with prudence [73]. Such papers must to be explicitly omitted or at least identified as such in the systematic assessment. It is also crucial to use the right technique to assess the quality of the evidence and any embedded biases in each study [72]. Quality evaluation standards from the 'Critical Appraisal Skills Programme (CASP)' has been used in this study, which is presented hereunder:

Section A: Is the basic study design valid for a randomised controlled trial?			
1. Did the study address a clearly focused research question? CONSIDER: • Was the study designed to assess the outcomes of an intervention? • Is the research question 'focused' in terms of: • Population studied • Intervention given • Comparator chosen • Outcomes measured?	Yes	No	Can't tell
2. Was the assignment of participants to interventions randomised? CONSIDER: • How was randomisation carried out? Was the method appropriate? • Was randomisation sufficient to eliminate systematic bias? • Was the allocation sequence concealed from investigators and participants?	Yes	No	Can't tell
3. Were all participants who entered the study accounted for at its conclusion? CONSIDER: • Were losses to follow-up and exclusions after randomisation accounted for? • Were participants analysed in the study groups to which they were randomised (intention-to-treat analysis)? • Was the study stopped early? If so, what was the reason?	Yes	No	Can't tell
Section B: Was the study methodologically sound?			

4. • Were the participants 'blind' to intervention they were given? • Were the investigators 'blind' to the intervention they were giving to participants? • Were the people assessing/analysing outcome/s 'blinded'?	Yes	No	Can't tell
5. Were the study groups similar at the start of the randomised controlled trial? CONSIDER: • Were the baseline characteristics of each study group (e.g. age, sex, socio-economic group) clearly set out? • Were there any differences between the study groups that could affect the outcome/s?	Yes	No	Can't tell

Section C: What are the results?

6. Apart from the experimental intervention, did each study group receive the same level of care (that is, were they treated equally)? CONSIDER: • Was there a clearly defined study protocol? • If any additional interventions were given (e.g. tests or treatments), were they similar between the study groups? • Were the follow-up intervals the same for each study group?	Yes	No	Can't tell

7.	Were the effects of intervention reported comprehensively? CONSIDER: • Was a power calculation undertaken? • What outcomes were measured, and were they clearly specified? • How were the results expressed? For binary outcomes, were relative and absolute effects reported? • Were the results reported for each outcome in each study group at each follow-up interval? • Was there any missing or incomplete data? • Was there differential drop-out between the study groups that could affect the results? • Were potential sources of bias identified? • Which statistical tests were used? • Were p values reported?	Yes	No	Can't tell
8.	Was the precision of the estimate of the intervention or treatment effect reported? CONSIDER: • Were confidence intervals (CIs) reported?	Yes	No	Can't tell
9.	Do the benefits of the experimental intervention outweigh the harms and costs? CONSIDER: • What was the size of the intervention or treatment effect? • Were harms or unintended effects reported for each study group? Was a cost-effectiveness analysis undertaken? (Cost-effectiveness analysis allows a comparison to be made between different interventions used in the care of the same condition or problem.)	Yes ☐	No ☐	Can't tell ☐

Section D: Will the results help locally?

10.	Can the results be applied to your local population/in your context? CONSIDER: • Are the study participants similar to the people in your care? • Would any differences between your population and the study participants alter the outcomes reported in the study? • Are the outcomes important to your population? • Are there any outcomes you would have wanted information on that have not been studied or reported? • Are there any limitations of the study that would affect your decision?	Yes ☐ No ☐ Can't tell ☐
11.	Would the experimental intervention provide greater value to the people in your care than any of the existing interventions? CONSIDER: • What resources are needed to introduce this intervention taking into account time, finances, and skills development or training needs? • Are you able to disinvest resources in one or more existing interventions in order to be able to re-invest in the new intervention?	Yes No Can't tell ☐ ☐ ☐

Figure 2.2: CASP TOOL

2.6.7 Data Extraction

Since a systematic review was utilised as the methodology for data gathering and evaluation, this study was carried out using a quantitative approach. As noted earlier, primary data gathering was undertaken in order to perform a systematic review of publications and academic papers. Two well-known databases, Google Scholar and Science Direct, served as the primary data extraction sources for the data extraction.

2.7 Identification of Software ecosystems' sustainability Indictors

In today's world, software systems are considered to be the backbone of the economy. The creation of the software ecosystem is considered to be one of the biggest systems created by humans. Software ecosystem can be defined as a set of various actors that are functioning together as a unit and interacting via a shared market for the services and software, and they also have a relation amongst them [61]. These relationships are frequently underpinned by a common technological platform or market, and they operate through the exchange of information, resources and artefacts. So, the sustainability of the global system of a software system involves the sustainability of the software ecosystems, and they cover the different aspects such as various sub-systems from the biggest interconnected system and all the interactions [51]. The different components are also included in it, like hardware, software and network that is utilised for the resolution of complexity in relations amidst different companies and organisations operating in various sectors or industries. The sustainability of the ecosystem's software includes how the system of software systems can bear with the evolving requirements of the users and, over time, usage with lower negative effects on the environment, society and humans as well [60]. This means the ability of the software ecosystem to continue to function and evolve irrespective of any glitch is some part of the ecosystem and should continuously fulfil users' needs

Table 2.2: Systematic Literature Review

Name	Definition	Green Software	Software for sustainability	Lower energy consumption	Sustainability considerations	Sustainability dimension	Design perspective	Software ecosystem's sustainability
Software sustainability [18]	It was defined that the green software processes are those that meet the demands of the present as they are designed in such a way.	Yes	Yes	Yes	No	Environment Technical	Not given	Yes, focused on

A model for estimating social and economic indicators of sustainable development [46]	This explicitly explains that the software systems are required to build such an endurable system that is not harmful to an ecosystem in the condition of present and the conditions of future as well.	Yes	Yes	Not mentioned	Yes	Environ ment	Not given	Highly focused
Towards sustainable development through the perspective of eco-efficiency [45]	Installation must be done in a way that they are able to hold the credit but not able to exceed the limit of the cap set by the EU.	No	Yes	Yes	Partial	Environ ment	Focused systematic review	The main indication, focused

Software sustainability [18]	The sustainability of the global system of a software system involves the sustainability of the software ecosystems, and they cover the different aspects such as various sub-systems from the biggest interconnected system and all the interactions.	No	Yes	Not mentioned	Highly focused	Environment ecosystem	Not mentioned	Major focus
Sustainable software products [52]	The dimensions of software sustainability are further categorised into five categories such as economic, social, individual, technical and environmental.	Not mentioned	Yes	Not mentioned	Yes	Environment, social, economic, individual and technical	Not given	Yes, focused

Digital sustainability: basic conditions for sustainable digital artefacts and their ecosystems [60]	It is generally believed that green software engineering builds software in an efficient manner that they are capable of consuming a lower amount of energy.	Yes	Yes	Yes	Yes	Environment Technical	Design perspective is given	Highly focused
Design and development of secure and sustainable software-defined networks [08]	The model related to green soft has the ability to demonstrate the categories of criteria for sustainability and metrics for the products of software.	Yes	Yes	Yes	Partial	Environment Technical	The design perspective for green software is given	Yes

Sustainability transformers [14]	A new empirically derived taxonomy and its policy implications driven from the transformation of sustainability practices supported by new start-up initiatives	Yes	Yes	Yes	No	Environment ecosystem	Not given	Yes, focused on
Software sustainability [13]	Development of sustainable software with the help of Lean ISO 14001 approach that has been facilitated by smart frameworks	Yes	Yes	Yes	No	Not mentioned	Based on Lean ISO 14001 setup	Highly focused

Software sustainability [20]	Development of sustainable software practices based on the machine learning methods	Yes	Yes	No	Yes	Not mentioned	Based on machine learning approach	Yes, focused

2.8 Conceptual Method

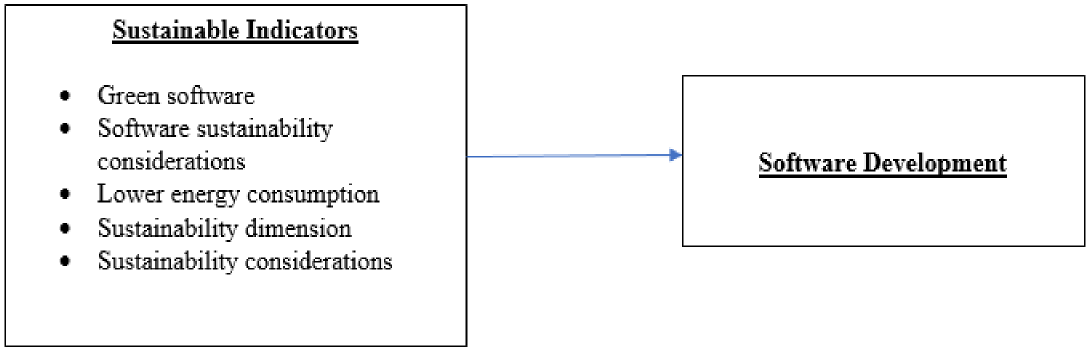

Figure 2.3: Conceptual Method

2.9 Theoretical Framework

In order to properly understand and study the taxonomy of sustainability indicators for software development, it is important to first establish a theoretical framework. This theoretical framework will provide the necessary foundation upon which the taxonomy can be built. There are three key concepts that must be understood in order to establish this theoretical framework: systems thinking, the triple bottom line, and sustainability indicators. Systems thinking is a holistic approach to problem-solving that considers the relationships between and among different elements in a system [21]. This type of thinking is important for understanding how sustainability indicators can be used to track the performance of software development ecosystems.

The triple bottom line is a framework for thinking about sustainability that includes three key dimensions: environmental, social, and economical. This framework is important for understanding how sustainability indicators can be used to track the performance of software development ecosystems. At the same time, sustainability indicators are tools that can be used to measure and track the performance of systems with respect to sustainability [22]. There are many different types of sustainability indicators, and they can be used to track the performance of software development ecosystems along the three dimensions of the triple bottom line. This theoretical framework provides a foundation for understanding how sustainability indicators can be used to track the performance of software development ecosystems. By understanding these three key concepts, it is possible to develop a taxonomy of sustainability indicators for software development.

In addition, there is another significant theory that can be used in this taxonomy, known as the systems approach. The systems approach is a theoretical framework that considers the relationships between and among different elements in a system [23]. This type of thinking is important for understanding how sustainability indicators can be used to track the performance of software development ecosystems. The systems approach is a holistic approach to problem-solving that considers the relationships between and among different elements in a system [24]. This type of thinking is important for understanding how sustainability indicators can be used to track the performance of software development ecosystems.

2.10 Knowledge Gap/Literature

The need for the sustainability of the software ecosystem is widely discussed in the literature. However, there is a lack of comprehensive and systematic taxonomy for the indicators of sustainability for software development [12]. The taxonomy is important as it provides a structured and standardised way to measure the sustainability of software development [25]. Moreover, it has also been identified that little research has been conducted on the assessment of the environmental impacts of software development. Finally, there is also a need to develop an assessment framework for the sustainability of software development [17]. This will help to improve the understanding of software development and its impacts on the environment and also help to evaluate the needs that must be addressed to improve the sustainability of software development.

Based on the identified literature gap, SWOT analysis of the study Sustainable development in developing societies by Nwankwo and Njoku (2020) highlighting the research gap has been presented for the reference. The study provides a comprehensive and systematic taxonomy for the indicators of sustainability for software development. The study discusses the need for the sustainability of the software ecosystem and identifies a lack of comprehensive and systematic taxonomy for the indicators of sustainability for software development as a key research gap. However, the major weakness is that, the study does not provide an assessment framework for the sustainability of software development. Moreover, the study provides an opportunity to develop an assessment framework for the sustainability of software development. Finally, the study does not identify any specific threats. However, the lack of an assessment framework for the sustainability of software development could be considered a threat to the success of the study.

2.11 Chapter Summary

This chapter overviews the literature on software development sustainability. It starts with a discussion on how sustainable software development started and the perception of sustainability in software systems. It then discusses the various considerations for the sustainability of software development. This is followed by a discussion on the sustainability of software ecosystems. Finally, the chapter provides a summary of the literature gap. Overall, the second chapter of the dissertation has summarised the research work of various researchers in this domain and provided the findings of the previous research. This has helped in providing the direction for conducting the current research work. Moreover, this chapter has also provided a detailed literature review which will help in understanding the research gap that exists in this domain.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This section of the study is based on research methodology. Research methodology is referred to as the particular process of techniques that is used for the identification, selection, processes and examining the data with regards to the particular topic. The methodology of the research allows the reader to critically examine the overall study's validity and reliability. In similar, another researcher concluded that research analysis is a significant aspect of the research findings that is processed based on providing justification of the methodology components. This chapter of the methodology is based on various components that are necessarily determining appropriate approaches and tools that are used during the analysis. The main component of the methodology is research philosophy, research approach, research design, data collection method, data analysis, ethical considerations and limitations.

3.2 Research Philosophy

Research philosophy is referred to as the belief regarding the way in which data related to a phenomenon should be collected, analysed and used [26]. Research philosophy means dealing with the source, nature and knowledge development which is important to identify the sustainability indicators for software development that assist in providing effective understanding of human perspectives and thoughts that will help in the following research. Research philosophy is categorised into three main types, including positivism, interpretivism and pragmatism. It deals with the source, nature and knowledge development [27]. This study utilises positivism research philosophy considering the nature of the study as the following study is based on mathematical study and numerical calculations that conveys the belief that the findings are precise and trustworthy. The theory of positivism observes truthful knowledge that is obtained with the help of observations that is developed in existing research or general interpretations formulated from the human mind, which is, therefore, observed by explanations of measurements and mathematical approaches [28]. The following paradigm lacks consideration of human interest within the research and is fully an independent form of research. The rationale behind adopting the positivism philosophy for this research is that this research prefers the quantitative method the most, which entails quantifiable surveys, questionnaires and statistics because of their improved reliability and effective representation of the dataset [29]. By making use of the positivism research philosophy, the researcher has analysed the sustainability indicators that play a major role for the purpose of software development through the statistical elements.

3.3 Research Approach

The approach of the research is an efficient element of the study. The research approach enables towards defining of the methods that could be used for obtaining data and applying different analysis techniques [30]. As per the research conducted by states that the research approach is divided into three common types, including inductive and deductive approaches [29]. Considering the current research, the deductive approach of the research has been opted for. The purpose behind opting deductive approach is that it assists in providing reasoning to the researcher, considering the properties of the research as proving strong evidence. Moreover, the following research approach has the capability of explaining casual connections among notions and variables that will help in benefitting generating the concepts regarding the taxonomy of sustainability indicators for software development [31]. Though, it creates the opportunity for determining the notions quantitatively in terms of streamlining the research findings to particular extents. The deductive approach is the right choice for collaborative use for SPSS data handling. Deductive approach aims to offer support from the general to the specific for reviewing the data. In this manner, the extra and unrequired information will be removed in the steps and only the relevant and required information will be streamlined in the end [33].

3.4 Research Design

Research design is defined as the overall strategy that is chosen by the researcher for the implementation of distinct components of the study in a coherent and logical manner that aids in effectively addressing the research problem with the help of effective data collection, measurement and assessment [33]. Moreover, the research design provides an effective structure for the overall research. Research design is characterised by three main types, including qualitative, quantitative and mixed-method design. Qualitative research design has relied on subjective analysis that is dependent on prior literature and findings. On the other hand, quantitative research is usually dependent on numeric and statistical data. Whilst, mixed-method design is a combination of both qualitative as well as quantitative research design.

For the present research, a quantitative approach is adopted by the researcher as the following research design aids in collecting primary data that has the ability to afford results with high precision along with appropriate statistics [34]. The purpose of this research design is that it is one of the best suitable methods for the research and could be obtained through the best probable results and findings. However, there are numerous reasons for the execution of quantitative research design that assists in the collection of data related to the taxonomy of sustainability indicators for software development from a larger sample size. The quantitative method enables examining the relationship among different variables, which is not possible in qualitative design [32]. With the utilisation of the following research design, the investigator has the ability to develop the research's objectivity and precision.

3.5 Data Collection Method

The approach of data collection is that in which the data is gathered for the purpose of the selected topic. Often data collection technique is implemented after the experiment or observation. As per [35], data collection methods are of two main kinds, namely primary and secondary data collection methods. The primary data collection method is the one in which the information is directly collected from the targeted audiences and is collected in raw form. Thus, it is known as first-hand information. Primary data is collected through interviews, surveys, questionnaires and focus groups. On the other hand, the data gathered from prior studies or literature conducted by different authors are known as the secondary data collection method. Therefore, it is known as second-hand information [36]. For the present research, the investigator has adopted the primary data collection method as it is obtained from the participants directly. The current research utilises the primary method of data collection, where the data will be collected from the survey questionnaire technique. Additionally, the primary data collection method provides consistent and efficient data as the data is gained from the actual source. In order to collect information, the questionnaire has helped in obtaining data from a larger sample in the form of subjective and objective ways [37]. The other reason behind using a qualitative survey questionnaire is that it allows the investigator to ask for close-ended questions with regards to substitute responses accessible. The questionnaire will be designed based on the Likert scale.

3.6 Sample Size

The sample size is defined as the term used in market research for explaining the number of subjects that are involved in the sample size [38]. A sample is a specific group from whom the data will be collected. The sample size is the one that represents the overall population. The method of survey questionnaire has been used in the current research in which the information will be collected from 81 respondents.

The estimation of adequate sample size is provided with the formula the equation which aids in providing adequate sample size for the purpose of determining the target population is illustrated below;

$$n = \frac{z^2 \times p \times q}{e^2}$$

As per the above equation, ‘z’ denotes to the z-score which is estimated at 0.9 and CI is computed at 95%. In addition to this, ‘e’ refers to the error which is estimated at 2% and ‘q’ denotes to the population which has not been considered in the study. Moreover, ‘p’ is considered as the variability proportion which is computed at 20%.

$$n = \frac{(0.9)^2 \times 0.2 \times 0.2}{(0.02)^2} = 81$$

Upon the prior estimation, it is evident that the adequate sample size is computed to be 81. In this manner, it infers that the sample size of 81 would be adequate for the purpose of conducting the study. In contrast to this, considering the response rate is also essential while targeting the desired respondents.

3.7 Data Analysis

According to [39], data analysis is an approach from which the data is modified and demonstrated for the development of efficient and valuable information. However, it will be useful for the business in terms of the decision-making process. There are various techniques of data analysis through which the data could be assessed thematic analysis, content analysis, frequency analysis, regression analysis and correlation analysis [40]. The focus of the present research is on the primary quantitative method. Therefore, the data has been analysed using SPSS software through frequency analysis, regression and correlation analysis. By using SPSS, bar charts are developed to illustrate the responses of the participants. Bar charts helps in determining the answers provided by the respondents regarding the questions they were asked. For example, these kinds of bar charts are developed as shown below, which determines the respondents reviews based on the questions they were asked which was measured on Likert scale.

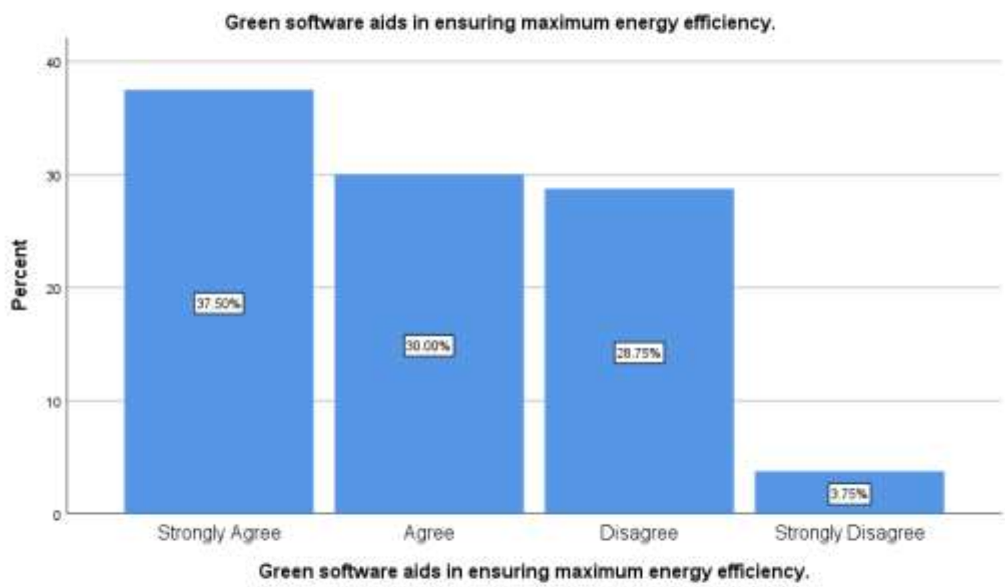

Figure 3.1: Demo

Furthermore, these methods ensued in effective consequences as frequency analysis aids in recognising the number of responses designated by the respondents. In a similar way, the regression test aids in assessing and understanding the relationships among different variables that are used for predicting the results [41]. Correlation analysis supports the study's author for understanding, measuring and analysing the connection of statistics.

3.8 Ethical Consideration

Ethical consideration is the gathering of principles and values that should be followed in doing human affairs [42]. The ethical considerations ensure that no one acts in such a manner that is harmful to society. There is a certain ethical consideration that focuses on certain basic fears of the study. It is necessary for the investigator to follow all suitable rules and regulations, and institutions when conducting the research. In terms of ethical consideration, the present research is licensed by the committee of ethical research. Furthermore, data and sources that are used in the research are developed through authentic and ethical sources where the personal information of the participants will not be shared with a third party. Thus, the survey questionnaire respondents are not endangered to any kind of damage. Furthermore, the confidentiality of the participants was guaranteed, and any kind of dishonesty in the aims of the study was disregarded. The responses of the participants will only be used for the present research and to maintain its confidentiality of it.

3.9 Limitations

There are various limitations of the study;

- The limitation that has been confronted by the researcher through the research process is of time. As time is the restraint that prevents the researcher from concluding the study more effectively.
- The sample size of the research is restricted to 80 participants for this research, whereas a larger sample size will help in enhancing the accuracy and interpretability of the research.
- The other constraint is budget and resources that have been faced by the researcher in this study.

CHAPTER 4

ANALYSIS AND RESULTS

4.1 Introduction

This chapter is predication on in-depth analysis and presentation of results based on the research area, which was focused on developing a taxonomy of sustainability indicators for software development. The results are generated via Microsoft Excel and IBM SPSS in the form of demographic, frequency, correlation, and regression analysis. The demographic analysis primarily determines the demographic characteristics such as age, gender, and occupation of relevant participants. The frequency analysis offers a simple interpretation on most frequent options selected by relevant respondents on each statement of closed-ended questionnaire. The correlation analysis indicates overall link between the independent and dependent variable(s). The results in regression analysis are mainly focused on determining the impact of each independent variable(s) on dependent variable(s).

4.2 Inferential Analysis

4.2.1 Demographic Analysis

According to [70], a popular form of inferential analysis that helps the investigator look at the movements and dimensions of a chosen sample is demographic analysis. In terms of science, [71] stipulated that prior acquiring actual data, researchers should acquire demographic details from the relevant group to determine their essential attributes, including such gender, age, and job position. Nevertheless, [72] argued that, for the sake of extrapolation, demographic data, which contains information on study participants is crucial for determining if the participants in a certain study are a realistic representative of the populace. Typically, the goal of demographic analysis is to gather meaningful data on the past or background of the target population [73].

In this study, a demographics analysis is done by utilising three aspects, including age, gender, and occupation of study respondents, since this study is centred on evaluating the taxonomies of sustainability indicators for software development. Below are several tables that illustrate the demographic analysis of the participants. Each statistic reflects a specific area where it is obvious which factor significantly affects the circumstance.

Table 4.1: Age Demographics

Age					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	18-25 years	23	28.7	28.7	28.7
	26-35 years	29	36.3	36.3	65.0
	36 years and above	28	35.0	35.0	100.0
	Total	80	100.0	100.0	

The age distribution of the participants who were selected for this study is shown in the accompanying table. The age ranges were divided into three groups: 18 to 25 years old, 26 to 35 years old and over 36 years old. According to the data, 28.7 percent of responders were between the ages of 18 and 25. In addition, participants aged 26 to 35 made up 36.3 percent of the overall population. Additionally, 35 percent of the respondents were older than 36 years old. The investigator was able to determine the mean age of the sampled population by using the age demographics. According to the results of the research above, most numbers of the participants are between the ages of 26 to 35 years. The validation shows that the higher percentage of 36.3% is from the major group of 26-35 years, followed by 36 years and above for 35%. This is a valid value as more experienced will be able to offer better output and good level of information. This has a positive inference on the research as the experience support will provide relevant and accurate information about the taxonomy of sustainability.

Table 4.2: Gender Demographics

Gender					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Male	57	71.3	71.3	71.3
	Female	23	28.7	28.7	100.0
	Total	80	100.0	100.0	

The gender profiles of study respondents are shown in the aforementioned table. These individuals were taken into account for the survey and offered informative responses on the key sustainability metrics for software development. From the table above, it can be seen that out of 80 responders, 71.3 percent of people were male and just 28.7 percent were female. Therefore, it can be determined that male respondents participated in this study at a higher rate than female respondents. The percentage validity shows dominance for the males as compared to the females. It shows a valid percentage of 71.3% for males followed by 28.7% for females which proves dominant responses value for valid output from male gender. The highest percentage of the males shows a positive validity towards data collection as they are continuing job at same workplace so they may provide relevant data.

Table 4.3: Occupation Demographics

Occupation					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Software Developer	8	10.0	10.0	10.0
	General Manager	72	90.0	90.0	100.0
	Total	80	100.0	100.0	

Focusing on the demographics of participants who were selected for this research according on their occupation, the table above was generated. Software developer, general manager, and strategic manager are the three categories that make up this demographic category.

In accordance with the aforementioned table, 10 percent of participants reported having a job title of "software developer," whereas general managers made up 90 percent of all survey respondents. Therefore, it may be assumed that the majority of respondents or 90 percent of the overall population surveyed for this study are working as General Managers. The validity output shows a difference of general manager and software developers among the participants in interviews. The valid value shows 90% validity for the general manager followed by 10% only for the software developers. This shows a higher demographic value of the general manager because they are specific in the information support for the sustainability taxonomy and its variables. So, in the responses as compared to the software developers, less percentage and manager's higher percentage is a positive support to cater right choices of data.

4.2.2 Frequency Analysis

One of the widely used techniques for descriptive statistics is frequency analysis, which shows the total number of responses formed by various respondents to a particular statement or proposition [74]. Frequency analysis is frequently utilised in inferential or numeric evaluations of studies when a scholar has obtained primary data from closed-ended questionnaire survey that could be measured and subjected to statistical assessment [75].

When frequency research is carried out, SPSS calculates the median, mean, and mode of the grouped data in order to look at the aggregate outcomes that may be utilised to offer accurate findings [74]. Nevertheless, it was unveiled that frequency analysis primarily assists the investigators in carrying out three main tasks of assessing the data accumulated from a survey, such as arranging and synthesising the general information of survey in table format, providing clear understanding of the survey data, and locating outliers that influence the final accurateness of result obtained [75].

In findings of systematic literature review, it was discovered that green software, software sustainability considerations, reduced energy usage, and sustainability dimension are the four independent factors in the current study whose influences are examined on software development, the dependent variable. Similarly, in this frequency analysis, the statements stated in the survey on each of the mentioned factors have all been carefully scrutinised to reflect on survey results, which can be compared with the findings systematic literature review in the next chapter. The graph below shows if green software aids in ensuring maximum energy efficiency:

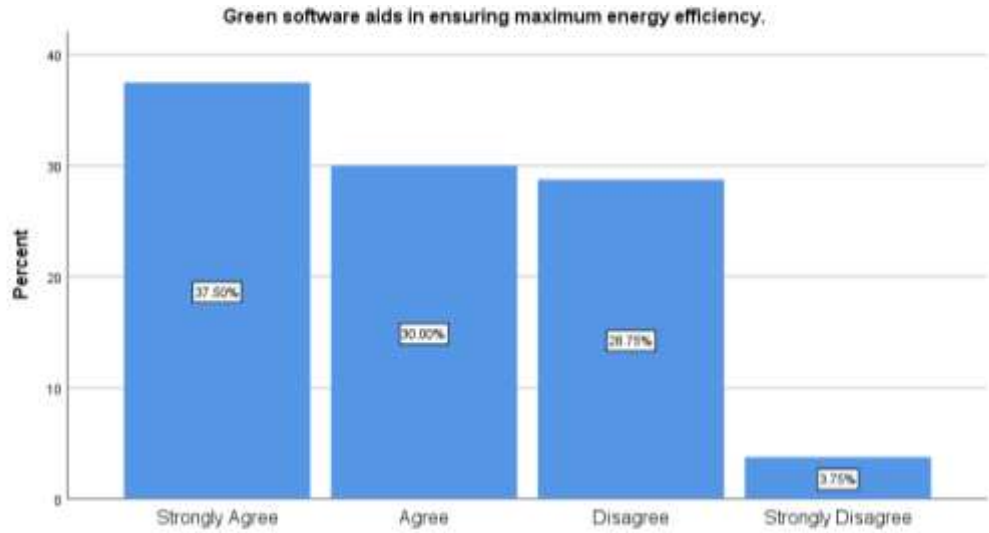

Figure 4.1: Green software aids in ensuring maximum energy efficiency

According to the accompanying figure, 37.50 percent of respondents indicated that they 'strongly agree' that green software helps to ensure optimum energy efficiency. Participants who "agreed" with the preceding assumption made up 30 percent of the population. However, 28.75 percent of respondents disagreed that using green software helps to ensure optimal energy efficiency while 3.75 percent of respondents strongly disagreed in the same situation. The above results are in conjunction with the findings of systematic review, which uncovered green software as effective sustainability indicator of software development. The following graph illustrates whether green software is the backbone of virtually all the intelligent solutions that is designed for supporting the overall environment.

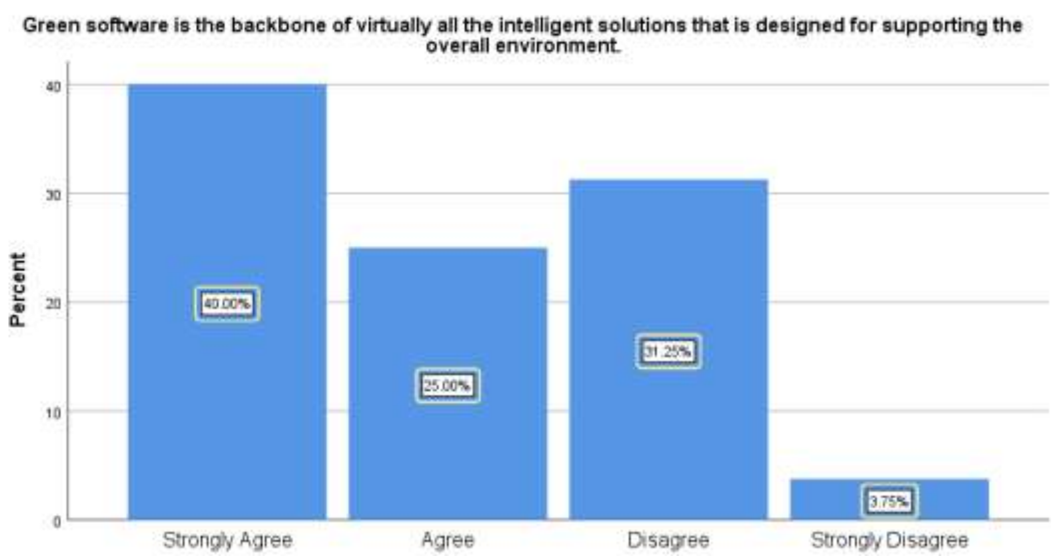

Figure 4.2: Green software is the backbone of virtually all the intelligent solutions that is designed for supporting the overall environment.

Considering the preceding graph, 40 percent of respondents said they 'strongly agree' that green software is the foundation of almost all intelligent solutions created to help the environment as a whole. The populace was composed of 25 percent of respondents who "agreed" with the previous premise. Nevertheless, 31.3 percent of respondents 'disagreed' that green software is the foundation of almost all intelligent solutions created to benefit the environment as a whole. 3.8 percent of respondents 'strongly disagreed' with the statement. The next chart is predicated on determining if green software's contribution to reduced greenhouse gas emissions:

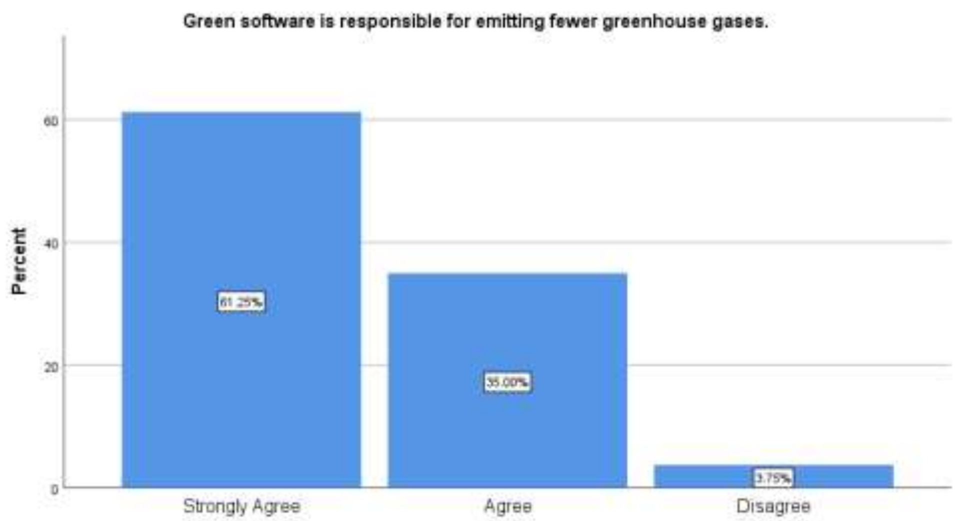

Figure 4.3: Green software is responsible for emitting fewer greenhouse gases.

Given the previous chart, 61.25 percent of participants reported that they "strongly agree" that green software is responsible for lower greenhouse gas emissions. 35 percent of participants "agreed" with the prior notion. However, 3.75 percent of participants "disagreed" that using green software results in a reduction in greenhouse gas emissions. The purpose of the following graph is to assess if it is crucial for software producers to carefully track their carbon and environmental impact.

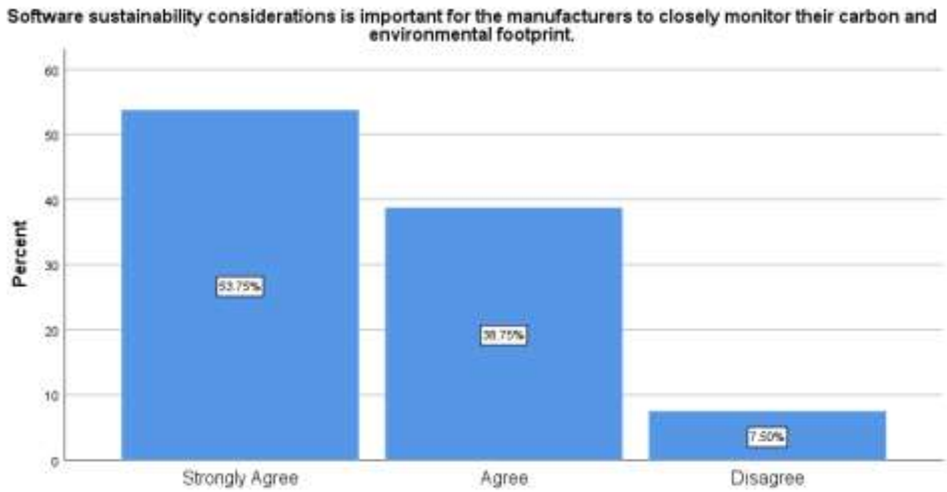

Figure 4.4: Software sustainability considerations is important for the manufacturers to closely monitor their carbon and environmental footprint.

From the graph above, it is clear that 53.75 percent of survey respondents "strongly agree" that software sustainability concerns are crucial for firms to constantly manage their carbon and environmental footprint. The previous assumption was "agreed" with by 38.75 percent of participants. 7.50 percent of participants "disagreed" that software sustainability concerns are crucial for firms to constantly manage their carbon and environmental footprint. The above results are in conjunction with the findings of systematic review, which uncovered that software sustainability consideration as effective sustainability indicator of software development. The following graph is centred on determining if the production of the rebounding effect is specifically influenced by the development of software:

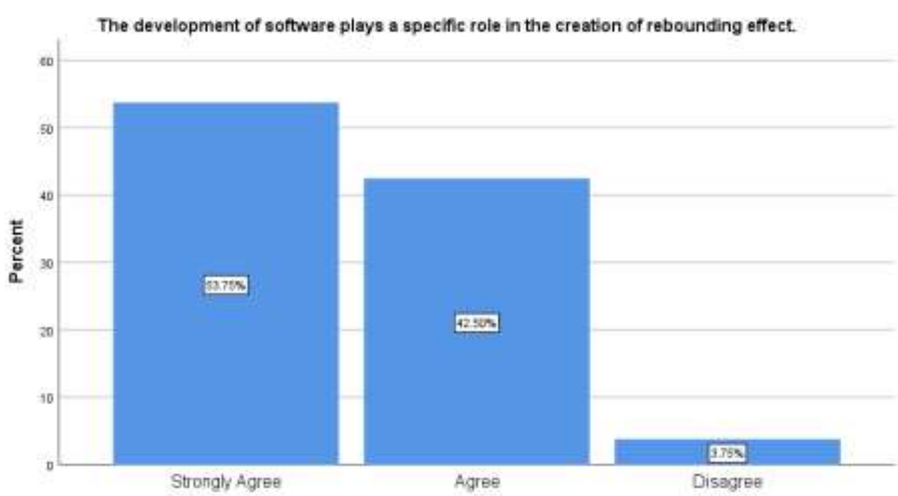

Figure 4.5: The development of software plays a specific role in the creation of rebounding effect

According to the graph above, 53.75 percent of participants "strongly agree" that software development has a particular impact on the development of the rebounding effect. On the other hand, 42.50 percent of respondents "agreed" with the preceding idea. Moreover, 3.75 percent of respondents "disagreed" that the creation of the rebounding effect is specifically influenced by software development. The following graph is focused on assessing whether software sustainability is important but are technically minded with regards to sustainability:

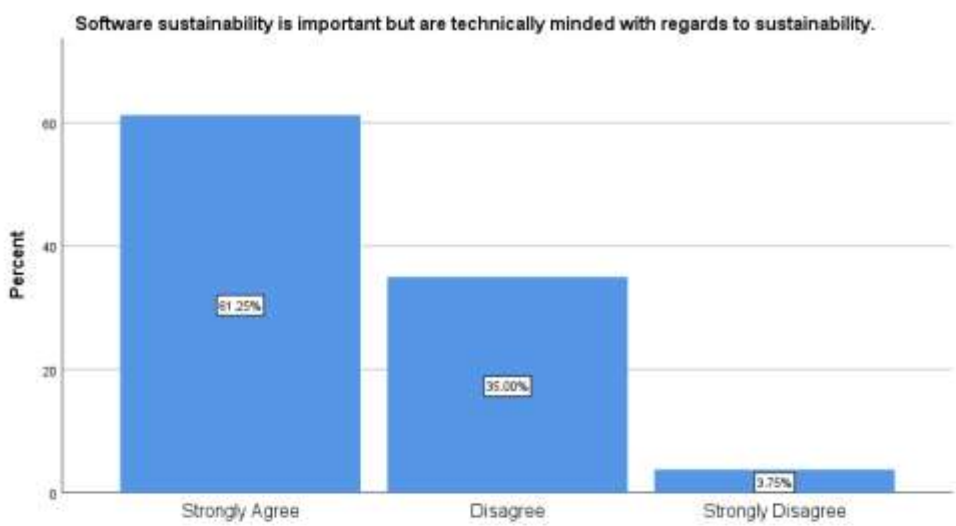

Figure 4.6: Software sustainability is important but are technically minded with regards to sustainability.

The graph above shows that while 61.25 percent of survey respondents "strongly agree" that software sustainability is vital, they have a technical outlook on sustainability. However, 35 percent of respondents "disagreed" and 3.75 percent "strongly disagreed" that software sustainability is important but are technically minded with regards to sustainability. The goal of the following graph is to evaluate if using less energy results in cleaner air and water as well as the preservation of natural resources:

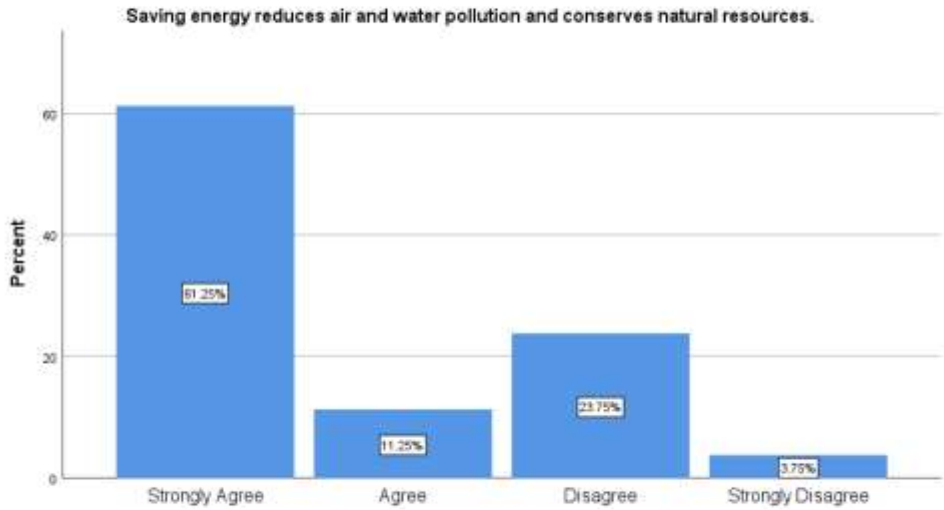

Figure 4.7: Saving energy reduces air and water pollution and conserves natural resources.

According to the graph above, 61.25 percent of participants "strongly agree" that conserving energy lowers air and water pollution and protects natural resources. Nonetheless, only 11.25 percent of participants, "agreed" with the aforementioned statement. In addition, 23.75 percent of respondents "disagreed" and 3.75 percent "strongly disagreed" that conserving energy helps the environment by lowering air and water pollution and preserving natural resources. The above results are in conjunction with the findings of a systematic review, which uncovered that energy conservation is as effective sustainability indicator of software development. The subsequent chart is focused on assessing if decreased energy usage contributes to efficiency, cost savings, and job creation:

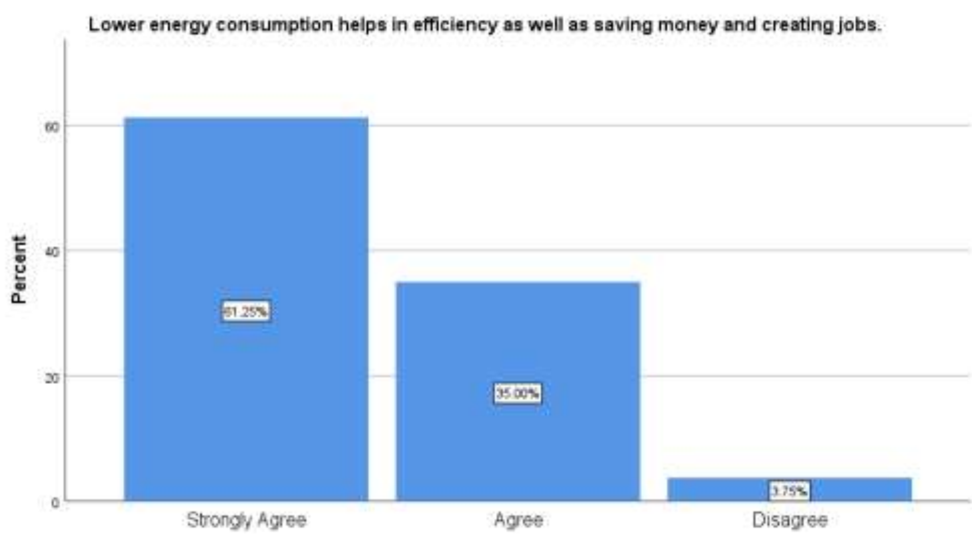

Figure 4.8: Lower energy consumption helps in efficiency as well as saving money and creating jobs

The graph above shows that 61.25 percent of respondents "strongly agree" that less energy use improves efficiency, as well as helping save money and generate employment. However, 35 percent of participants said that they "agreed" with the aforementioned assertion. Additionally, 3.75 percent of respondents 'disagreed' that a decreased energy usage also helps with efficiency, cost savings, and job creation. The following graph examines whether conserving energy tends to lower the demand for the energy sources that are now available:

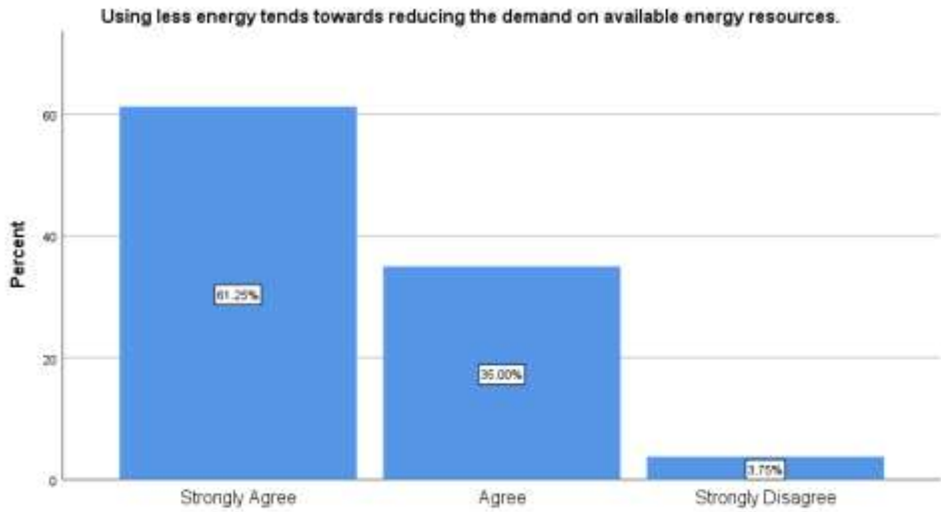

Figure 4.9: Using less energy tends towards reducing the demand on available energy resources.

According to the graph above, 61.25 percent of participants "strongly agree" that using less energy tends towards reducing the demand on available energy resources. On the other hand, 35 percent of respondents "agreed" with the preceding idea. Moreover, 3.75 percent "strongly disagreed," that using less energy tends towards reducing the demand on available energy resources. The following graph is focused on assessing whether sustainability dimension enables towards thinking about future considering environmental, societal and economic aspects:

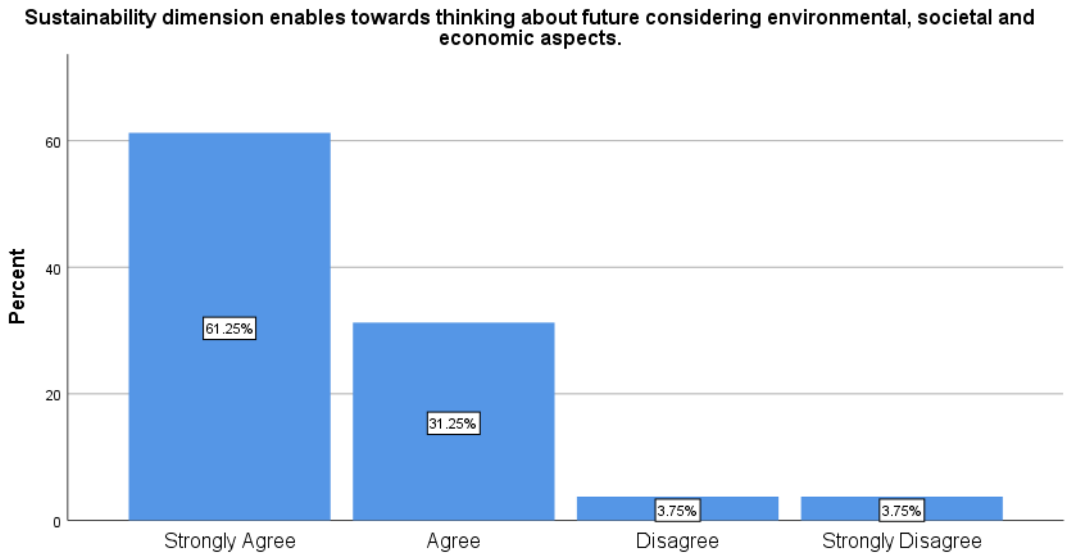

Figure 4.10: Sustainability dimension enables towards thinking about future considering environmental, societal and economic aspects.

It can be observed from the above illustration that 61.25 percent of participants reported that they "strongly agree" that sustainability dimension enables towards thinking about future considering environmental, societal and economic aspects. On the contrary, 31.25 percent of participants "agreed" with the prior notion. However, 3.75 percent of participants "disagreed" and 3.75 percent "strongly disagreed" that sustainability dimension enables towards thinking about future considering environmental, societal and economic aspects. The above results are in conjunction with the findings of systematic review, which uncovered that sustainability dimension as effective sustainability indicator of software development. The following graph is centred on examining if sustainability dimension aids in balancing in pursuit of an improved life quality:

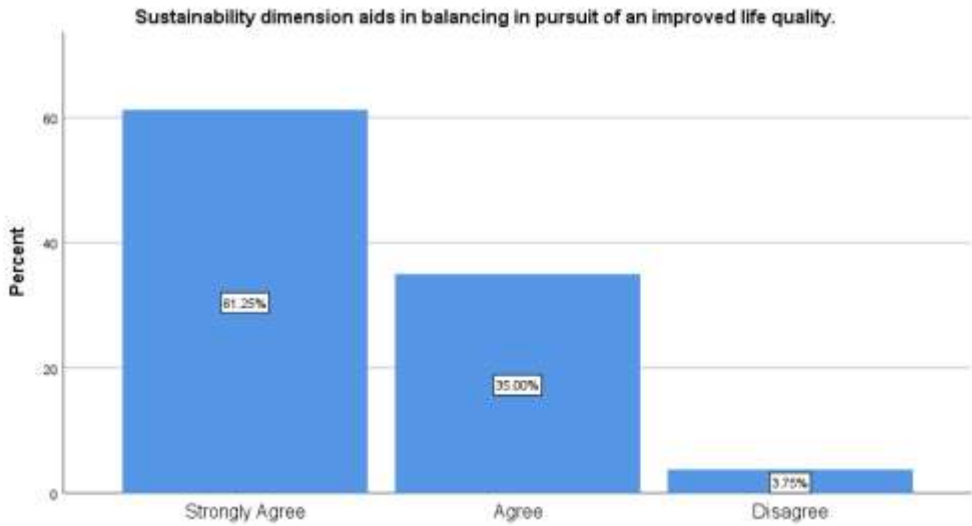

Figure 4.11: Sustainability dimension aids in balancing in pursuit of an improved life quality.

Considering to the above depiction, 61.25 percent of participants "strongly agree" that sustainability dimension aids in balancing in pursuit of an improved life quality. On the other hand, 35 percent of respondents "agreed" with the preceding idea. Moreover, 3.75 percent of respondents "disagreed" that the sustainability dimension aids in balancing in pursuit of an improved life quality. The following graph is focused on assessing whether sustainability dimension aids in obtaining all their needs and aspirations through different modalities:

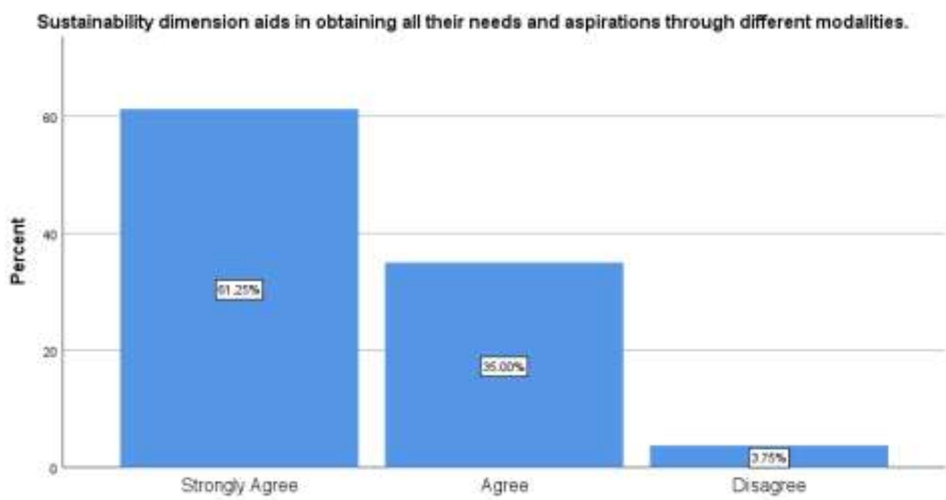

Figure 4.12: Sustainability dimension aids in obtaining all their needs and aspirations through different modalities.

It can be observed from the above chart that 61.25 percent of participants reported that they "strongly agree" that sustainability dimension aids in obtaining all their needs and aspirations through different modalities. On the contrary, 35 percent of participants "agreed" with the prior notion. However, 3.75 percent of participants "disagreed" that sustainability dimension aids in obtaining all their needs and aspirations through different modalities. The following graph is centred on examining if it is the set of instructions or programs that directs the computer in terms of what to do:

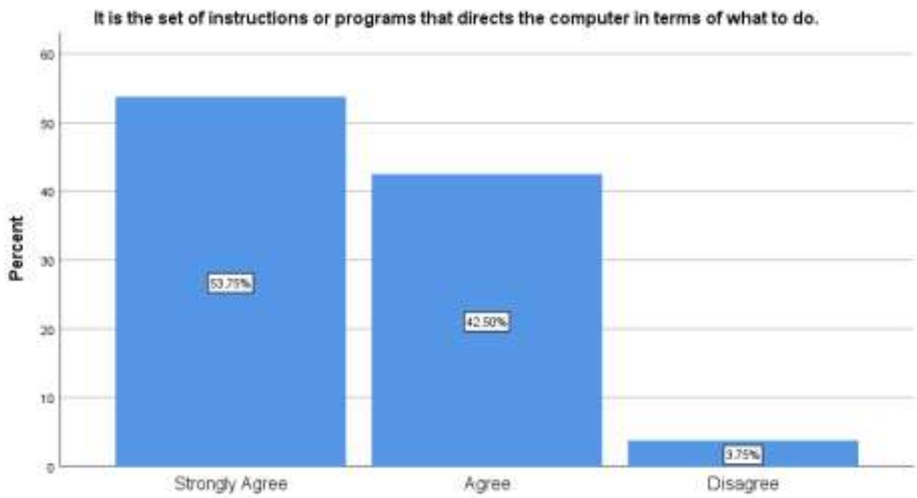

Figure 4.13: It is the set of instructions or programs that directs the computer in terms of what to do.

As mentioned in the above depiction, 53.75 percent of participants reported they "strongly agree" that it is the set of instructions or programs that directs the computer in terms of what to do. On the other hand, 42.50 percent of participants "agreed" with the prior notion. However, 3.75 percent of participants "disagreed" that it is the set of instructions or programs that directs the computer in terms of what to do. The following graph is centred on examining if software development helps in building software systems that power networks and devices and make sure that those systems remain functional:

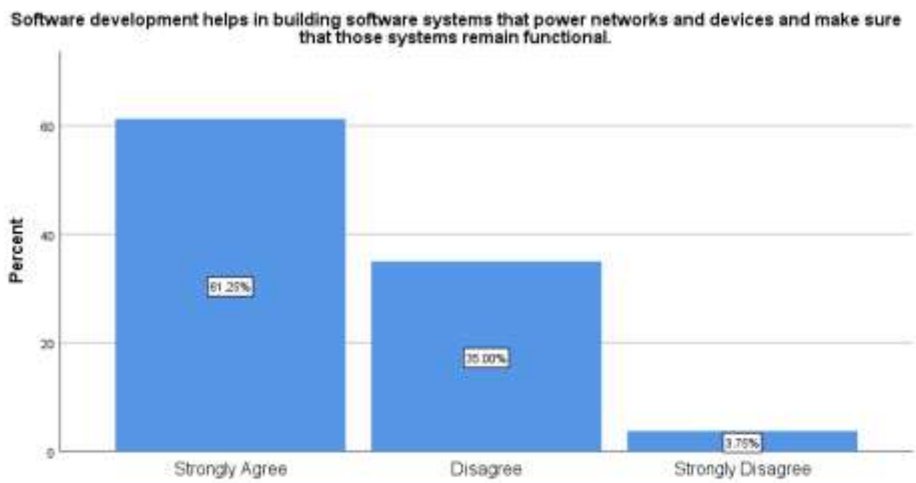

Figure 4.14: Software development helps in building software systems that power networks and devices and make sure that those systems remain functional

According to the graph above, 61.25 percent of participants "strongly agree" that software development helps in building software systems that power networks and devices and make sure that those systems remain functional. Nonetheless, 35 percent of respondents "disagreed," and 3.75 percent "strongly disagreed," that software development helps in building software systems that power networks and devices and make sure that those systems remain functional. The following graph is focused on assessing whether it helps in attaining wide knowledge of development and design along with providing particular solutions:

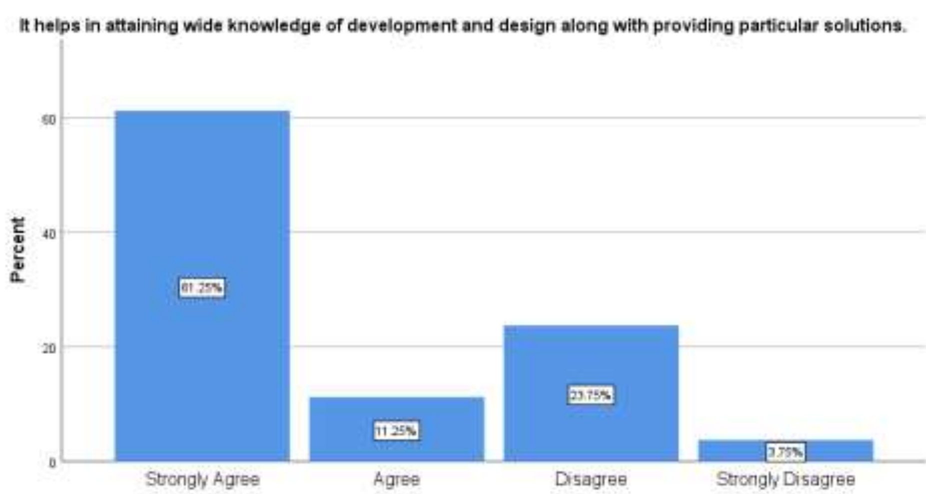

Figure 4.15: It helps in attaining wide knowledge of development and design along with providing particular solutions.

It can be seen from the above illustration that 61.25 percent of participants reported that they "strongly agree" that it helps in attaining wide knowledge of development and design along with providing particular solutions. On the contrary, 11.25 percent of participants "agreed" with the prior notion. However, 23.75 percent of participants "disagreed" and 3.75 percent "strongly disagreed" that it helps in attaining wide knowledge of development and design along with providing particular solutions.

4.2.3 Correlation Analysis

The Pearson correlation, which is represented by the symbol r, is a statistical tool used to assess the degree of relationship between two variables. When analysing the Pearson coefficient, it is assumed that the magnitude of the correlation ranges between 0 and 1 [76]. Weak, moderate, and strong associations are the three types of associations that might exist between parameters. It has been claimed that there is only a weak link between the variables when the value of correlation is between 0 and 0.3 [77]. The prior source added that that there is a moderate link between the variables when the correlation value falls between 0.3 and 7. Finally, a substantial link between the variables is seen to exist when the Pearson correlation value ranges between 0.7 and 1.

In this study, the independent variables are green software, software sustainability considerations, Lower energy consumption, and sustainability dimension whose link is measured with software development, which is the only dependent variable in this study. The findings of the correlation test are presented in tabular form as below:

Table 4.4: Correlation Analysis

Correlations				
		Green Software	Software Sustainability considerations	Lower energy consumption
Green Software	Pearson Correlation	1	.161	.162
	Sig. (2-tailed)		.153	.151
	N	80	80	80
Software Sustainability considerations	Pearson Correlation	.161	1	.952**
	Sig. (2-tailed)	.153		.000
	N	80	80	80

Lower energy consumption	Pearson Correlation	.162	.952**	1
	Sig. (2-tailed)	.151	.000	
	N	80	80	80
Sustainability Dimension	Pearson Correlation	.241*	.950**	.968**
	Sig. (2-tailed)	.031	.000	.000
	N	80	80	80
Software Development	Pearson Correlation	.081	.976**	.976**
	Sig. (2-tailed)	.474	.000	.000
	N	80	80	80

		Sustainability Dimension	Software Development
Green Software	Pearson Correlation	.241*	.081
	Sig. (2-tailed)	.031	.474
	N	80	80
Software Sustainability considerations	Pearson Correlation	.950**	.976**
	Sig. (2-tailed)	.000	.000
	N	80	80
Lower energy consumption	Pearson Correlation	.968**	.976**
	Sig. (2-tailed)	.000	.000
	N	80	80
Sustainability Dimension	Pearson Correlation	1	.926**
	Sig. (2-tailed)		.000
	N	80	80
Software Development	Pearson Correlation	.926**	1
	Sig. (2-tailed)	.000	
	N	80	80

According to the table, the Pearson correlation coefficient for the association between green software and software development is .081, indicating a positive but weak correlation between the two variables. The dependent variable, software development, and software sustainability considerations have a significant positive link, as indicated by the Pearson coefficient's value of .976. Additionally, the correlation between Lower energy usage and Software Development is .976, demonstrating a positive and substantial association between the two variables. Finally, the Pearson correlation value of .926 was revealed to illustrate the association between Sustainability Dimension and Software Development, demonstrating the strong and positive link between these two variables. All of the parameters are significant since their significant values are all greater than 0.05. The single "*" value represents weak association between the variables while "**" represents strong significant relation between the factors.

4.2.4 Regression Analysis

To ascertain the link or intensity of the link that emerges between parameters, correlation analysis is utilised [78]. On the other hand, regression analysis is used to investigate the impact of an independent variable on the dependent variable. Model summary, ANOVA, and tables of coefficients are the three categories within which regression analysis is characterised [79].

Table 4.5: Model Summary

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.999[a]	.997	.997	.06390

The model summary table presented above is employed to anticipate the connection between the research's components as well as to identify any variation that may have been brought on by the independent variable's effects on the dependent variable. The recommended regression model effectively accounts for a sizeable portion of the total variance of the dependent variable, as shown by the R-square value of .997. Additionally, the adjusted R-squared value reveals that the independent factors are responsible for .997 percent of the variation in the dependent variable.

Table 4.6: ANOVA

ANOVA						
	Model	Sum of Squares	df	Mean Square	F	Sig.
1	Regression	103.737	4	25.934	6351.977	.000b
	Residual	.306	75	.004		
	Total	104.043	79			

The study's F-value, which is 6351.97 and quite high, shows that the model is suitable for purpose. Additionally, the significant level for the model used in this study is 0.000, which is lower than 0.05, indicating that it is significant.

Table 4.7: Coefficient

Coefficients						
		Unstandardized Coefficients		Standardized Coefficients		
	Model	B	Std. Error	Beta	t	Sig.
1	(Constant)	.026	.021		1.235	.221
	Green Software	-.045	.009	-.035	-5.201	.000
	Software Sustainability considerations	.811	.027	.678	30.515	.000
	Lower energy consumption	1.078	.035	.876	31.052	.000
	Sustainability Dimension	-.833	.043	-.558	-19.351	.000

The sig value of Green Software is .221, as can be seen from the coefficient table above. The resulting value is greater than 0.05, demonstrating that the Green Software has an insignificant impact on the Software Development, the dependent variable. Furthermore, Software Sustainability consideration has a significant value of .000, which is greater than the baseline threshold of 0.05, showing that it has a significant impact on Software Development. Thirdly, the significance value of Lower energy usage is .000, demonstrating that Software Development is significantly impacted by Lower Energy Consumption. Finally, the significance value of the sustainability dimension is .000, indicating that it significantly influences software development. To portray the link of above findings with study goals, it can be inferred that three sustainability indications (independent variables) such as software sustainability considerations, lower energy consumption, and sustainability dimension, have a significant impact on software development (dependent variable). However, only one independent variable i.e. green software has insignificant impact on software development.

4.3 Chapter Summary

This chapter was centred on presented detailed results using the technique of inferential analysis, whereby the findings were presented using demographic, frequency, correlation, and regression analysis. The overall results demonstrated that there is a significant association between green software, software sustainability considerations, lower energy consumption, sustainability dimension, and software development. In terms of results from regression analysis, it was unveiled that green software, software sustainability considerations, lower energy consumption, and sustainability dimension have a significant impact on software development.

CHAPTER 5

DISCUSSION

5.1 Introduction

In this part, the results that have been obtained from both primary and secondary research are analysed and discussed in depth. The results are discussed in this chapter in light of the previous chapter's literature review. The conclusion on the classification of sustainability indicators for software development is established based on the information presented in this chapter. Due to the fact that it provides an in-depth analysis of the primary results, this part has the potential to be regarded as one of the most significant elements of the whole research study.

5.2 Sustainability in software development

According to what was found in the research, sustainability is linked to striking an appropriate equilibrium between economic, social, and environmental factors in order to meet the needs of the current human population without jeopardising the availability of resources for future generations [14]. Both secondary and primary research have come to the conclusion that the concept of sustainability indicators has been created and that these indicators provide a suitable method for assessing the degree to which sustainable development is being achieved. When it comes to achieving sustainability in software development, a variety of frameworks and ideas have been put up as potential solutions. According to the findings of a recent study [15], the triple bottom line (TPL) is one of the frameworks that is used the most often. On the other hand, a second piece of the study claimed that integrated reporting and the SDGs framework are also valuable frameworks that could be taken into consideration in the software business and other fields [16].

According to the findings, the environmental aspect of sustainability in the field of software development refers to the impact that software-related activities have on the surrounding environment [16]. The key results examined the relevance of environmentally friendly software and its contribution to ensuring the highest possible level of energy efficiency.

Software that is considered sustainable or environmentally friendly is written and operated in a way that ensures a high level of energy efficiency and has either no or a minimum impact on the surrounding environment [44]. It is a new discipline that ties together the architecture of software development with the practises of the industry. According to the research that was conducted, sustainability also takes into account the impact that software development has on energy consumption, climate change, and the production of trash [17].

According to the findings of the main study, eco-friendly software may be regarded as the fundamental component of nearly any intelligent strategy produced with the intention of assisting the environment [44]. According to the published research, green software places a significant amount of importance on making a significant contribution to the preservation of the natural world. Because it is a new field, software corporations and other large IT businesses are not yet particularly interested in it [51]. Green software, on the other hand, may contribute to the preservation of the natural world and make use of algorithms that are gentler on the environment.

According to the research that was conducted, even though the social aspect of sustainable software development places a significant amount of emphasis on the impact that software-development initiatives have on society, which includes the impact that software development has on human rights, social inclusion, and social cohesion [45], this aspect of software development still prioritises its own sustainability. According to the research that was conducted, the economic component of sustainability relates to the impact that the creation of software has on the economy as a whole. It entails the impact that the creation of software has on employment rates, economic expansion, and the reduction of levels of poverty. The sustainability perspective makes it possible to think about the future by taking into account characteristics of the environment, society, and economy.

In terms of sustainability, the findings revealed that the utmost emphasis had been placed on the effort to fight against pollution issues, particularly the efforts of the EU. This is because the EU has restricted the amount of greenhouse gas emissions that can be produced by each country, and these restrictions can be traded in the form of emission permits [53]. According to the findings of a research, installation should be carried out in such a way that customers may keep their credit but are unable to go over the limit that has been established by the European Union [58]. They are required to make a payment in order to be eligible for the credit if the installations have the potential to cause the emission of a significant quantity of gases. Because of this, the research has shown that it is vital for them to keep a careful eye on their environmental and carbon footprint in order to prevent the emission of any potentially harmful compounds into the environment.

According to the findings of the main study, the sustainability factor contributes to a more balanced approach to the improvement of life quality. The research that was conducted lent credence to the idea that an increase in job prospects was brought about by the rise in the information technology industry [16]. It has ultimately led to an improvement in people's quality of life and has been shown to be helpful for the economy as a whole. The fact that software makes things simpler for humans is another factor to consider. The results also showed that the development of smartphones, the implementation of programming in medical equipment, the creation of automated systems via programming, and the creation of large software systems have all contributed to an increase in the quality of life [63].

According to the research that was conducted, the most important factors to consider when determining the long-term viability of software are its usability, its functional appropriateness, its efficiency, its dependability, its flexibility, and its manageability [18]. According to the research that was conducted, the functional adequacy of software emphasises its capacity to successfully carry out the required activities. In order for software to be deemed sustainable, it is essential that it be able to carry out each and every required function in an efficient and effective manner. According to the findings, in order for software to be sustainable, it must have characteristics such as dependability, efficiency, manageability, and usability [18]. These components suggest that the programme may be used for an extended period of time without becoming obsolete. Nevertheless, the researchers also claimed that additional criteria are defined by the specialists who critically examine the software's longevity. The research also suggested that the long-term viability of software is dependent on a variety of factors, and that software makers should make sure their products have a significant effect on end users.

5.3 Important Sustainability Indicators in software development

According to the available research, there are many distinct types of sustainability indicators for the software development industry. These include sustainability dimensions, software sustainability factors, green software, low energy usage, and sustainability considerations. A green software that results in reduced emission of gases and assures optimum energy efficiency is the primary measure of sustainability [55]. According to the results, a piece of software may be termed sustainable if it has the qualities that are typical of environmentally friendly software. The green elements bring attention to the notion that software should centre itself on satisfying its criteria or attributes. The literature went on to explain that progress in the software engineering systems of QA is that these strategies are being developed to fulfil the needs of future research in the development of sustainability metrics for green software as well as the techniques that may evaluate them [56]. This was further explained by the fact that these strategies are being developed to fulfil the needs of future research in the development of green software. Calculating and achieving software system-wide energy awareness may be done with the help of green metrics, which provide a framework for the environmentally responsible and sustainable practise of software engineering. These metrics are referred to as green performance indicators. According to the findings of a research [57], the concept that green metrics are an essential approach that may assist in assuring the environmental, social, technological, and economic factors is upheld. These measurements may be taken into consideration by the developers while they work on the required software, and the product's features can be based entirely on the metrics themselves.

According to the available research, one of the most important fundamental sustainability indicators of sustainable software is its low energy usage. [47] Vaio cited an example in which the software installed in the refrigerator was designed to reduce the amount of energy that was wasted. Another method for reducing overall energy use is the use of mobile apps for regulating air conditioners. The data also shown that the green software systems may be regarded in terms of their effect due to the fact that these systems make use of low-energy resources and support certain policies, both of which contribute to the process of raising awareness about green software [45]. The study went on to clarify that popular software systems such as Granole, Auto shut down and power, and CO2 save are the types of software that assist in ensuring energy efficiency and minimise excessive energy usage [44]. There are often a variety of activities and recommendations on the procedures for green software development [54]. Green software development is characterised by its capacity to make use of less energy, also known as energy efficiency, and its ability to generate a small amount of waste. Brovkina continued by arguing that there are techniques to increase the energy efficiency of software on systems that have many cores [62]. The researchers proposed that one of the primary drivers of motivation should be the significant influence that information and communication technology has had on global CO2 emissions [62]. The results also showed that being able to anticipate the future in terms of the interplay between environmental, social, and economic factors is facilitated by using sustainable software. It is also feasible to use it to make predictions about the probable effects that sustainability will have on the future in terms of social, environmental, and economic factors.

Additionally, the term "sustainable dimensions" refers to an essential sustainability indicator of software development. The aspects consist of human, societal, environmental, economic, and technological considerations [16]. According to the results, developing environmentally friendly software should prioritise having a good impact on the economy. It is possible that it may offer job chances for a number of different people, which will ultimately have a good effect on the economy as a whole [15]. In a similar vein, Schon said that the primary objective of developing sustainable software is to improve the economic circumstances of a nation and a society. The researcher went on to clarify that environmentally friendly software should have an emphasis on lowering emissions of greenhouse gases and carbon dioxide, both of which have a favourable impact on the environment [59]. It demonstrates the environmentally beneficial characteristics that are linked with using sustainable software systems. The use of software such as "CO2 save" is very important in the process of making environmental circumstances better [18]. According to the published research, eco-friendly software solutions have the potential to contribute significantly to big environmental transformations. Writing software in such a manner that the hardware uses less power is one of the techniques, and it is one of the approaches that is being considered [16]. Mobile apps may also play a significant part in the efficient use of energy by playing a function that includes delivering notifications to users and monitoring the temperature as well as the usage of various electrical equipment.

According to the results, one of the most important sustainability indicators of sustainable software is whether or not it takes into account environmental factors. According to the research that was conducted, the stakeholders are the ones who are responsible for ensuring that software is sustainable [14]. This is because they must adapt to the rapidly changing circumstances of the environment and the economy. In addition, the findings demonstrated that the creation of software plays a significant part in the production of rebounding effects. The answer that the software developers have come up with is to double the amount of storage and processing power that is available at the given price in order to get more of the same [12]. The research that was done on the topic provided support for the idea that while assessing software, one should give primary consideration to how it would affect society, the economy, and the environment as a whole [18]. The findings also showed that environmentally friendly software is an essential component which is a significant predictor of environmental friendliness. Whenever stakeholders do an analysis of a piece of software, they should keep in mind how that programme may reduce its negative effects on the surrounding environment. According to the research that's been done, indications of software durability are the ones that matter the most throughout the implementation process [25]. The fundamental emphasis should be on minimising the minimal amount of power used, lowering the amount of gases emitted, giving chances for employment, and creating a sustainable environment for the people of the community.

5.4 Taxonomy of sustainability indicators for software development

The findings revealed that TPL refers to a specific framework for thinking about sustainability that includes three key dimensions: environmental, social, and economic. This particular framework is significant to evaluate how sustainability indicators may be utilised for tracking the performance of software development ecosystems [15]. Simultaneously, the sustainability indicators can be categorised based on the taxonomy used for tracking and measuring the performance of systems related to sustainability. The findings revealed that based on the framework, the taxonomy can be divided further into economic, environmental, and social elements [63]. Considering the economic aspects, software must have a strong emphasis on reducing emissions, improving the environment, and having a profound impact on the environment. The literature backed the findings as it explained that green software is suitable in this case that ensures maximum energy efficiency and emitting lower greenhouse gases [62].

The literature and primary results explained that green software processes are those that meet the demands of the present environment and situations as they are designed in such a way [56]. Considering the environmental aspects, the literature revealed that software systems are required to build such an endurable system that is not harmful to the ecosystem. The literature supported the notion that installation must be done in a way that they can hold the credit but not able to exceed the limit of the cap set by the EU [55]. Considering the social aspects, the sustainability of the global system of a software system involves the sustainability of the software ecosystems, and they cover the different aspects such as various sub-systems from the biggest interconnected system and all the interactions. The technical aspect is another dimension that has been considered in the taxonomy for sustainable indicators [53]. It includes the overall features, functionality, usability, and reliability of the software. This aspect generally fascinates the users and is specifically related to the internal use of the software. It does not impact the external elements and hence, is limited to the users' experience only [62].

Primary and secondary research revealed that energy efficiency is another environmental aspect of sustainable software. A study argued that green software engineering efficiently builds software that they are capable of consuming a lower amount of energy [56]. The model related to green software can demonstrate the categories of criteria for sustainability and metrics for the products of software. Energy efficiency can also be linked to social aspects as it positively affects society [62]. Consumers can save on electricity bills and learn how to consume energy efficiently through the use of sustainable software [55]. Based on the taxonomy, social factors cannot be neglected specifically in the case of green software as it eventually provides benefit to society as a whole. The literature further explained the economic benefits associated with sustainable software. It is a major consideration that should be addressed when designing software applications [47].

The literature analysis revealed that employment opportunities, contribution to GDP, and exports are the major economic elements associated with sustainable software. The primary and secondary research findings explained that employment opportunities could be created through sustainable software as applications like Uber and Foodpanda have provided jobs to several individuals worldwide [43]. The literature further claimed that sustainable software could contribute to the GDP to a considerable degree. Uber significantly contributes to the GDP of different countries and it is a prime example of economic benefits associated with sustainable software [21]. The findings also revealed that exports could be increased through sustainable software. Revolutionary ERP systems and software like SAP increases the export of a country to a considerable degree [18]. Studies also claimed that different countries are relying on IT exports for supporting the GDP and economy as a whole [43]. It is a prime example, which highlights that the economic aspect is a major element in the taxonomy of sustainable indicators.

Based on the results, it is observed that the taxonomy of sustainability indicators for software development is significant because it provides a way to measure and compare the sustainability of different software development processes [3]. The taxonomy can be used to identify best practices and make recommendations for improving the sustainability of software development. The findings further revealed that taxonomy could be considered by software developers when developing sustainable software [4]. It can help in critically assessing both the strengths and weaknesses of the software systems, eventually allowing the developers to make possible recommendations in the software systems.

5.5 Gaps in the existing research on sustainability in software development

The requirement for ensuring the software ecosystem's sustainability is discussed widely in the literature. Nonetheless, there is insufficient systematic and comprehensive taxonomy of the sustainability indicators for software development. A study explained that the taxonomy is significant as it delivers a standardised and structured way of measuring software development's sustainability [19]. Nonetheless, it is observed that only a little research has been carried out on the evaluation of the environmental influence of software development. There is also a requirement for developing an evaluation framework for software development's sustainability [18]. This can assist in enhancing the understanding of software development and its influence on the environment and also assists in assessing the requirements that should be addressed for enhancing software development sustainability.

Despite the need to have sustainability in almost every domain, the software lacks behind when it comes to ensuring sustainability. Only a limited number of researchers have talked about the need to have a proper framework for assessing the sustainability of software development. There is literature available regarding green software; however, researchers have not focused their attention on the taxonomy of sustainability indicators [64]. The gap also exists in the case of sustainability indicators which is a major gap to fill. Researchers have talked about sustainability; however, they did not talk about what possible indicators can define sustainability when it comes to software development.

Software engineers generally focus their attention on usability and functionality; however, most of them do not have a strong emphasis on having an impact on the environment and economy. The revolutionary software has been able to generate employment for different individuals [65]. The example of Uber is evident that emerged merely as software and provided employment opportunities to various individuals. In this case, sustainability is of significant importance since it should be discussed in the literature [66]. Also, the authors and researchers need to teach the software developers about the importance of sustainability and its vast definition.

Another gap that has been observed is that green software is not being implemented at a high scale. Despite the presence of literature, the researchers have not been successful in communicating the maximum benefits of green software to the readers and experts [67]. This gap has restricted the developers to have a strong emphasis on developing green software. Most companies develop software merely for profitability and reputation; however, only a little number of firms focus on developing sustainable and green software that positively affects the economy and environment. Researchers need to contribute to the literature regarding the taxonomy of sustainability indicators for software development [8]. It can eventually teach the experts and professionals how the sustainability indicators should be categorised.

The findings also revealed that the concept of energy consumption can be linked to software development [68]. The literature should further discuss how green software can help in lowering energy consumption and how it can eventually lead to energy efficiency. Currently, only a limited number of studies can be found that discuss the relationship between software development and energy consumption [69]. Hence, this gap needs to be filled, which can bring revolutionary changes in the software industry and overall environment.

Researchers should also conduct studies in different regions of the world regarding sustainability in software development. In the EU, there have been cases of high GHG and hence, the literature should discuss that concerning software sustainability. However, other countries may have other environmental and economic issues that can be resolved through sustainable software development. Hence, this gap should be filled so that the maximum number of stakeholders and developers can gain benefit from the literature. It can eventually assist the economy because software prevails now in almost every sector.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Introduction

Notably, the primary purpose of this research study is to provide a summary of the findings of the study that was conducted in order to develop the taxonomy of sustainability indicators for software development. Additionally, this research study will provide an analysis of how sustainability indicators that are of significant importance will be presented, as well as an evaluation of the process of developing the taxonomy of sustainability indicators for software development. In addition to that, the purpose of this part of the research is to provide some suggestions while also providing an analysis of the practical consequences of the study.

6.2 Findings

Based on the above analysis, it has been found that the heightened awareness of the degree to which human activities can be maintained has come about as a direct consequence of the rise of environmental and social concerns. The industry of software development is not an exception, and there has recently been an increase in the number of studies that focus on environmentally responsible practises in the software development process (SSD). Despite this, there is still a lack of understanding on the specifics of SSD, including what it is and how it may be achieved. This study is looking at the problem of the lack of a comprehensive taxonomy of sustainability measures for software development as a possible solution. The problem that is being looked at is the absence of a complete taxonomy. In addition, as people in every region of the globe become more aware of the significance of sustainability, there is an increasing need for a taxonomy of sustainability indicators that is particularly adapted to the process of developing software. The number of problems that crop up at various stages of the software development process has greatly increased, which has led to an increase in the number of individuals who are interested in sustainability indicator.

Human life is balanced by sustainability. It is seen as a method for using products, services, and other resources in a way that minimises damage to living beings. This specifies that software systems must build a resilient system that will not harm an ecosystem in the present or future. They must also satisfy current and future users with little environmental impact. They must simultaneously advance corporate development and societal values. The sustainability of software is influenced by economic, social, personal, technical, and environmental aspects. Software engineering lacks a distinct understanding of the categories. Today's software sustainability encompasses sustainability in software development, software for sustainability, green software systems, sustainable software ecosystems, and other related concepts. Procedures are required for eco-friendly software development. Software for sustainability refers to the manner in which software promotes sustainability. Software for refrigerators conserves electricity. Green software systems save energy and promote awareness campaigns. Lastly, the impact of software ecosystems on the larger software ecosystem might be indicative of their longevity. Recently, researchers have concentrated on software durability. Advanced study focuses on green software systems and the appropriateness of software development.

Sustainable software development is a term that refers to the practise of incorporating procedures into the software development process. In addition, the term "software for sustainability" refers to the use of software in a manner that advances sustainable practises. For example, the software in a refrigerator is designed to decrease the amount of energy used. It has also been discovered that green software systems may be assessed in terms of their effects. This is due to the fact that green software systems use less energy resources and encourage guidelines that assist create awareness about green software. In conclusion, the viability of software ecosystems may be evaluated by assessing the effect that they have on the whole software ecosystem. Over the last several years, many industry professionals have shown a growing interest in the idea of software sustainability. However, in terms of views for more study, green software systems and software development appropriateness are often the ones that get the greatest focus.

The outcomes of the research indicate that the development of environmentally friendly software is feasible. Green features put a focus on the requirements of the programme. According to the published study, software engineering quality assurance approaches are now being developed in order to satisfy the needs of future research. Green metrics allow for the measurement of sustainability as well as an awareness of energy use in software engineering. Green metrics.

Green metrics might be a useful tool for safeguarding the health of the environment, society, technology, and the economy. Programmers have the ability to design software by using these criteria.

According to the findings of the study, a decrease in the amount of energy used is another sign of sustainable software. Sustainability is one of the indicators of successful software development. Individual, ecological, economic, and technical factors [16]. According to the findings of study, environmentally responsible software provides financial benefits. It is possible that employment would be created, which would be good for the economy. The outcomes of the research indicate that environmentally responsible software is beneficial to the economy. A more sustainable software should produce less greenhouse gas and carbon dioxide emissions. It places a focus on software that is favourable to the environment. A "reduction in CO2" is beneficial to the health of the ecosystem.

The findings of the research study suggest that environmentally friendly software may be beneficial to the world's natural resources and ecosystems. The usage of energy might be reduced via software. Mobile applications might potentially notify consumers and regulate electrical appliances in order to conserve energy. In addition, the outcomes of the research suggest that sustainability may accurately predict sustainable software. The promotion of software's sustainability by its stakeholders is necessary because of shifts in the economic and environmental variables. The findings of the investigation showed that the development of software had an influence on rebounding. Software developers improve both the storage and processing capacities of computer programmes. The conclusions that were gathered from the research provide evidence for evaluating software based on its effects on the economy, society, and the environment. According to the results, a sign of sustainability is the use of environmentally friendly software. The impact of the software on the environment should be included into the assessment.

On the other hand, research conducted in this field and the most important findings indicated that the term "green software processes" should be defined as procedures that are created to suit the requirements of the existing environment and the circumstances that are already in place [56]. Research that has been made public has shown that the use of software systems is essential in order to create a system that will persist for a long time and will not affect the natural environment. This discovery was made in consideration of ecological factors. The research that was carried out provided support for the idea that the installation needed to be carried out in such a manner that the credit could be maintained without going over the limit that had been specified by the EU. When social issues are taken into consideration, the sustainability of the software ecosystems is required for the global system of a software system to be sustainable. These ecosystems are made up of a variety of components, including the subsystems that are a part of the larger connected system as well as all of the interactions that take place within those components. The technological component was one of the aspects that were taken into consideration throughout the process of developing the taxonomy for sustainable indicators. It takes into account the characteristics of the application as a whole, such as its functions, usability, and dependability for users. The functionality of this feature, which is closely connected to the internal workings of the programme, is of particular importance to the users. Since it is unable to affect the environment around it in any way, the only thing that can be affected by it is the experience of the people who use it.

According to the findings of both primary and secondary research, energy efficiency is another aspect of the environment that contributes to the creation of sustainable software. According to one piece of research, the process of designing programmes that are efficient in their use of energy is referred to as "green software engineering." The model for ecologically responsible software is able to provide the categories of metrics and sustainability requirements that are relevant to software products. Because increasing energy efficiency has a beneficial impact on society, there is a connection between social issues and the pursuit of greater energy efficiency. Customers have the opportunity to discover how to reduce their overall energy consumption and save money on their monthly power bills if they utilise environmentally friendly software. According to the taxonomy, social considerations cannot be ignored, and this is especially true for software that is favourable to the environment, which ultimately benefits society as a whole. The findings of the research study provided more data demonstrating that environmentally responsible software offers a variety of financial benefits. The most important economic variables connected to sustainable software are the employment opportunities, contributions to the gross domestic product, and exports that may be generated. According to the results of both primary and secondary research, the availability of work prospects may be facilitated by environmentally friendly software.

6.3 Recommendations

This section of the research study is mainly indebted to give recommendations, which are derived from the findings of the study

- At first, it is recommended that a proper comprehensive research should be done with the primary motive to make sure that the suitable software and software indicators are applied.
- It is also recommended that significant value and consideration should be given to the paradigm of green software system as it will mainly assist the corporations to develop eco-friendly system, which is cost-effective as well.
- Lastly, it is also recommended that proper training and development programme should be instigated in which significant consideration must be given to make sure that all the stakeholders learn and classify the right way and approach to implicate the software system righteously.

6.4 Implications of Study

Since it gives comprehensive information on the taxonomy of sustainability indicators for software development, it has been presumed that this research has a broad range of practical implications. The reason for this assumption is that it offers detailed data about the taxonomy. Organizations have the capability of using this taxonomy of sustainability indicators for software development in order to monitor and compare the environmental impact of the several software development processes. The taxonomy may be used to find best practises and provide ideas for improving the software development process in a manner that is more environmentally friendly. The research study is highly significant for this reason, and its significance may be interpreted in a number of various ways. To get things started, the taxonomy could be of use to organisations in better appreciating the degree to which the sustainability of their software development processes. Second, companies may utilise the taxonomy to improve the long-term sustainability of their software development processes by integrating it into their business operations. This can be accomplished with the help of the taxonomy.

6.5 SWOT Analysis

The following is the strengths, weaknesses, opportunities and threats (SWOT) analysis for the current study, which provides a comprehensive overview of the strengths of the research and potential weaknesses and threats to the research's validity and reliability. It also offers an overview of the potential opportunities and areas of further research.

Strengths: A thorough explanation of the taxonomy of sustainability criteria for software development is provided by the research. This taxonomy of sustainability criteria for software development can be used by organisations to track and assess the ecological impact of various software development methods.

Weaknesses: The research's weakness is that it offers a comprehensive view of the taxonomy of sustainability in software development. However, by concentrating on specific initiatives and case studies that might offer a more realistic and valuable plausibility of taxonomic sustainability, a more focused study could be conducted. The fact that this research leans on respondent perspective and lacks a specific real-world situation or useful illustration to balance the benefits and drawbacks of taxonomic sustainability in software development is a shortcoming of the study.

Opportunities: Companies may find the taxonomy useful in better understanding how sustainable their processes for developing software are. By incorporating the taxonomy into their daily business activities, businesses can use it to enhance the long-term sustainability of their software development practices. The taxonomy will be useful in doing this. Additionally, thorough research can be conducted with the main goal of ensuring that the right software and software characteristics are used. There is also a chance to place more attention on the sustainable software concept, which will primarily help businesses create eco-friendly systems that are also economical.

Threats: Researcher has made an effort to reduce the various threats to authenticity by taking a variety of mitigating measures. There might be researcher bias, but by clearly defining the research objectives, exclusion and inclusion criteria, and the justification for doing the search, researcher minimised the consequences of such a bias. Furthermore, the initial search string had an excessive number of pointless documents. For each source, the search term was adjusted as a result.

REFERENCES

[1] Salam, M. and Khan, S.U. Challenges in the development of green and sustainable software for software multisourcing vendors: Findings from a systematic literature review and industrial survey. Journal of Software: Evolution and Process, 30(8), p.e1939, 2018.

[2] Mourão, B.C., Karita, L. and do Carmo Machado, I, October. Green and sustainable software engineering-a systematic mapping study. In Proceedings of the 17th Brazilian Symposium on Software Quality (pp. 121-130), 2018.

[3] Calero, C., Mancebo, J., García, F., Moraga, M.Á., Berná, J.A.G., Fernández-Alemán, J.L. and Toval, A. 5Ws of green and sustainable software. Tsinghua Science and Technology, 25(3), pp.401-414, 2019.

[4] Iatsyshyn, A., Iatsyshyn, A., Artemchuk, V., Kameneva, I., Kovach, V. and Popov, O. Software tools for tasks of sustainable development of environmental problems: peculiarities of programming and implementation in the specialists' preparation. In E3S Web of Conferences (Vol. 166, p. 01001). EDP Sciences, 2020.

[5] Torre, D., Procaccianti, G., Fucci, D., Lutovac, S. and Scanniello, G. On the presence of green and sustainable software engineering in higher education curricula, 2017. In 2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM) (pp. 54-60). IEEE.

[6] Tamburri, D.A. Sustainable mlops: Trends and challenges, 2020, September. In 2020 22nd international symposium on symbolic and numeric algorithms for scientific computing (SYNASC) (pp. 17-23). IEEE.

[7] Naseer, M., Zhang, W. and Zhu, W. Early prediction of a team performance in the initial assessment phases of a software project for sustainable software engineering education. Sustainability, 12(11), p.4663, , 2020.

[8] Anand, J.V. Design and development of secure and sustainable software defined networks. Journal of Ubiquitous Computing and Communication Technologies (UCCT), 1(02), pp.110-120, 2019.

[9] Koniukhov, S. and Osadcha, K. Implementation of education for sustainable development principles in the training of future software engineers, 2020

[10] Symonenko, S. Complementing content of English courses for enhancing communication of IT-professionals for sustainable development., 2020

[11] Sharifi, A. and Allam, Z., 2022. On the taxonomy of smart city indicators and their alignment with sustainability and resilience. Environment and Planning B: Urban Analytics and City Science, 49(5), pp.1536-1555.

[12] Olteanu, Y. and Fichter, K., 2022. Startups as sustainability transformers: A new empirically derived taxonomy and its policy implications. Business Strategy and the Environment.

[13] Poth, A. and Nunweiler, E., 2022, January. Develop Sustainable Software with a Lean ISO 14001 Setup Facilitated by the efiS® Framework. In International Conference on Lean and Agile Software Development (pp. 96-115). Springer, Cham.

[14] Beerbaum, D.O., 2022. Development of a Sustainability Taxonomy for investor decision usefulness–Reflection of SEC and ISSB climate-related disclosures. Available at SSRN.

[15] Verdejo, Á., Espinilla, M., López, J.L. and Melguizo, F.J., 2022. Assessment of sustainable development objectives in Smart Labs: technology and sustainability at the service of society. Sustainable Cities and Society, 77, p.103559.

[16] Bharany, S., Badotra, S., Sharma, S., Rani, S., Alazab, M., Jhaveri, R.H. and Gadekallu, T.R., 2022. Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy. Sustainable Energy Technologies and Assessments, 53, p.102613.

[17] Zada, I., Shahzad, S., Ali, S. and Mehmood, R.M., 2022. OntoSuSD: Software engineering approaches integration ontology for sustainable software development. Software: Practice and Experience.

[18] Gill, S.S., Kumar, A., Singh, H., Singh, M., Kaur, K., Usman, M. and Buyya, R., 2022. Quantum computing: A taxonomy, systematic review and future directions. Software: Practice and Experience, 52(1), pp.66-114.

[19] Park, S.M. and Kim, Y.G., 2022. A Metaverse: Taxonomy, components, applications, and open challenges. Ieee Access, 10, pp.4209-425

[20] Hamdi, M., 2022. Towards a classification of sustainable software development process using manifold machine learning techniques. Journal of Intelligent & Fuzzy Systems, (Preprint), pp.1-12.

[21] Hustad, E. and Olsen, D.H., 2021. Creating a sustainable digital infrastructure: The role of service-oriented architecture. Procedia Computer Science, 181, pp.597-604.

[22] Batkovskiy, A.M., Kurennykh, A.E., Semenova, E.G., Sudakov, V.A., Fomina, A.V. and Balashov, V.M., 2019. Sustainable project management for multi-agent development of enterprise information systems. Entrepreneurship and Sustainability Issues, 7(1), p.278.

[23] Durmanov, A., Bartosova, V., Drobyazko, S., Melnyk, O. and Fillipov, V., 2019. Mechanism to ensure sustainable development of enterprises in the information space. Entrepreneurship and Sustainability Issues, 7(2), p.1377.

[24] Jones, P., Wynn, M., Hillier, D. and Comfort, D., 2017. The sustainable development goals and information and communication technologies. Indonesian Journal of Sustainability Accounting and Management, 1(1), pp.1-15.

[25] Nwankwo, W. and Njoku, C., 2020. Sustainable development in developing societies: the place of ICT-driven computer education. International Journal of Emerging Technologies in Learning (iJET), 15(12), pp.290-297.

[26] Hürlimann, C., 2019. Research Philosophy and Ethics. In Valuation of Renewable Energy Investments (pp. 111-126). Springer Gabler, Wiesbaden.

[27] Dougherty, M.R., Slevc, L.R. and Grand, J.A., 2019. Making research evaluation more transparent: Aligning research philosophy, institutional values, and reporting. Perspectives on Psychological Science, 14(3), pp.361-375.

[28] Kirongo, A. and Odoyo, C., 2020. Research Philosophy Design and Methodologies: A Systematic Review of Research Paradigms in Information Technology.

[29] Žukauskas, P., Vveinhardt, J. and Andriukaitienė, R., 2018. Philosophy and paradigm of scientific research. Management culture and corporate social responsibility, 121.

[30] Maarouf, H., 2019. Pragmatism as a supportive paradigm for the mixed research approach: Conceptualizing the ontological, epistemological, and axiological stances of pragmatism. International Business Research, 12(9), pp.1-12.

[31] Pandey, J., 2019. Deductive approach to content analysis. In Qualitative techniques for workplace data analysis (pp. 145-169). IGI Global.

[32] Bloomfield, J. and Fisher, M.J., 2019. Quantitative research design. Journal of the Australasian Rehabilitation Nurses Association, 22(2), pp.27-30.

[33]. Sileyew, K.J., 2019. Research design and methodology (pp. 1-12). Rijeka: IntechOpen.

[34] Asenahabi, B.M., 2019. Basics of research design: A guide to selecting appropriate research design. International Journal of Contemporary Applied Researches, 6(5), pp.76-89.

[35] Nelson, J., Berlin, A. and Menold, J., 2019, August. Archie: An automated data collection method for physical prototyping efforts in authentic design situations. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 59278, p. V007T06A044). American Society of Mechanical Engineers.

[36] Rahmatizadeh, S., Rajabifard, A., Kalantari, M. and Ho, S., 2018. A framework for selecting a fit-for-purpose data collection method in land administration. Land use policy, 70, pp.162-171.

[37] Ebert, J.F., Huibers, L., Christensen, B. and Christensen, M.B., 2018. or web-based questionnaire invitations as a method for data collection: cross-sectional comparative study of differences in response rate, completeness of data, and financial cost. Journal of medical Internet research, 20(1), p.e8353.

[38] Kyriazos, T.A., 2018. Applied psychometrics: sample size and sample power considerations in factor analysis (EFA, CFA) and SEM in general. Psychology, 9(08), p.2207.

[39] Amarasinghe, S.L., Su, S., Dong, X., Zappia, L., Ritchie, M.E. and Gouil, Q., 2020. Opportunities and challenges in long-read sequencing data analysis. Genome biology, 21(1), pp.1-16.

[40] Vasishth, S., Nicenboim, B., Beckman, M.E., Li, F. and Kong, E.J., 2018. Bayesian data analysis in the phonetic sciences: A tutorial introduction. Journal of phonetics, 71, pp.147-161.

[41] Bergin, T., 2018. An introduction to data analysis: Quantitative, qualitative and mixed methods. Sage.

[42] Arifin, S.R.M., 2018. Ethical considerations in qualitative study. International Journal of Care Scholars, 1(2), pp.30-33.

[43] Alves, C., de Oliveira, J.A.P. and Jansen, S., 2017. Software Ecosystems Governance-A Systematic Literature Review and Research Agenda. ICEIS (3), pp.215-226.

[44] Betz, S. and Caporale, T., 2014, December. Sustainable software system engineering. In 2014 IEEE Fourth International Conference on Big Data and Cloud Computing (pp. 612-619). IEEE.

[45] Caiado, R.G.G., de Freitas Dias, R., Mattos, L.V., Quelhas, O.L.G. and Leal Filho, W., 2017. Towards sustainable development through the perspective of eco-efficiency-A systematic literature review. Journal of Cleaner Production, 165, pp.890-904.

[46] Dalevska, N., Khobta, V., Kwilinski, A. and Kravchenko, S., 2019. A model for estimating social and economic indicators of sustainable development. Entrepreneurship and sustainability issues, 6(4), p.1839.

[47] Di Vaio, A., Palladino, R., Hassan, R. and Escobar, O., 2020. Artificial intelligence and business models in the sustainable development goals perspective: A systematic literature review. Journal of Business Research, 121, pp.283-314.

[48] Fitzgerald, B., Mockus, A. and Zhou, M., 2017. Towards engineering free/libre open source software (floss) ecosystems for impact and sustainability

[49] Galán, O.A.A., Valdéz, J.L.C., Medina, H.F., Contreras, G.A.V. and Sumuano, J.L.S., 2020. Proposal of a sustainable agile model for software development. International Journal of Advanced Computer Science and Applications, 11(1).

[50] Hoda, R., Salleh, N., Grundy, J. and Tee, H.M., 2017. Systematic literature reviews in agile software development: A tertiary study. Information and software technology, 85, pp.60-70.

[51] Kamble, S.S., Gunasekaran, A. and Gawankar, S.A., 2018. Sustainable Industry 4.0 framework: A systematic literature review identifying the current trends and future perspectives. Process safety and environmental protection, 117, pp.408-425.

[52] Kern, E., Hilty, L.M., Guldner, A., Maksimov, Y.V., Filler, A., Gröger, J. and Naumann, S., 2018. Sustainable software products—Towards assessment criteria for resource and energy efficiency. Future Generation Computer Systems, 86, pp.199-210.

[53] Lubberink, R., Blok, V., Van Ophem, J. and Omta, O., 2017. Lessons for responsible innovation in the business context: A systematic literature review of responsible, social and sustainable innovation practices. Sustainability, 9(5), p.721.

[54] McInnes, L.C., Heroux, M.A., Draeger, E.W., Siegel, A., Coghlan, S. and Antypas, K., 2021. How community software ecosystems can unlock the potential of exascale computing. Nature Computational Science, 1(2), pp.92-94.

[55] Mohankumar, M. and Kumar, D.M.A., 2015. Empirical study on green and sustainable software engineering. Advances in Software Engineering and Systems.

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

[57] Piccarozzi, M., Aquilani, B. and Gatti, C., 2018. Industry 4.0 in management studies: A systematic literature review. Sustainability, 10(10), p.3821.

[58] Ren, R., Hu, W., Dong, J., Sun, B., Chen, Y. and Chen, Z., 2020. A systematic literature review of green and sustainable logistics: bibliometric analysis, research trend and knowledge taxonomy. International journal of environmental research and public health, 17(1), p.261.

[59] Schön, E.M., Thomaschewski, J. and Escalona, M.J., 2017. Agile Requirements Engineering: A systematic literature review. Computer standards & interfaces, 49, pp.79-91.

[60] Stuermer, M., Abu-Tayeh, G. and Myrach, T., 2017. Digital sustainability: basic conditions for sustainable digital artifacts and their ecosystems. Sustainability science, 12(2), pp.247-262.

[61] Venters, C.C., Capilla, R., Betz, S., Penzenstadler, B., Crick, T., Crouch, S., Nakagawa, E.Y., Becker, C. and Carrillo, C., 2018. Software sustainability: Research and practice from a software architecture viewpoint. Journal of Systems and Software, 138, pp.174-188.

[62] Zelentsov, V., Brovkina, O., Pimanov, I. and Potryasaev, S., 2020, May. Automatization of forest ecosystems sustainability estimation based on complex modelling and Earth observation data. In IOP Conference Series: Earth and Environmental Science (Vol. 507, No. 1, p. 012034). IOP Publishing.

[63] Iatsyshyn, A., Iatsyshyn, A., Artemchuk, V., Kameneva, I., Kovach, V. and Popov, O., 2020. Software tools for tasks of sustainable development of environmental problems: peculiarities of programming and implementation in the specialists' preparation. In E3S Web of Conferences (Vol. 166, p. 01001). EDP Sciences.

[64] Taina, J., 2016. Good, bad, and beautiful software-In search of green software quality factors. Cepis Upgrade, 12(4), pp.22-27.

[65] Anthony, B.J. and Majid, M.A., 2016. Green IS for sustainable decision making in software management. Journal of Soft Computing and Decision Support Systems, 3(3), pp.20-34.

[66] Hindle, A., 2016, March. Green software engineering: the curse of methodology. In 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER) (Vol. 5, pp. 46-55). IEEE.

[67] Calero, C., Mancebo, J., García, F., Moraga, M.Á., Berná, J.A.G., Fernández-Alemán, J.L. and Toval, A., 2019. 5Ws of green and sustainable software. Tsinghua Science and Technology, 25(3), pp.401-414.

[68] Kern, E., 2018. Green computing, green software, and its characteristics: Awareness, rating, challenges. In From Science to Society (pp. 263-273). Springer, Cham.

[69] Robillard, M.P., 2016, November. Sustainable software design. In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (pp. 920-923).

[70] Aromataris, E. and Riitano, D., 2014. Systematic reviews: constructing a search strategy and searching for evidence. AJN The American Journal of Nursing, 114(5), pp.49-56.

[71] Bramer, W.M., Rethlefsen, M.L., Kleijnen, J. and Franco, O.H., 2017. Optimal database combinations for literature searches in systematic reviews: a prospective exploratory study. Systematic reviews, 6(1), pp.1-12.

[72] Henriette, E., Feki, M. and Boughzala, I., 2015. The shape of digital transformation: A systematic literature review.

[73] Snyder, H., 2019. Literature review as a research methodology: An overview and guidelines. Journal of business research, 104, pp.333-339.

[74] Khan, A.A., Shameem, M., Kumar, R.R., Hussain, S. and Yan, X., 2019. Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development. Applied Soft Computing, 83, p.105648.