CHALLENGES AND SOLUTIONS FOR SUSTAINABLE SOFTWARE DEVELOPMENT: A DEVELOPER'S PERSPECTIVE

AHSAN BALAL SHAKER

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

SEPTEMBER 2021

THESIS AND DEFENSE APPROVAL FORM

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

THESIS TITLE: Challenges and Solutions for Sustainable Software Development: A Developer's Perspective

Ahsan Balal Shaker
Submitted By:

003 MSSE/Ibd/F18
Registration No#

Master in Software Engineering
Title of the Degree

Software Engineering
Name of Discipline

Dr. Huma Hayat Khan
Name Of Research Supervisor

Signature: _______________

Dr. M. Nauman Malik
Name of Co Supervisor

Signature: _______________

Dr. Muzafar Khan
Name of HoD (SE)

Signature: _______________

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

Signature: _______________

Prof. Dr. Muhammad Safeer
Name of Pro-rector Academics

Signature: _______________

“I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of Master of Science in (*Software Engineering*)”

Signature: ______________________________

Name: Dr. Huma Hayat Khan

Date: ______________________________

Signature: ______________________________

Name: Dr. M. Nauman Malik

Date: ______________________________

CHALLENGES AND SOLUTIONS FOR SUSTAINABLE SOFTWARE DEVELOPMENT: A DEVELOPER'S PERSPECTIVE

AHSAN BALAL SHAKER

A thesis submitted in fulfillment of the
requirements for the award of the degree of
Master of (Software Engineering)

Department of Software Engineering
National University of Modern Languages

SEPTEMBER 2021

DECLARATION

I declare that this thesis entitled “*Challenges and Solutions for Sustainable Software Development: A Developer’s Perspective*” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ______________________

Name : Ahsan Balal Shaker

Date : ______________________

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.

ACKNOWLEGEMENT

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 supervisor, Dr. Huma Hayat Khan and Co-Supervisor Dr. M. Nauman Malik, who did not leave any stone unturned to guide me during my research journey.

I shall also acknowledge the extended assistance from the administration of Department of Software Engineering who supported me 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.

ABSTRACT

Sustainable software development has emerged as one of the leading concerns in the recent years. Sustainability in software development is a challenging task. With increasing development and use of software in the present world, software sustainability and its sustainable development are becoming big challenges for the industry. There is a need to check the awareness level of developers and identify the challenges and solutions for sustainable software development. It is found that there are few studies investigating the awareness level of software developers for sustainable software development besides its implementation challenges. The objective of this study is to check awareness level of software developer and provide strategy against the challenges for sustainable software development. To fulfil this objective the survey methodology is used. Survey is conducted through questionnaire to check the awareness level, and to catalog challenges and provide strategies against those challenges for sustainable software development. A total of 109 software developers participated in survey, and their opinion was taken that either they face enlisted challenges during sustainable software development or not. And different strategies were provided by the survey respondents to overcome those challenges. This research contributes towards state of knowledge and practice by highlighting the awareness level of software developers regarding sustainable software development, by enlisting challenges and by providing strategies to overcome those challenges which developers face during sustainable software development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVATIONS

OECD	-	Organization for Economic Cooperation and Development
ICT	-	Information and Communication Technology
SDGs	-	Sustainable Development Goals
SSDC	-	Software Sustainability Design Catalog
SEI	-	Software Engineering Institute
R_ID	-	Respondents ID

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter documents the research's background, statement of the research problem, research questions and research objective, scope of the study, aim of the research, methodology, and contribution, and significance of the research.

1.2 Background of Research

Software engineering is defined as *" A discipline that is concerned with the application of theory, knowledge, and practice to the effective and efficient development of software systems that meet the user requirements"* [1]. Software is among one of the most fast-growing sectors in the Organization for Economic Cooperation and Development (OECD) countries and comprehensively producing strong performances in all economic sectors. It directly contributes to the economy due to different software applications help in boosting the economy through an increasing array of applications and their use [2].

The 21st century has significant progress in information technology. Therefore the 21st century is defined as the information age, as it has experienced a major evolution in software development.at present a wide range of software applications used in every phase of daily life [3]. It's been barely more than 75 years in which computers and software have been developed, but their impact on humans is very important similar to the printing press, television, automobile, and airplanes. The changes are happening in every aspect of life due to software's and computers and contributing to human life: education, warfare, business, entertainment, medical, law, aviation industry, research, and orders other services that provide life's basic need like nuclear plants and electrical grid [4]. It is becoming part of daily life work.

Sustainability is becoming a concern as the role of software in society is increasing and impacting the environment, Addressing such concerns in the operation and development of software systems is a vital concern [5]. So, it's becoming very important to follow sustainable software development, as the Sustainability of software is generally referred to as economic, technical, social, and environmental [6].

Sustainability is defined as or refers to a wide set of objectives involved in sustainability: which are range from the software which use less energy (environmental sustainability), save or reduce the software maintenance and development cost (economic sustainability), deal with the well-being of humans (social sustainability), technical sustainability which refers to the long age of the software or characteristics of software that keeps the software alive for a longer period of the time [7]. Sustainability design refers to deal the sustainability as the first concern during the development of the software product [8].

Sustainability of software also sometimes refers to 'green software'. Software is being built without taking into consideration its following dimensions which are social, environmental, technical, and economic aspects [9]. The literature related to challenges, concerns, importance, and benefits for sustainable software development was observed.

Software sustainability very important as in the present age technology is changing very quickly and it can be very costly and painful for people and organizations to compete with this changing nature of technology in the future. Which led people in understanding that what sustainability means and how it's related to the software engineering field [10].

The achievement in the global software engineering environment is not due to the influence of technology and process, but people who play a vital role in the context of specific organizational culture, who apply the technology and initiate processes. For starting stage process maturity is a good idea; though, the success of sustainability depends upon people's maturity. While people's maturity depends on their excellence and growth at what they do [11]. Sustainable systems are different from other systems in regards that their functionality must be balanced between the factors social, environmental, technical, and economic [12].

In recent few years, the software engineering field is wideout software sustainability has got intention towards it with ranging challenges and concerns about sustainability. Sustainability is now a major concern in society, but there is very little understanding that how sustainability is perceived by software engineer professionals and how it can become a part of the software development process [8]. The organization also has less knowledge and awareness regarding sustainability and its benefits to society [8]. So, there is a need for some effort in educating the organization and software professionals about the sustainability factor, impacts, and benefits.

Software sustainability is a systematic challenge, it's not an individual, team, or organizational failing but it affects the many communities which include the professional software developer, open-source communities, and scientist who develop their research software [13]. Sustainability has become a world's concern, according to the UN Brundtland commission sustainability should meet present needs without compromising future generation's capability of achieving their own needs [14]. Hence the next generation's approaches should meet all the dimensions related to sustainability.

A huge majority of CEOs (89%) recognize the fact that after a huge commitment of industries they still face sustainability-related challenges to attain it. Particularly in smaller projects, the challenge of lack of resources occurs to create an infrastructure and methodology [15]. H.S. Zhu, C.L. Yu, D. Liu presents a novel and simple model named 'Eco' in their paper which is to help developers to develop sustainable software by design [16]. Which highlight that sustainability can be achieved by negotiating supply and demand and keeping the balance between the two like market economics [16].

While software almost involves or present in every aspect of modern society, there is a lack of knowledge in building them green. Greens are "Towards a green software body of knowledge. The reason behind this can be that either software developers are unknown to sustainability concept or they do not have clear guidelines or methods towards building green software.

Sustainable software's are the need of future development to deal with the sustainability challenges such as technical sustainability, economic, social, and environmental sustainability, which shows the importance of sustainable software development. To deal with challenges and concerns a lot of focus required to develop such approaches which deal with these challenges and concern [17].

1.3 Problem Statement

Sustainability in software is a long and challenging task. With the increasing development and use of software are in the present world, software sustainability and its sustainable development are becoming a big challenge for the software industry and world to tackle. As sustainability is stated the development of the guidelines and practices emphasizes software development. The developer has aim responsibility of practicing sustainability and building reliable and sustainable software, which lessens the adverse impact on the economy to meet the user needs, societies, and environments [2].

The lives of people have started to change due to the increase in ICTs which put society in the way of insurgency[18]. As for now the sustainability concern has not been tackled and researched at large, yet the increasing concern is detected related to sustainable software. Sustainability concerns include social, economic, environmental, and technology-related. Based on such research it's been clear that in the software engineering field sustainable software has appeared as a new challenge.

Software is nowadays a very important part of our daily life, so does sustainable software is the need of today's world. A lot of research has been conducted on software sustainability regarding importance, practices, assessment and a little research has focused on the hands-on practices that how the industry is looking at the matter and what level of awareness exist, moreover software developers are at what extent exercising the sustainable practices [6][19][20].

1.4 Research Questions

RQ1: What is the awareness level of sustainability among software developers in Pakistan?

RQ 2: What challenges software developers to face during sustainable software development?

RQ 3: What solutions software developers need to overcome the challenges?

1.5 Research Objectives

Objective 1: Investigate the awareness level of software developers regarding sustainable software development.

Objective 2: Enlist the challenges software developers face while developing sustainable software.

Objective 3: Provide Strategies to overcome the challenges software developers face during sustainable software development.

1.6 Scope of Study

- In this study the challenges software developers face during sustainable software development are identified. Besides the solution strategies are also identified against those challenges.
- Awareness level of software developer regarding sustainable software development is also highlighted.
- Survey through questionnaire was conducted for the study.

- Target population for the survey was software developers who were working in the software industry.
- The targeted population was having one year plus of experience in development.
- Small to medium size organizations were selected and visited to distribute the questionnaire and get response from software developers working in those organizations.

1.7 Aim of Research

This study focuses on the awareness level of the developer that how many developers are aware related to sustainable software, what practices they are following for sustainable software development, and what challenges they are facing in developing sustainable software. This paper also focuses to give the solutions and strategies against the challenge's software developers face during software development.

1.8 Methodology

The survey was conducted to get answers regarding the research questions. The survey is a method of gathering data from a large population of interest [21]. The survey of this study will be followed by the guidelines given by Kasunic [31]. The steps of the survey are shown in figure 1.

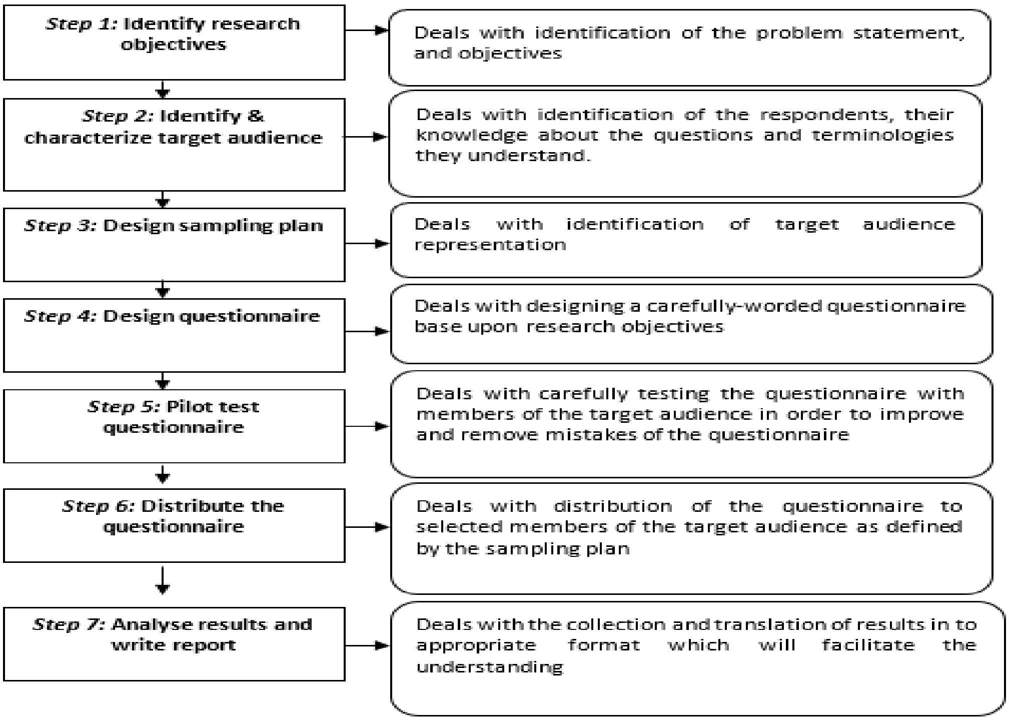
Step 1: Identify research objectives
Deals with identification of the problem statement, and objectives
Step 2: Identify & characterize target audience
Deals with identification of the respondents, their knowledge about the questions and terminologies they understand.
Step 3: Design sampling plan
Deals with identification of target audience representation
Step 4: Design questionnaire
Deals with designing a carefully-worded questionnaire base upon research objectives
Step 5: Pilot test questionnaire
Deals with carefully testing the questionnaire with members of the target audience in order to improve and remove mistakes of the questionnaire
Step 6: Distribute the questionnaire
Deals with distribution of the questionnaire to selected members of the target audience as defined by the sampling plan
Step 7: Analyse results and write report
Deals with the collection and translation of results in to appropriate format which will facilitate the understanding

Figure 1.1: Steps for Survey Conduction, Adopted from Work of Kasunic [31]

The first step of the survey was defining research objectives which are to check the awareness level of software developers regarding sustainable software, identify the challenges software developers face, and give solutions to those challenges. In the 2nd step, the targeted audience for the survey was identified which were the developers and they can be both male and female, who were selected from the software companies or independent software developers.

The third step in the survey was to design the sampling plan, the sample was consisting of at least 109 developers. The 4th step was designing a survey instrument, which comprises questionnaires that will be adopted from the paper [23]. In the next step survey, data were analyzed using SPSS software, in which frequency statistical technique was used.

1.9 Research Contribution and Significance

As software sustainability is the leading concern these days, Green software or environmental sustainability, economic, technical, and social sustainability are very important in any software development.

The study provides the following contributions:

The first contribution of this study is to identify the awareness level of sustainability among software developers. This furthers the existing state of practice and shows the developers about their knowledge regarding sustainable software development.

The second contribution of this study is a list of challenges software developers face during sustainable software development. This will add an existing state of knowledge and further the software engineering body knowledge.

The third contribution of this study is the list of solution strategies against the challenges. This will help the software developers working in the software development environment to overcome the challenges faced by them.

1.10 Outline of Thesis

The thesis outline is given in Figure 2, which highlights the topic discussed in each chapter. The scope of the study is to identify the challenges and solutions for sustainable software development: A developer's perspective. The thesis report consists of five chapters. The first chapter concerning with the introduction of sustainable software development its importance, problem statement also describes in the first chapter which is the main objective and concern area of this research. Further chapter one is also describes the research questions and their objectives, scope of the study, contribution and significance of the study. The purpose of this chapter is to set the foundation for the rest of the thesis.

Chapter two is “Literature Review” which discusses the existing studies related to challenges and solution for sustainable software development. It also discusses the gap in existing studies. Third chapter is “Methodology” which highlights the methodology used to highlight the awareness level of software developers and challenges they face during sustainable software development and solutions against those challenges.

The fourth chapter is “Challenges and Overcoming Strategies” which provides the result we got by conducting the survey and analysis of those results. Chapter five of the thesis report discuss the conclusion, limitations and future work of the thesis.

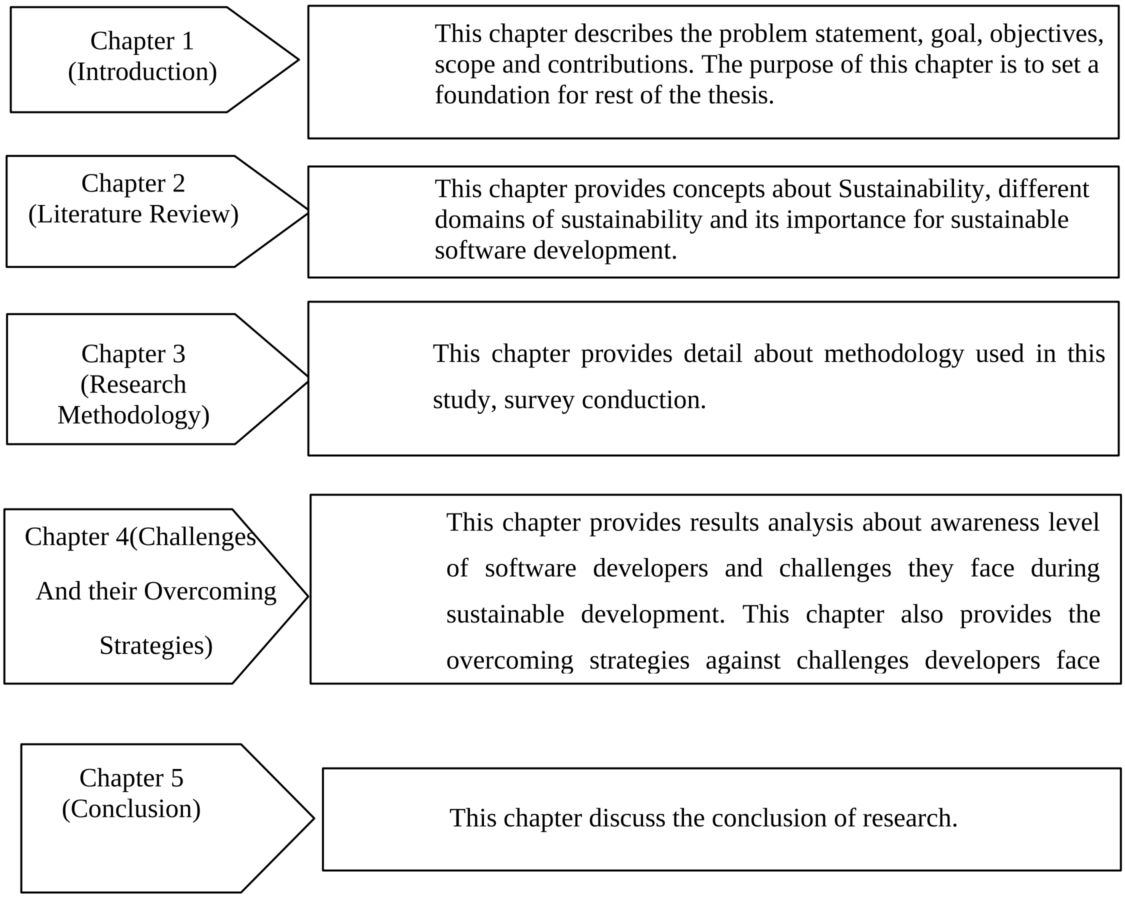
Chapter 1 (Introduction)
This chapter describes the problem statement, goal, objectives, scope and contributions. The purpose of this chapter is to set a foundation for rest of the thesis.
Chapter 2 (Literature Review)
This chapter provides concepts about Sustainability, different domains of sustainability and its importance for sustainable software development.
Chapter 3 (Research Methodology)
This chapter provides detail about methodology used in this study, survey conduction.
Chapter 4(Challenges And their Overcoming Strategies)
This chapter provides results analysis about awareness level of software developers and challenges they face during sustainable development. This chapter also provides the overcoming strategies against challenges developers face
Chapter 5 (Conclusion)
This chapter discuss the conclusion of research.

Figure 1.2: Thesis Outline

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In our previous chapter, we have reported the introduction of our study where the background of the problem, problem statement, research questions and objective, research's aim, and contribution of the research is reported. This chapter further extends it by exploring the related work published to our problem area and our research.

2.2 Software sustainability

Software sustainability is an emerging, most important, leading, and vital method in developing software engineering for sustainable software development. Also, degradation of the environment has directed an increasing concern towards sustainable software development. And a lot of research and work is being done in the sustainability domain. Applying sustainable software development is a tough and challenging task. So due to increasing interest and focus on sustainability lots of challenges are rising which are needs to be addressed. And also needs to check the awareness level among the software developers working in the industry.

Maintenance, adaptation, and evaluation are painful and costly because of constant and reckless evolution in the technology field (e.g. paradigms, programming languages, and tools) for an organization, which makes it challenging to use software-based systems in the future more proficiently. Which resulted in understanding means of sustainability in the software engineering field in recent years [10].

Sustainable Software Engineering is the most concerning matter in the current era. Sustainable Software Engineering and Sustainable Software are among the statements which are being presented by researchers for sustainability. For example, Sustainability denotes *"capacity to endure"* although Sustainable Software refers to *"software that can be continuously assessable, maintainable and well documented*". It aims to provide 'Sustainable Software' due to sustainability as a major concern. In software, engineering sustainability is still evolving [24].

The "Sustainability" is used in many fields which states capacity to last for a longer period. Besides some researchers have also described Sustainable Development as *"includes the aspects to develop a sustainable software product, as well as the aspect to develop a product using a sustainable development process".* According to Brundtland's report from United Nations (UN), Sustainable Development refers to "*meet the needs of the present without compromising the ability of future generations to satisfy their own needs".* The field of software engineering is named "Sustainable Software" which refers to software being environment friendly and which last for a longer duration of time. Sustainable Software is defined as "*software whose impact on the economy, society, human beings and environment that results from the development, deployment, and usage of software are minimal and/or which have a positive effect on sustainable software development"*[6].

Sustainable Software Engineering is now recognized as the most valuable concern among researchers, standardization bodies, and industry. In a Microsoft report as well as IBM global chief executive officer's (CEO) study it is shown that a huge number of organizations to encompass sustainability are reforming their business models. 48% out of 4000 managers and executives who were interviewed agreed that they updated their business model to implement sustainability [25].

At United Nations (UN) Sustainable Development Summit held in September 2015 the plan for Sustainable Development 2030 was agreed upon.it included 17 Sustainable

Development Goals (SDG), which included 169 targets related to economic, social, and environmental features. It recognizes that "the spread of ICT and global interconnectedness has great potential to accelerate human progress, to bridge the digital divide and to develop knowledge societies". To achieve SDGs 2030 plan there is a need to highlight the awareness related to innovating ICTs and assisting all nations to achieve this. However, ICTs may have effects on sustainability, like electronic waste, power consumption, waste emission, etc. which makes ICTs also a critical part of achieving the SDGs 2030 [26].

Multiple dimensions of sustainability have been shown by researchers in the field of software engineering. These dimensions are Economic, Environmental, Social, and Technical sustainability as shown in figure 2.1.

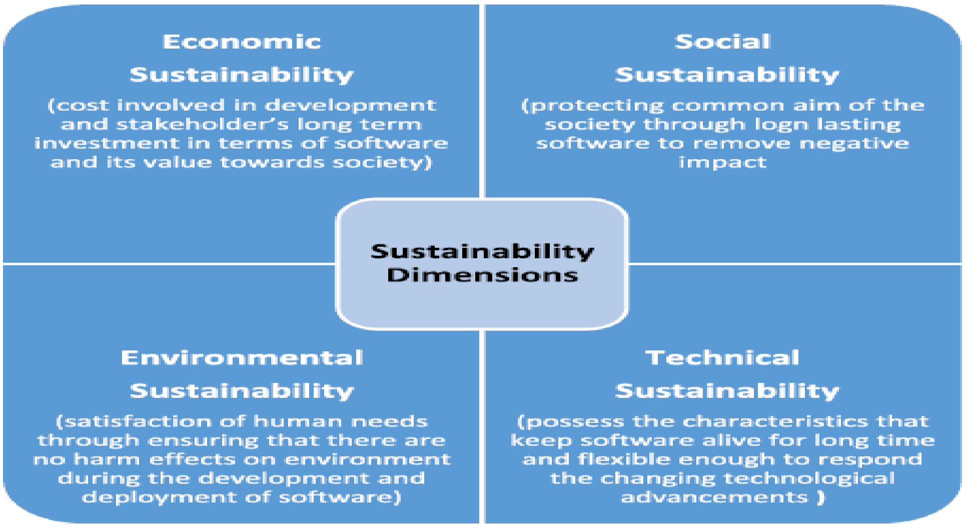
Economic Sustainability
(cost involved in development and stakeholder's long term investment in terms of software and its value towards society)
Social Sustainability
(protecting common aim of the society through logn lasting software to remove negative impact
Sustainability Dimensions
Environmental Sustainability
(satisfaction of human needs through ensuring that there are no harm effects on environment during the development and deployment of software)
Technical Sustainability
(possess the characteristics that keep software alive for long time and flexible enough to respond the changing technological advancements)

Figure 2.1: Software Sustainability Domains [6]

2.2.1 Economic Sustainability

Economic sustainability is defined as the *"cost involved in development and stakeholder's long-term investment in terms of software and its value towards society"*. The economic sustainability dimension covers the financial aspects and business values. It also covers Customer Relationship Management, Supply Chain, Governance, Processes, R&D, and innovation. It about how the system creates or destroys the value, how it affects the relationships between business and customers, whether it changes the business supply chain, R&D, processes, and governance.

This deals with the matter that how can software be developed so that there is a minimal economic risk for customers or stakeholders and their investments are for a longer duration and are safe. For software engineering, it means that how economically it is safe [6][25][29].

2.2.2 Social Sustainability

Social sustainability is defined as *"protecting the common aim of society through long-lasting software to remove negative impact"*. It covers relationships among individuals and groups. It also describes the domain that how the system might affect the peoples' sense of belonging, their trust in the surroundings, perception of others, are there receiving equal treatment compare to others and how they participate in social groups.

Its main aim is to lessen the harmful impacts on society. It also mentions giving reimbursements to society through diverse software systems in different software fields. There are two categories of society: (a) technical community and (b) the users' one [6][27][28].

2.2.3 Environmental Sustainability

Environmental sustainability can be defined as *"Satisfaction of human needs through ensuring that there are no harmful effects on the environment during the development and deployment of the software"*. Environmental sustainability highlights the dimensions of use and stewardship of natural resources. It also includes "Material and Resources, Atmospheric and Water pollution, Energy, Soil, Biodiversity, Logistics, and Transportation and Land Use". It also refers to how the system may use the resources, production of waste, and emissions.

Protection of natural resources like water, land, air, and energy, etc. is among the main aims. It deals with reducing wastage, saving resources for human need satisfaction. Moreover, there is a need to be certain that no damage is or will be done during the development and deployment of the system. For instance, through the emission of heat from the computer systems. Environmentally friendly software refers as "Green Software". Environmental sustainability can be evaluated through two aspects: Energy consumption and Resource consumption [6][27][28].

2.2.4 Technical Sustainability

Technical sustainability is defined as *"possess the characteristics that keep the software system alive for a long time and flexible enough to respond to the changing technical advancements"*. The technical dimension covers the software system's ability to change while providing the required features and capabilities. It also covers the topic of maintainability, usability, extensibility, adaptability, security, and scalability.

Technical sustainability is related to long-lasting and adaptability to future changes. Software survivability is influenced by functional and non-functional characteristics. Function software evolvement is because of requirement change, while technical evolvement is because of quality characteristics like maintainability, security, adaptability, and Usability. So as the technology is changing or advances at a rapid pace it is required to be developed the new software according to the new technological paradigm [6][27][28].

Table 2.1: Existing Studies on Software Sustainability

Author/ Year	Domain	Contribution	Limitations	Methodology
N.C. Fernandez, P. Lago/ 2019 [10]	Towards a Software Sustainability-Quality Model Insights from a Multi-Case Study	This research contributed by investigating in a different context the applicability of the	A niche sourcing approach to identify and extend the model with other dependencies.	Multiple Case Study

		Sustainability Model. Also, identify the new quality features missing in previous research and contributed to dimensions of sustainability.	Increase the number of case studies or conduct more case studies to investigate the applicability of the sustainability model to identify QAs and for detecting other dependencies.	
Sumaira Nazir, Nargis Fatima, Haslina Sarkan etl./ 2020 [24]	A perspective of individual software sustainability in sustainable software engineering.	Individual sustainability challenges were highlighted and prioritized. Provides mitigation strategies related to individual sustainability for the top five.	Enhanced by identifying the degree of challenges. Find better mitigation strategies.	Literature Review
M.N. Malik, H.H. Khan /2018 [6]	Sustainability for Software Crowdsourcing	The importance of software crowdsourcing was emphasized.	Selection of IEEE standards.	Literature Review
Shola Oyedeji, Ahmed Seffah, Birgit Penzenstadler/ 2018 [25]	Software sustainability design	This paper contributes by introducing a sustainability design catalog (SSDC) which contains or provides a series of guidelines. It aims to contribute by assisting software developers in sustainable requirement gathering. And by testing, measuring sustainability.	The SSDC also has automation potential. For recommender system design catalog can become base.	Literature Review
José, Juan etl. / 2018 [26]	Surveying Technical and Environmental dimensions in industry.	Identified practices that can help in implementing sustainability. Identified gaps among software development processes that companies apply.	No clear guidelines for proceeding in this work.	Survey

Birgit Penzenstadler, Leticia Duboc, Sedef Akinli Kocak/ 2020 [20]	Improve awareness related to sustainability for system designs and business processes.	The workbook at hand provides the instruments used to carry out the Sustainability Awareness Workshops we use with companies.	Interviewees might not consider long-term, compounded impacts.	Feasibility study
Rahma AMRI, Narjès BELLAMINE BEN SAOUD/ 2014 [28]	Generic Sustainable Software model	Can help in evaluating the software sustainability of a project.	Controlling of sustainability aspects during software development.	Literature Review

All the studies are shown in Table 2.1 report the significance/importance of sustainable software in the future. The recent works were done by B. Penzenstadler [20] who identified the instrument used to carry out sustainability awareness workshops. They provide the questions against each domain which can be used to check the awareness level of software developers working in the industry.

Similarly, the study performed by M.N. Malik & H.H. Khan [6] contributed by stressing the software crowdsourcing importance which can cause the failure of software, if not dealt. They study the IEEE software values with a lens of sustainability software crowdsourcing. They followed the three-layered protocol for searching in which the most relevant one was IEEE standards which were recovered and inspected.

Another study performed by Nelly and Lago [10] contributed by investigating in a different context the applicability of the Sustainability Model. It also identifies the new quality features missing in previous research and contributed to dimensions of sustainability. Two software projects were selected as a case study for their study where individually project was conducted self-sufficiently in precise circumstances.

We reviewed another study by Ahmed and Birgit [25] in which they introduce a sustainability design catalog (SSDC) which contains a series of guidelines. It aims to contribute by assisting software developers in sustainable requirement gathering. And by testing, measuring sustainability. For analyses, by using Karlskrona manifesto principles on sustainability design they used four case studies.

Another study was done by José and Juan etl. [26] Identified practices that can help in implementing sustainability. They also contributed by identifying gaps among software development processes that companies apply. Identified practices that can help in implementing sustainability. They surveyed practitioners to gather information related to their awareness regarding sustainable software development. They used the gathered data for studying professional's prospective regarding sustainability.

The study carried out by Rahma and Saoud [28] which contributed to formulating a generic sustainable model, which can be used for evaluating the sustainability of software project. The model covers the different software sustainability domains e.g. economic, environmental, social, and technical. Against every dimension, they describe the consistent software sustainability standards.

All the studies discussed in this section were discussing software sustainability in their respective context. Although every one of them was having a significant contribution to this, none of them focused on the to check the awareness level of software developers regarding sustainable software development and what challenges they are facing during sustainable software development and provide solutions against those challenges, which was our point of concern and area of research. The gaps between existing studies are reported in Table 2.2.

Table 2.2: Gap in Existing Studies on Sustainable Software Development

Authors, Year of Publication	Domain of Contribution	Identified Gap		
		Awareness Level	Challenges	Solutions/Strategy
Nelly Condori-Fernandez, Patricia Lago/ 2019 [10]	This research contributed by investigating in a different context the applicability of the Sustainability Model.	X	X	X

	Also, identify the new quality features missing in previous research and contributed to dimensions of sustainability.			
Sumaira Nazir, Nargis Fatima, Haslina Sarkan etl./ 2020 [24]	Individual sustainability challenges were highlighted and prioritized. Provides mitigation strategies related to individual sustainability for the top five.	X	Provide a list of challenges against the individual sustainability domain	X
M. N. Malik, H.H Khan /2018 [6]	The importance of software crowdsourcing was emphasized. If not dealt with it could be caused by the failure of software.	X	X	X
Shola Oyedeji, Ahmed Seffah, Birgit Penzenstadler/ 2018 [25]	This paper contributes by introducing a sustainability design catalog (SSDC) which contains or provides a series of guidelines. It aims to contribute by assisting software developers in sustainable requirement gathering. And by testing, measuring sustainability.	X	X	X
José, Juan etl. /2018 [26]	Identified practices that can help in implementing sustainability. Identified gaps among software development processes that companies apply.	X	X	X
Birgit Penzenstadler, Leticia Duboc, Sedef Akinli Kocak/ 2020 [20]	The workbook at hand provides the instruments used to carry out the Sustainability Awareness Workshops we use with companies.	Provide guidelines or questions to check the awareness level	X	X
Rahma AMRI, Narjès BELLAMINE BEN SAOUD/ 2014 [28]	This paper contributed by evaluating the sustainability of the project.	X	X	X

As shown in Table 2.2 Nelly and Lago [10] provides a quality model insight based on the multi-case study. They investigated the pertinence of the sustainability model in diverse contexts, they contributed by identifying the missing quality attributes related to sustainability. But they didn’t focus to check the awareness level of software developers and provide challenges software developers face during sustainable development and provide solutions or strategies against those challenges. The same is the case with studies of Malik and Khan [6], Shola, Ahmed and Penzenstadler [25], Rahma and Saoud [28], José and Juan etl. [26] that they significantly supported sustainability but lack to provide challenges developers face during sustainable software development. They also lacked to provide any guidelines to check the awareness level of software developers working in the industry.

The study was done by Birgit and Duboc etl [20] provide the guidelines or questions to check the awareness level of software developers regarding sustainable software development but lacks a survey that how much percent of developers or known to sustainable software development or how much percent of developers or following the sustainable software development while developing the software product.

So, we have found out there is a lack of studies that have focused on awareness level and challenges developers face during sustainable software development and provide solutions against those challenges. In this research our guide is to check the awareness level, challenges developers face and provide solutions or strategies against those challenges.

2.3 Chapter Summary

This chapter described the concept of Sustainable Software. It further explains the concept of sustainable software development. The chapter reports the existing literature that supports software sustainability and showed some of the studies that report software sustainability or sustainable software development directly or indirectly. Furthermore, the gaps in the existing studies are highlighted and described in this chapter. The methodology for conducting this research is discussed in Chapter 3.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In Chapter 2, we reported literature that showed the gap in the field of sustainable software development. In the third Chapter, the methodology which was used to pursue the research is reported and discussed.

3.2 Methodology

Research methodology is defined as "*A systematic way to solve a problem, the procedure by which researchers go about their work of describing, explaining and predicting phenomena are called research methodology".* It is also defined as *"The study of method by which knowledge is gained" or "It is a science of studying how research is to be carried out"* [29].

From the view point of objectives, research can be classified as descriptive, correlational, exploratory and explanatory. Our research comes under exploratory classification.

3.2.1 Exploratory Research

It is defined as *"Exploratory Research is undertaken to explore an area where little is known or to investigate the possibilities of undertaking a particular research study (feasibility study/pilot study)".* Exploratory research is conducted to get better idea about the problem, but it does not provide conclusive results. For such research, the researcher uses general ideas to identify the issue by using this technique. It is also referred to as grounded theory approach or interpretive research as it uses to answer questions like what, why and how [30].

It is also known as preliminary research which is used to clarify the exact nature of problem. Exploratory research can add an important, insightful and quality information to a study which can be significant for the study. In exploratory research is inexpensive, highly interactive and open ended in nature. In such kind of research there is usually no prior information is available from earlier researchers. It is usually flexible and scattered [29].Survey methodology was used and following survey guidelines were followed to conduct the survey for the study.

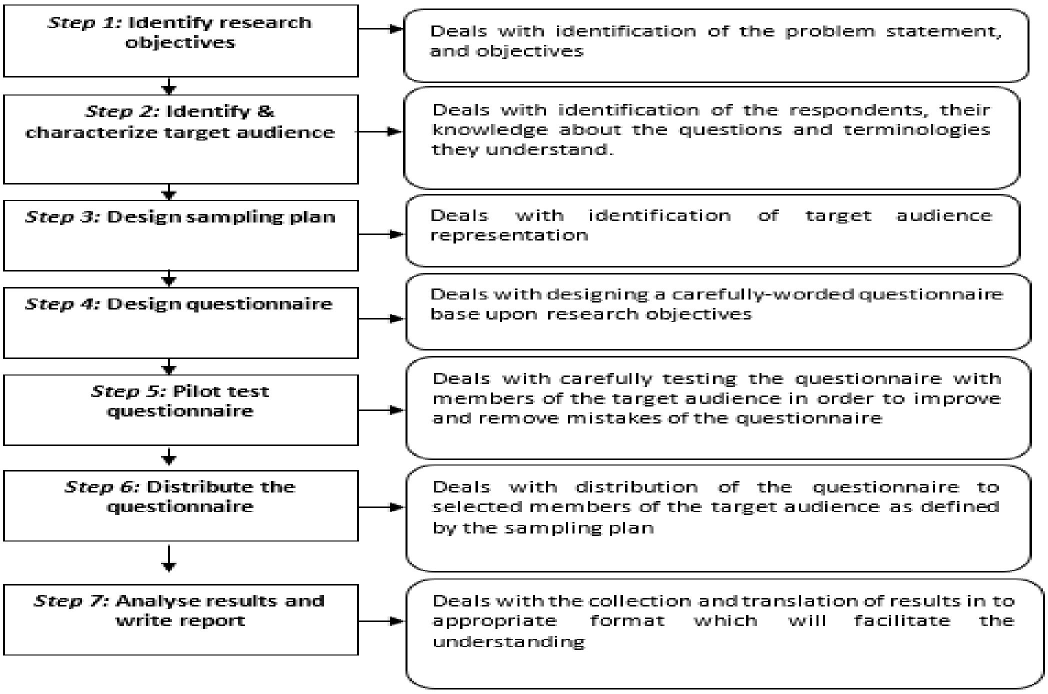
Step 1: Identify research objectives
Deals with identification of the problem statement, and objectives
Step 2: Identify & characterize target audience
Deals with identification of the respondents, their knowledge about the questions and terminologies they understand.
Step 3: Design sampling plan
Deals with identification of target audience representation
Step 4: Design questionnaire
Deals with designing a carefully-worded questionnaire base upon research objectives
Step 5: Pilot test questionnaire
Deals with carefully testing the questionnaire with members of the target audience in order to improve and remove mistakes of the questionnaire
Step 6: Distribute the questionnaire
Deals with distribution of the questionnaire to selected members of the target audience as defined by the sampling plan
Step 7: Analyse results and write report
Deals with the collection and translation of results in to appropriate format which will facilitate the understanding

Figure 3.1: Steps for Conduction of Survey, Adopted from Kasunic [31]

3.2.2 Survey

Guidelines provided by Kasunic which were published by “Software Engineering Institute (SEI)” were followed for conducting the survey. For conducting the survey his work was used as it among the most common and most used in the software engineering’s field. Figure 3.1 shows steps for the conduction of the survey [31].

Research Objectives for Survey Conduction: The survey conduction questions and objectives were identified before the conduction of the survey.

Target Audience: This study focuses on the software developers who develop software products or are involved in the procedure for the development of software. Software developers working in software organizations or as a freelancer were the target audience for the study. For recognition of the appropriate target audience, the focus was put on particular questions adopted from the work of Kasunic [31] as shown in table 3.1. Table 3.1 shows the question’s set, which was focused while classifying and characterizing our target audience.

Table 3.1: Questions for identifying and characterizing target audience, Adopted from work of Kasunic [31]

Questions for Identifying and Characterizing the Target Audience
1. “How many people are in the population we are studying?”
2. “What are their jobs and responsibilities?”
3. “What is the most common education level?”
4. “What relevant experience do they possess?”
5. “What technical abilities do they possess?”
6. “What is the age range of the respondent population?”
7. “Do we anticipate that they would have difficulty with using a questionnaire that is: − mailed to them? − completed using a computer via the internet? − handed to them?”
8. “What can we assume about their knowledge of the domain we are studying in the survey?”
9. “How much of their time can we assume they will spend completing the questionnaire?”

The target audience selected for the survey was developers who possess the required and relevant qualifications as shown in figure 3.2. About 65% of respondents were one who possesses a bachelor’s degree in the field of Information Technology, Cyber Security, Software Engineering, and Computer Science. Similarly, around 31% were the ones who possess the Master’s level qualification and their specializations were Project Management, Software Engineering, Information Technology, Cyber Security and Computer Science, while 4% of the respondent possess a Ph.D. in software engineering and computer science as shown in figure 3.2.

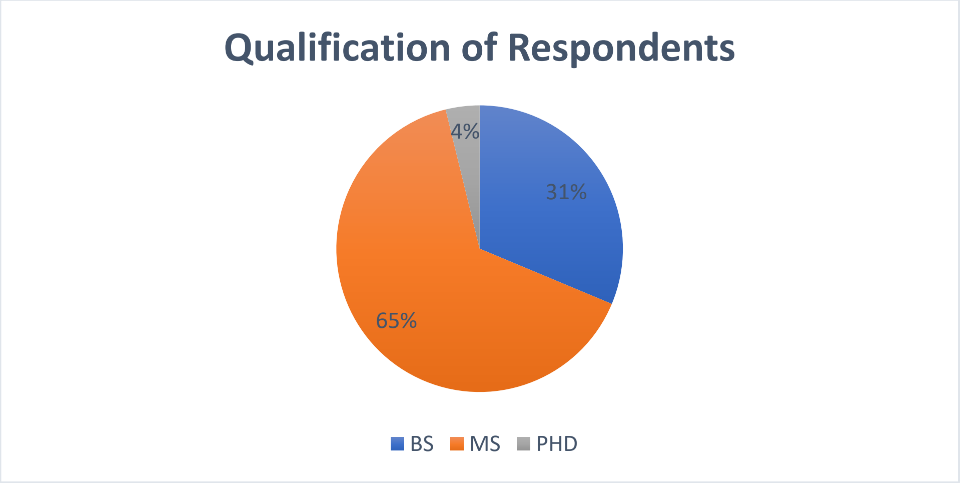

Figure 3.2: Qualification of Respondents

The target audience also selected the software developers at different positions or designations in their organizations as shown in figure 3.3.

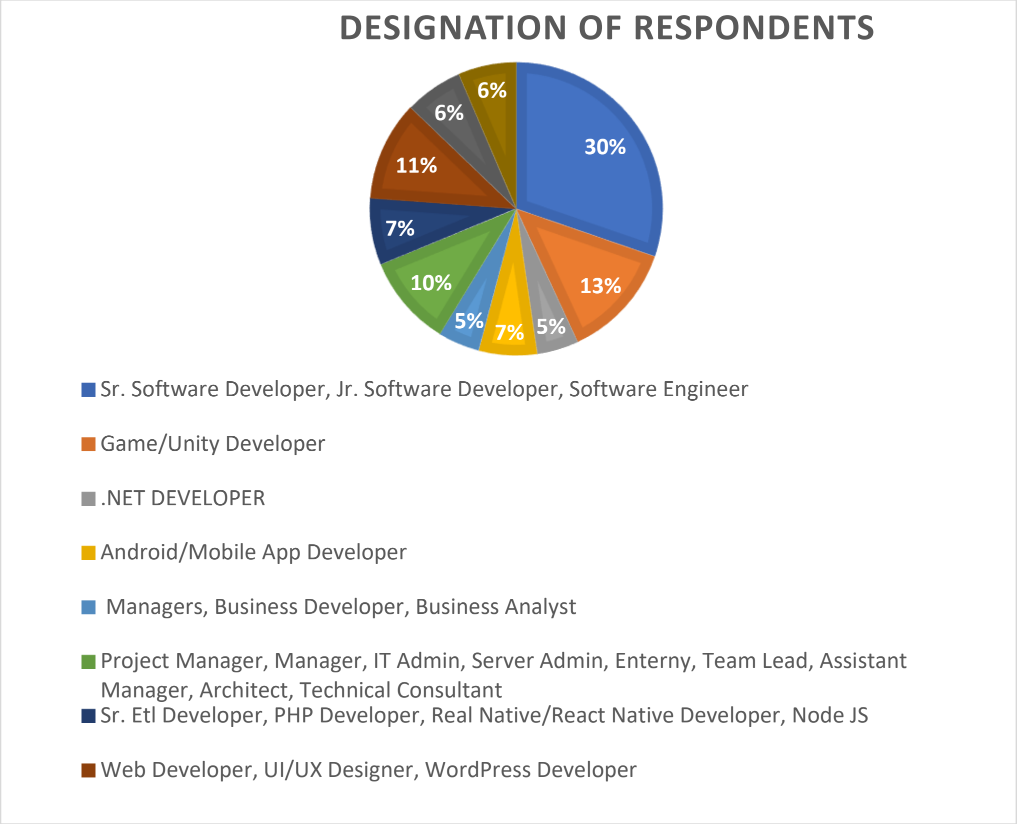

Figure 3.3: Respondents Designations

Figure 3.3 shows the respondent's designations and their percentage. It includes .Net Developers, Android Developers, Mobile App Developers, Game/Unity Developers, Sr. Software Developers, Managers, Business Analysts, Project Managers, Admins, Team Leaders, Technical Consultants, Sr. ETL Developers, PHP Developers, Freelancers, Node JS, React Native, Web Developers, Database Developers, IOS Developers, SEO Experts, and Software Architects.

Sampling: A subset of the total population, which has the characteristics of the population is known as sampling. This questionnaire was sent/emailed to the companies and software developers through google form and also was filled out through hard copies from different software companies and software developers.

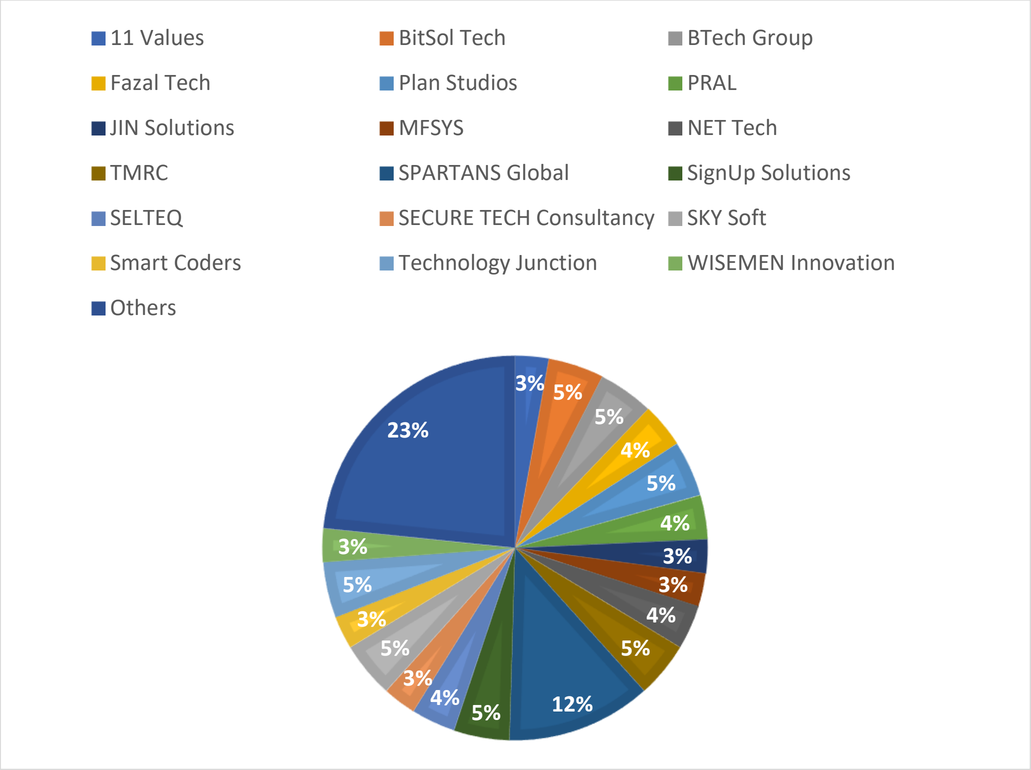

Figure 3.4: Organizations/Companies Contacted for Survey

Figure 3.4 shows some of the organizations which were contacted by survey conduction through email and directly in person. All the organizations are working in the field of software development and are recognized under PSEB (Pakistan Software Export Board).

The detail of the companies and persons are attached in Appendix B. The criteria for short listing and sending the questionnaire to companies and other persons was that they must be doing software development. For this study convenience sampling strategy to access and send the questionnaire to them. A specific person in the company was selected and was given the questionnaire so that the questionnaire could be filled out by the specific person.

Questionnaire Development: Designed Questionnaire was for:

1. To highlight the awareness level of software developers regarding sustainable software development.

2. To check/enlist the challenges developers face during sustainable software development.
3. To provide solutions against the challenges developers face during sustainable development.

By addressing the questions adopted from the work of Kasunic[31] it was made possible. Table 3.2 shows which considered and while designing and developing the questionnaire. The questionnaire is attached in Appendix A.

Table 3.2: For Designing the Questionnaire the considered questions, Adopted from the work of Kasunic [31]

Questions for Designing a Questionnaire
1. “How will the survey be mediated (e.g., via paper, email soft copy, Web)?” 2. “How long should the questionnaire be?” 3. “How should the questionnaire be structured and organized?” 4. “What page design and formatting will be most effective?”

The questionnaire was divided into nine sections. The first section of the questionnaire was used to get the respondent's name, qualification, organization name, and designation. The 2nd section of the questionnaire contains questions to check awareness levels regarding Social Sustainability. For this section of the questionnaire, we used a three-point Likert scale; Agree=1, Disagree=2, Don’t know=0. Similarly, the third section which was aimed to check the awareness level of developers regarding Environmental Sustainability. For this section of the questionnaire, we also used a three-point Likert scale; Agree=1, Disagree=2, Don’t know=0.

The fourth and fifth sections of the questionnaire were also designed to check the awareness level of respondents/ software developers regarding Economic sustainability and Technical sustainability. And for the fourth and fifth sections of the questionnaire we also use a three-point Likert scale; Agree=1, Disagree=2, Don’t Know=0.

The sixth section of the questionnaire consists of Environmental Sustainability and aims to check the challenges software developers face during sustainable software development and solutions against those challenges they face during sustainable software development. This section of the questionnaire used a five-point Likert scale; Strongly-Agree=1, Agree=2, Strongly-Disagree=3, Disagree=4 and Don't Know=0. The five-point Likert scale was used to get a more detailed and wide perspective of software developers regarding the challenges they face during sustainable software development. And space was given under each question to write down the solution or strategy against those challenges.

Similarly, the seventh, eighth and ninth section of the questionnaire consists of Economic Sustainability, Technical Sustainability, and Social Sustainability respectively also was aimed to check the challenges software developers face during sustainable software development and to ask for the solutions or strategy against those challenges they face. These sections also used a five-point Likert scale; Strongly-Agree=1, Agree=2, Strongly-Disagree=3, Disagree=4, Don't Know=0. And in each section, space was given under each question to write down the solution or strategy against those challenges.

Pilot Test: For the development of the questionnaire a pilot study was also conducted, for modification and exclusion of information until the final questionnaire was finalized. It was conducted for further improvement of questionnaire and validation, in terms of wording, statement, and sequences as well as interests of contributors. Then it was forwarded to about 10-15 members of the target audience. The remarks given were generally about the sentence and wording used in the questions. Some respondents commented that they find it difficult to understand some definitions defined in the questionnaire and similarly some show concern related to questionnaire description. Based upon their suggestions and comments questionnaire was modified and improved/corrected. For their clear understanding and to lessen the ambiguity of the questionnaire the questions were corrected and improved. The final version of the questionnaire is attached in Appendix A.

Questionnaire Distribution and Data Collection: After completion of the pilot test questionnaire was sent to the target audience. The survey package included an invitation email and questionnaire. In this study, an online survey was used and a questionnaire was also distributed via hard copies among the target audiences to get the data from the respondents. It was made sure that respondents are working as software developers and have respective field experience. The detail of the distribution of respondent responses is in chapter 5. The questionnaire is attached in Appendix A.

3.3 Chapter Summary

This chapter described the research design used for data collection and data analysis with methodology and techniques to achieve our research objectives. This Chapter has reported the implementation of survey methodology and its steps are discussed in detail. The result and analysis of the survey are discussed in Chapter 4.

CHAPTER 4

CHALLENGES AND THEIR OVERCOMING STRATEGIES

4.1 Introduction

Chapter 3 discuss the methodology and steps followed for survey conduction. While this Chapter illustrates the result of the survey.

4.2 Survey Conduction

Survey guidelines given by Kasunic [31] were followed for survey conduction. All information related to steps followed for the survey, survey's objectives, selection of target audience, for survey conduction is reported in Chapter 3. This study specifically focuses on Sustainable Software development from a developer perspective, so the target audience for research was software developers. The questionnaire was sent to the target audience and a list of the companies is attached in Appendix B. The questionnaire is comprised of nine sections. Section 1 was designed to gather respondent Bio data. Section 2 comprised of questions to check the awareness level of the software developers regarding Social Sustainability. Section 3 of the questionnaire was designed to check the awareness level of software developers regarding Environmental Sustainability. Similarly, Sections 4 & 5 were designed to check the awareness level of software developers regarding Economic Sustainability and Technical Sustainability.

Section 6 of the questionnaire was designed to check the challenges developers face during Environmentally Sustainable software development and to get the response of the respondents to provide the strategy against those challenges. Section 7, 8, and 9 were also designed to check what challenges developers face during Sustainable Software Development and to get solutions or strategies against those challenges.

The email was sent to about 50+ organizations and persons individually, the online response rate was very low and we only received a response from about 15 respondents which were very low. So, after that, it was decided to visit the organizations personally and distribute the questionnaire among the software developers. More than 150 questionnaires were distributed out of which we only got about 95+ responses. Finally, by combining all a total of 109 responses were received. Thus, for data analysis 109 responses were used.

4.3 Descriptive Statistics of Respondents

The first part of the questionnaire was to investigate or gather information about the respondent’s Bio. Respondents were asked about their Names, Qualifications, Designation, and Organization names. Figure 4.1 shows respondent distribution based upon their designation in the organization.

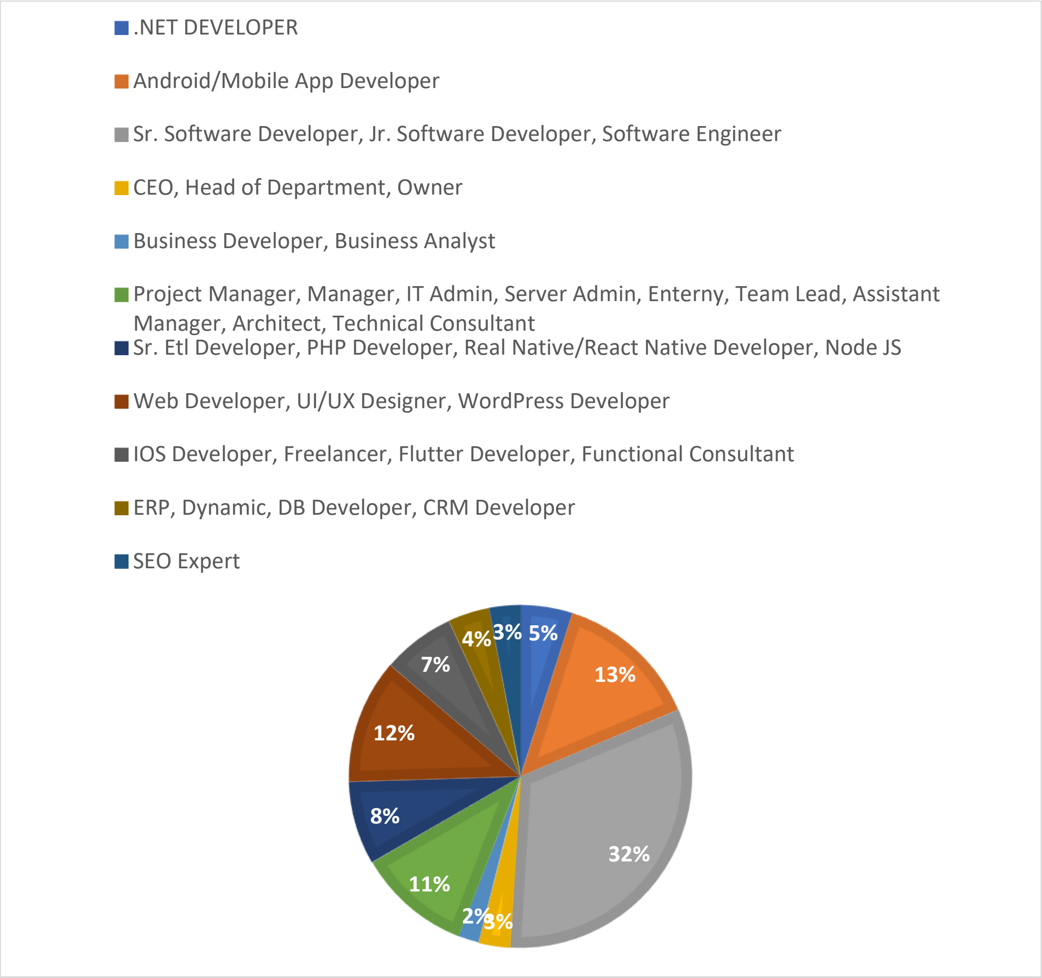

Figure 4.1: Distribution of Respondents, Based Upon Their Position/Designation in Organization

It is found that 30.27% of respondents were working as Sr. Software Developer, Jr. Software Develop, and as Software Engineer. About 12.84% were working as Game/Unity Developers. Similarly, 6.4% were working as Game/Unity Developers, 10.1 as Project Manager, Manager, IT Admin, Server Admin, Team Lead, Assistant Manager, Architect, Technical Consultant. Furthermore, 7.33% were working as Sr. ETL Developer, PHP Developer, Real Native/React Native Developer, Node JS, and 3.67% were working as ERP, Dynamic. The number of DB Developer, CRM Developer was 2.75%, SEO Expert and 6.4% as IOS Developer, Freelancer, Flutter Developer, Functional Consultant. The 11.01% as Web Developer, UI/UX Designer, WordPress Developer, and 4.5% were working as .NET Developer. Managers were about 2.75% and 1.83% were working as Business Analysts, Business Developers.

4.4 Result Analysis Related to Sustainable Software Development

This section reports the result related to the awareness level of software developers during sustainable software development and the challenges developers face during sustainable software development. For this purpose, the data was processed through statistical analysis, and frequency analysis was used. Frequency analysis is a descriptive analysis that shows the number of occurrences of each response chosen by the respondent of the questionnaire. Higher the frequency higher the awareness level and lower frequency means lower the awareness level.

4.5 Analysis related to Awareness Level of Software developers

Analysis related to awareness of software developers regarding sustainable software development was performed on the data gathered from respondents through the conduction of the survey.

4.5.1 Awareness Level of Software Developers Regarding Social Sustainability

The purpose of conducting this part of the survey was to check the awareness level of software developers regarding social sustainability during sustainable software development, "Social sustainability refers to protecting common aim of the society through long-lasting software to remove negative impact". The three-point Likert Scale ranging from 0-2 (Agree = 1, Disagree = 2, Don't Know = 0) was provided (Explained in Chapter 3).

Table 4.1: Social Sustainability Awareness Level Statistics

	1. Long-lasting software can remove the negative impact on society	2. Software can change the trust between users and the business	3. Software applications should be used by different backgrounds, age groups,	4. Software should help people take action to achieve the goals, projects, and

			that owns the software	and education level people.	tasks of a group.
N	Valid	109	109	109	109
	Missing	0	0	0	0

This section of the questionnaire includes four questions regarding social sustainability to check the awareness level of software developers regarding software sustainability.

Table 4.2: Long-lasting software can remove the negative impact on society

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	16	14.7	14.7	14.7
	Agree	50	45.9	45.9	60.6
	Disagree	43	39.4	39.4	100.0
	Total	109	100.0	100.0	

To check social sustainability awareness level the questions which were asked from software developers result against the first question which was “Long-lasting software can remove the negative impact on society”. In that, 45.9% agreed that long-lasting software’s can remove the negative impact on society while 39.4% disagreed and 14.7% answered: “Don’t know”.

Table 4.3: Software can change the trust between users and the business that owns the software

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	1	.9	.9	.9
	Agree	96	88.1	88.1	89.0
	Disagree	12	11.0	11.0	100.0

	Total	109	100.0	100.0	

The second question which was asked related to the social sustainability domain was that "software can change trust between users and business that owns the software", in which 88.1% agreed, 11.0% disagreed and .9% didn't know that either its can be done or not.

Table 4.4: Software application should be used by different backgrounds, age groups, and education level people

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	1	.9	.9	.9
	Agree	101	92.7	92.7	93.6
	Disagree	7	6.4	6.4	100.0
	Total	109	100.0	100.0	

The 3rd question which was "Software application should be used by different backgrounds, age groups, and education level people", 92.7% out of asked respondents agreed with this while 6.4% disagreed and .9% didn't know that either it can be used by different aged people or not.

Table 4.5: Software should help people take the actions to achieve the goals, projects, and tasks of a group

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	2	1.8	1.8	1.8
	Agree	105	96.3	96.3	98.2
	Disagree	2	1.8	1.8	100.0
	Total	109	100.0	100.0	

The 4th question asked related to social sustainability domain to check the awareness level of software developers regarding sustainable software development, 96.3% agreed that "software should help people take the action to achieve the goals, projects, and task of a group". While 1.8% disagreed and 1.8% out of which answered that they don't know.

4.5.2 Awareness Level of Software Developers Regarding Environmental Sustainability

This section of the survey refers to check the awareness level of software developers regarding Environmental Sustainability, Environmental Sustainability refers to "refers to satisfaction of human needs through ensuring that there are no harmful effects on the environment during the development and deployment of software". The three-point Likert Scale ranging from 0-2 (Agree=1, Disagree=2, Don't Know=0) was provided (Explained in Chapter 3).

Table 4.6: Environmental Sustainability Awareness Level Statistics

		1. Software should generate less waste and emissions.	2. Software should be produced that can influence institutions to generate less waste or emission.	3. Software should not impact the plants or animals around it.	4. Software should encourage less energy consumption
N	Valid	109	109	109	109
	Missing	0	0	0	0

This section of the questionnaire includes four questions regarding environmental sustainability to check the awareness level of software developers regarding software sustainability.

Table 4.7: Software should generate less waste and emission

	Frequency	Percent	Valid Percent	Cumulative Percent

Valid	Don't Know	8	7.3	7.3	7.3
	Agree	93	85.3	85.3	92.7
	Disagree	7	6.4	6.4	100
	Total	109	100	100	

To check the awareness level of software developers regarding environmental sustainability 1st question in this section was “software should generate less waste and emission”, 85.3% of the respondents agreed that software should generate less waste and emission while 6.4% disagreed and 7.3% didn’t know about this.

Table 4.8: Software should be produced that can influence institutions to generate less waste or emission

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	4	3.7	3.7	3.7
	Agree	95	87.2	87.2	90.8
	Disagree	10	9.2	9.2	100.0
	Total	109	100.0	100.0	

The second question asked in the environmental sustainability section was “Software should be produced that can influence institutions to generate less waste or emission”, 87.2% of respondents agreed with it while 9.2% disagreed and 3.7% responded as don’t know.

Table 4.9: Software should not impact the plants or animals around it

		Frequency	Percent	Valid Percent	Cumulative Percent

Valid	Don't Know	12	11.0	11.0	11.0
	Agree	69	63.3	63.3	74.3
	Disagree	28	25.7	25.7	100.0
	Total	109	100.0	100.0	

The 3rd question of the environmental sustainability section was that “software should not impact the plants or animals around it”, in which 63.3% agreed with it while 25.7% disagreed and 11.0% responded that they don’t know that either software should not impact animals or plants around it.

Table 4.10: Software should encourage less energy consumption

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	4	3.7	3.7	3.7
	Agree	97	89.0	89.0	92.7
	Disagree	8	7.3	7.3	100.0
	Total	109	100.0	100.0	

“Software should encourage less energy consumption” is the 4th question in the environmental sustainability section to check the awareness level of software developers regarding sustainable software development. In this 89.0% agreed, 7.3% disagreed and 3.7% responded as don’t know against the 4th question asked from them.

4.5.3 Awareness Level of Software Developers Regarding Economic Sustainability

This section of the questionnaire deals with economic sustainability which refers to the “cost involved in development and stakeholder’s long-term investment in terms of software in its value towards society”. The three-point Likert Scale ranging from 0-2 (Agree-1, Disagree-2, Don’t Know-0) was provided (Explained in Chapter 3).

Table 4.11: Economic Sustainability Awareness Level Statistics

		1. Software should create monetary value.	2. Software should build a positive relationship between the business and its customers.	3. Developed software should positively impact the financial situation of their customers and their people/institutions.
N	Valid	109	109	109
	Missing	0	0	0

This section of the questionnaire includes three questions regarding economic sustainability to check the awareness level of software developers regarding software sustainability.

Table 4.12: Software should create monetary value

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	15	13.8	13.8	13.8
	Agree	90	82.6	82.6	96.3
	Disagree	4	3.7	3.7	100.0
	Total	109	100.0	100.0	

The first question in the economic sustainability section asked was that "Software should create monetary value", 82.6% agreed with this question while 3.7% disagreed and 13.8% responded as don't know against this question.

Table 4.13: Software should build a positive relationship between the business and its customers

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	1	.9	.9	.9
	Agree	101	92.7	92.7	93.6
	Disagree	7	6.4	6.4	100.0
	Total	109	100.0	100.0	

The second question asked was "Software should build a positive relationship between the business and its customer", 92.7% of the respondents agreed that software should create/build a positive relationship between business and its customers while 6.4% disagreed with this and .9% responded as don't know.

Table 4.14: Developed software should positively impact the financial situation of their customers and their people/institutions

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	3	2.8	2.8	2.8
	Agree	103	94.5	94.5	97.2
	Disagree	3	2.8	2.8	100.0
	Total	109	100.0	100.0	

The third question which was asked from software developers was "Developed software should positively impact the financial situation of their customers and their people/institutions". In response to this question, 94.5% agreed that software should provide benefits to the customers while 2.8% disagreed and 2.8% responded as don't know.

4.5.4 Awareness Level of Software Developers Regarding Technical Sustainability

This section asked questions related to technical sustainability which refers to "possesses the characteristics that keep the software alive for a long time and flexible enough to respond the changing technological advancement". The three-point Likert Scale ranging from 0-2 (Agree=1, Disagree=2, Don't Know=0) was provided (Explained in Chapter 3).

Table 4.15: Technical Sustainability Awareness Level Statistics

		1. "Software should be used for a long time."	2. "Software can easily be used by the first-time users (technical/non-technical)."	3." It should be easy to add substantial new features."	4. "Software should be adapted to fit new usage scenarios."	5. "Assets controlled by the software are not easy to be attacked."
N	Valid	109	109	109	109	109
	Missing	0	0	0	0	0

This section of the questionnaire includes five questions regarding technical sustainability to check the awareness level of software developers regarding sustainable software development.

Table 4.16: Software should be used for a long time

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	3	2.8	2.8	2.8
	Agree	88	80.7	80.7	83.5
	Disagree	18	16.5	16.5	100.0
	Total	109	100.0	100.0	

The technical section includes five questions to check the awareness level of software developers regarding sustainable software development, the first question asked was that “Software should be used for a long time”. In this 80.9% of respondents agreed in the favor of these questions while 16.5% disagreed and 2.8% responded that they don’t know that either software should be built/developed for a longer period or not.

Table 4.17: Software can easily be used by first-time users (technical/non-technical)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	3	2.8	2.8	2.8
	Agree	91	83.5	83.5	86.2
	Disagree	15	13.8	13.8	100.0
	Total	109	100.0	100.0	

The 2nd question in the technical sustainability domain which asked was that “Software can easily be used by the first-time users (technical/non-technical)”, in which response 83.5% of respondents agreed that we should develop software which can be easily used by technical and non-technical users of the system at the first time while 13.8% disagreed and 2.85 responded as don’t know.

Table 4.18: It should be easy to add substantial new features

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	4	3.7	3.7	3.7
	Agree	97	89.0	89.0	92.7
	Disagree	8	7.3	7.3	100.0
	Total	109	100.0	100.0	

In technical sustainability to check the awareness level of software developers regarding sustainable software development 3rd question which we asked was "It should be easy to add substantial new features". In the response to this question 89.0% of respondents agreed, 7.3% disagreed while 3.7% answered don't know.

Table 4.19: Software should be adapted to fit new usage scenarios

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	6	5.5	5.5	5.5
	Agree	89	81.7	81.7	87.2
	Disagree	14	12.8	12.8	100.0
	Total	109	100.0	100.0	

The fourth question asked was "Software should be adapted to fit new usage scenarios", 81.7 agreed, 12.8% disagreed and 5.5% responded as don't know against the question asked.

Table 4.20: Assets controlled by the software are not easy to be attacked

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	6	5.5	5.5	5.5
	Agree	79	72.5	72.5	78.0
	Disagree	24	22.0	22.0	100.0
	Total	109	100.0	100.0	

The fifth and last question asked from respondents related to their awareness level regarding sustainable software development in the technical sustainability section was “Assets controlled by the software are not easy to be attacked”. In response to the question, 72.5% of respondents agreed that software should be secure and hard to attack while 22.0% disagreed and 5.5% of respondents answered that they don’t know.

4.6 Challenges in Sustainable Software Development

Questions were asked from the software developers to enlist the challenges developers face during sustainable software development. We enlisted certain challenges against each sustainability domain and asked the developers that either they face those challenges or not.

4.6.1 Environmental Sustainability Challenges Analysis

This section of the questionnaire includes questions related to environmental sustainability challenges. The five-point Likert Scale ranging from 0-4 (Strongly-Agree=1, Agree=2, Strongly-Disagree=3, Disagree=4, Don’t Know=0) was provided (Explained in Chapter 3).

Table 4.21: Environmental Sustainability Challenges Statistics

		1. While developing the software, do you face the challenge of lack of practices for getting the sustainable requirements.	2. While developing sustainable software, do you face the challenge of high energy consumption (process, resources, and the product itself).	3. While developing the software, do you face the challenge of high carbon emission throughout the software development.
N	Valid	109	109	109
	Missing	0	0	0

The environmental sustainability challenges section of the questionnaire includes three questions. Which were asked to check either software developers face these enlisted challenges during sustainable software development or not.

Table 4.22: While developing the software, do you face the challenge of lack of practices for getting the sustainable requirements

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	4	3.7	3.7	3.7
	Strongly Agree	41	37.6	37.6	41.3
	Agree	52	47.7	47.7	89.0
	Strongly Disagree	4	3.7	3.7	92.7
	Disagree	8	7.3	7.3	100.0
	Total	109	100.0	100.0	

37.6% strongly agreed and 47.7% agreed that they face the challenge of lack of practices for getting the sustainable requirements, while 3.7% strongly disagreed and 7.3%

disagreed that they don’t face the challenge of “lack of practices for getting the sustainable software requirements”. And 3.7% of respondents responded as don’t know.

Table 4.23: While developing sustainable software, do you face the challenge of high energy consumption (process, resources, and the product itself)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	2	1.8	1.8	1.8
	Strongly Agree	35	32.1	32.1	33.9
	Agree	59	54.1	54.1	88.1
	Strongly Disagree	2	1.8	1.8	89.9
	Disagree	11	10.1	10.1	100.0
	Total	109	100.0	100.0	

The second question was asked to check that either developer faces the challenge of energy consumption or not during software development. In response to this question, 32.1% strongly agreed and 54.1% agreed that they face the challenge of high energy consumption during software development. 1.8% strongly disagreed and 10.1% disagreed that they don’t face the challenge of high energy consumption, while 1.8% respond as don’t know that either they face this or not.

Table 4.24: While developing the software, do you face the challenge of high carbon emission throughout the software development

		Frequency	Percent	Valid Percent	Cumulative Percent
	Don't Know	27	24.8	24.8	24.8
	Strongly Agree	16	14.7	14.7	39.4

Valid	Agree	15	13.8	13.8	53.2
	Strongly Disagree	13	11.9	11.9	65.1
	Disagree	38	34.9	34.9	100.0
	Total	109	100.0	100.0	

The third question in the environmental sustainability challenges section was used to ask developers that either they face the challenge of high carbon emission throughout the software development or not. In the response to this question, 14.7% strongly agreed and 13.8% agreed that they face this challenge. While 11.9% strongly disagreed and 34.9% disagreed that they don't face this challenge. And 24.8 respondents responded as "Don't know".

4.6.2 Economic Sustainability Challenges Analysis

This section of the questionnaire includes questions related to economic sustainability challenges. The five-point Likert Scale ranging from 0-4 (Strongly-Agree=1, Agree=2, Strongly-Disagree=3, Disagree=4, Don't Know=0) was provided (Explained in Chapter 3).

Table 4.25: Economic Sustainability Challenges Statistics

		1. To have sustainable development, do you face the challenge of poor software design (architectural, logical, physical, and user interface).	2. To have sustainable software, do you face the challenge of lack of ICTs for coordination and communication.	3. While developing sustainable software, do you face the challenge of high resource requirements.
N	Valid	109	109	109

	Missing	0	0	0

The economic sustainability challenges section of the questionnaire includes three questions. Which were asked to check either software developers face these enlisted challenges during sustainable software development or not.

Table 4.26: To have sustainable development, do you face the challenge of poor software design (architectural, logical, physical, and user interface)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	4	3.7	3.7	3.7
	Strongly Agree	29	26.6	26.6	30.3
	Agree	58	53.2	53.2	83.5
	Strongly Disagree	3	2.8	2.8	86.2
	Disagree	15	13.8	13.8	100.0
	Total	109	100.0	100.0	

The first question asked in this section was related to either developer face the challenge of poor software design (architectural, logical, physical, and user interface), in which 26.6% strongly agreed and 53.2% agreed that they face the challenge of poor software design during software development. And 2.8% strongly disagreed and 13.8% disagreed that they don't face this challenge during the development process and 3.7% responded as "don't know".

Table 4.27: To have sustainable software, do you face the challenge of lack of ICTs for coordination and communication

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	14	12.8	12.8	12.8
	Strongly Agree	19	17.4	17.4	30.3
	Agree	52	47.7	47.7	78.0
	Strongly Disagree	9	8.3	8.3	86.2
	Disagree	15	13.8	13.8	100.0
	Total	109	100.0	100.0	

The second question in the economic sustainability challenges section was that do developers face the challenge of lack of ICTs for coordination and communication during software development, 17.4% strongly agreed and 47.7% agreed that they face the challenge of lack of ICTs for coordination and communication. While 8.3% strongly disagreed and 13.8% disagreed that they face this challenge and 12.8% of respondents respond as don't know.

Table 4.28: While developing sustainable software, do you face the challenge of high resource requirements

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	3	2.8	2.8	2.8
	Strongly Agree	39	35.8	35.8	38.5
	Agree	51	46.8	46.8	85.3
	Strongly Disagree	4	3.7	3.7	89.0

	Disagree	12	11.0	11.0	100.0
	Total	109	100.0	100.0	

In the economic sustainability challenges section, the third question which was asked was that does developers face the high resource requirement challenges. About 35.8% strongly agreed and 46.8% agreed that they face the challenge of high resource requirement, while 3.7% strongly disagreed and 11.0% disagreed that they face this challenge. 2.8% out of which responded as "don't know".

4.6.3 Technical Sustainability Challenges Analysis

This section of the questionnaire includes questions related to technical sustainability challenges. The five-point Likert Scale ranging from 0-4 (Strongly-Agree=1, Agree=2, Strongly-Disagree=3, Disagree=4, Don't Know=0) was provided (Explained in Chapter 3).

Table 4.29: Technical Sustainability Challenges Statistics

		1. To have sustainable software development, do you face the challenge of lack of coding standards.	2. Do you face the challenge of a lack of green software development knowledge for sustainable software development?	3. While developing sustainable software, do you face the challenge of the use of a throw-away prototyping strategy	4. For developing sustainable software, do you face the challenge of redundant copies of data.	5. For developing sustainable software, do you face the challenge of poor documentation.
N	Valid	109	109	109		
	Missing	0	0	0		

The technical sustainability challenges section of the questionnaire includes five questions related to challenges developers face during software development. Which were designed to check either software developers face these enlisted challenges during software development or not.

Table 4.30: To have sustainable software development, do you face the challenge of lack of coding standards

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	1	.9	.9	.9
	Strongly Agree	35	32.1	32.1	33.0
	Agree	50	45.9	45.9	78.9
	Strongly Disagree	3	2.8	2.8	81.7
	Disagree	20	18.3	18.3	100.0
	Total	109	100.0	100.0	

The first question asked in the technical sustainability section deals with that do developers face the challenge of a lack of coding standards. In response to this question, 32.1% strongly agreed and 45.9% agreed that they face the challenge of lack of coding standards, while 2.8% strongly disagreed and 18.3% disagreed that they face the challenge of lack of coding standards. And 0.9% responded as “don’t know”.

Table 4.31: Do you face the challenge of lack of green software development knowledge for sustainable software development

		Frequency	Percent	Valid Percent	Cumulative Percent
	Don't Know	26	23.9	23.9	23.9

Valid	Strongly Agree	25	22.9	22.9	46.8
	Agree	38	34.9	34.9	81.7
	Strongly Disagree	3	2.8	2.8	84.4
	Disagree	17	15.6	15.6	100.0
	Total	109	100.0	100.0	

The second question that does software developers face the challenge of lack of green software development knowledge for sustainable software development, 22.9% strongly agreed and 34.9% agreed that they face this challenge. Strongly disagreed 2.8% and 15.6 % disagreed that they don’t face the challenge of green software development knowledge and 23.9% were not aware of this challenge.

Table 4.32: While developing the sustainable software, do you face the challenge of the use of throw-away prototyping strategy

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	24	22.0	22.0	22.0
	Strongly Agree	22	20.2	20.2	42.2
	Agree	33	30.3	30.3	72.5
	Strongly Disagree	4	3.7	3.7	76.1
	Disagree	26	23.9	23.9	100.0
	Total	109	100.0	100.0	

The third question in the technical sustainability challenges section was that either developer faces the challenges of the use of throw-away prototyping strategy, the respondents who strongly agreed was 20.2% and 30.3% out of them agreed that they face throw-away prototyping strategy challenge. While strongly disagreed 3.7% and 23.9% disagreed that they face this challenge and 22.0% responded as "don't know".

Table 4.33: For developing sustainable software, do you face the challenge of redundant copies of data

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	4	3.7	3.7	3.7
	Strongly Agree	30	27.5	27.5	31.2
	Agree	50	45.9	45.9	77.1
	Strongly Disagree	8	7.3	7.3	84.4
	Disagree	17	15.6	15.6	100.0
	Total	109	100.0	100.0	

The fourth question in the technical sustainability challenges section was that does software developers face the challenge of redundant copies of data during the development process of software, 27.5% strongly agreed and 45.9% agreed that they face the challenge of redundant copies of data. 7.3% strongly disagreed and 15.6% disagreed that they don't face this challenge. Out of a total of 3.7%, respondents responded as "don't know".

Table 4.34: For developing sustainable software, do you face the challenge of poor documentation

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	4	3.7	3.7	3.7
	Strongly Agree	44	40.4	40.4	44.0
	Agree	45	41.3	41.3	85.3
	Strongly Disagree	5	4.6	4.6	89.9
	Disagree	11	10.1	10.1	100.0
	Total	109	100.0	100.0	

The fifth question was that does developers face the challenge of poor documentation during software development, 40.4% strongly agreed and 41.3% agreed that they face the challenge of poor documentation. While 4.6% strongly disagreed and 10.1% disagreed that they face this challenge. And 3.7% responded as "don't know".

4.6.4 Social Sustainability Challenges Analysis

This section of the questionnaire includes questions related to social sustainability challenges. The five-point Likert Scale ranging from 0-4 (Strongly-Agree=1, Agree=2, Strongly-Disagree=3, Disagree=4, Don't Know-0) was provided (Explained in Chapter 3).

Table 4.35: Social Sustainability Challenges Statistics

		1. For developing sustainable software, do you face the challenge of poor understanding of the system.	2. For developing sustainable software, do you face the challenge of lack of social and ethical responsibility.	3. For developing sustainable software, do you face the challenge of lack of government support towards green software development.
N	Valid	109	109	109
	Missing	0	0	0

The social sustainability challenges section of the questionnaire includes three questions related to challenges developers face during software development. Which were designed to check either software developers face these enlisted challenges during software development or not.

Table 4.36: For developing sustainable software, do you face the challenge of poor understanding of the system

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	4	3.7	3.7	3.7
	Strongly Agree	38	34.9	34.9	38.5
	Agree	46	42.2	42.2	80.7
	Strongly Disagree	3	2.8	2.8	83.5
	Disagree	18	16.5	16.5	100.0
	Total	109	100.0	100.0	

In the social sustainability section's first question 34.9% strongly agreed and 42.2% agreed that they face the challenge of poor understanding of the system. While 2.8% strongly disagreed and 16.5% disagreed that they don't face the challenge of poor system understanding. And 3.7% 's response was "don't know".

Table 4.37: For developing sustainable software, do you face the challenge of lack of social and ethical responsibility

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Don't Know	11	10.1	10.1	10.1
	Strongly Agree	17	15.6	15.6	25.7
	Agree	41	37.6	37.6	63.3
	Strongly Disagree	16	14.7	14.7	78.0
	Disagree	24	22.0	22.0	100.0
	Total	109	100.0	100.0	

The second question deals with the challenge of lack of social and ethical responsibility, 15.6% strongly agreed and 37.6% agreed that they face this challenge. And 14.7% strongly disagreed and 22.0% disagreed that they don't face the challenge of lack of social and ethical responsibilities, and 10.1% wasn't aware of this challenge.

Table 4.38: For developing sustainable software, do you face the challenge of lack of government support towards green software development

		Frequency	Percent	Valid Percent	Cumulative Percent
	Don't Know	10	9.2	9.2	9.2

Valid	Strongly Agree	14	12.8	12.8	22.0
	Agree	31	28.4	28.4	50.5
	Strongly Disagree	27	24.8	24.8	75.2
	Disagree	27	24.8	24.8	100.0
	Total	109	100.0	100.0	

The last question asked in the social sustainability section was does software developers face the challenge of lack of government support towards green software development. The respondents who strongly agreed were 12.8% and 28.4% agreed that they face the challenge of lack of government support. While 24.8% strongly disagreed and 24.8% disagreed that they face this challenge. And 9.2% of respondents responded as “don’t know”.

4.7 Challenges

Table 4.39 shows the enlisted challenges which developers face during sustainable software development.

Table 4.39: Enlisted Challenges, Adopted from [19]

No	Challenges
1	“Lack of requirement gathering practices”
2	“High energy consumption”
3	“High carbon emission”
4	“Poor software design (architectural, logical, physical and user interface)”
5	“Lack of ICTs for coordination and communication”
6	“High resource requirement”
7	“Lack of coding standards”
8	“Lack of green software development knowledge”

9	“Throw away prototyping strategy”
10	“Redundant copies of data”
11	“Poor documentation”
12	“Poor understanding of the system”
13	“Lack of social and ethical responsibility”
14	“Lack of government support towards green software development”

Table 4.39 shows the enlisted challenges which developers face during sustainable developers face as it was also proved through the survey conduction, that most of the software developers face these challenges while doing sustainable software development. Most of the developers strongly agreed and agreed that they face these enlisted challenges while doing or following sustainable software development. To overcome these challenges strategies or solutions are also provided by software developers during survey conduction.

4.8 Strategies against Challenges

Strategies against the challenges which developers face during sustainable software development are provided by the respondent of the survey who is software developers and those strategies are also discussed.

4.8.1 Strategies against Environmental Sustainability Challenges

Table 4.40 shows the strategies to overcome the environmental sustainability challenges. It comprises two columns namely ‘R_ID (Respondent_ID)’ and ‘Strategies to Overcome Challenge’.

Table 4.40: Strategies to Overcome Environmental Sustainability Challenges

R_ ID	Strategies to Overcome Lack of Requirements gathering Practices Challenge	R_ ID	Strategies to Overcome High Energy Consumption Challenge	R_ ID	Strategies to Overcome High Carbon Emission Challenge
R12	Use Software Development Life Cycle	R2	Green SDLC Models	R12	Use suitable Green Software development life cycle
R34	Use Good Requirements Gathering Techniques / Practices	R39	Follow Software development Life Cycle for Green IT	R15	Green Cloud technique
R38	Agile Methodology	R49	Agile Methodology	R59	Green Software development process
R48	Constant communication Among stakeholders	R50	Agile Approach	R 70	Green Software development techniques/Models can be used to overcome this challenge
R49	Agile Methodology	R58	Simple & clear process /Green software development Methodology	R 76	Green SDLC
R50	Agile Methodology	R80	Follow Green Development Method while developing	R80	Green SDLC
R52	Agile Methodology			R87	Green development Methodology
R77	Learn more Knowledge about the problem			R 102	Green Software method
R79	Requirements gathering techniques				
R81	Agile Methodology				
R83	Agile Methodology				
R85	SDLC				
R89	SDLC				
R95	Requirement Prioritization				
R97	Brainstorming, Observation, Interviews				
R98	Agile Methodology				

- **Lack of Requirement gathering Practices Challenge**

1. While developing the software, do you face the challenge of lack of practices for getting the sustainable requirements.

- **Strategies to Overcome Lack of Requirement gathering Challenge:**

Requirements gathering is a very important part of software development. More than 70% of projects failed due to wrong requirements, and lots of people face this challenge during software development which causes software failures. So, requirements should be the priority while developing the software, and a lot of focus should be spent on requirements, some of the strategies as suggested by the respondents to deal with this challenge which is as follow:

I. ***Use Software Development Life Cycle (SDCL):*** SDCL is a defined standard or procedure used for properly developing the software. So, to overcome the challenge of lack of practices for getting the sustainable requirement SDCL (Software Development Life Cycle) can be used. Different software development life cycles are very useful in requirements gathering like agile development methodology is one of them.

II. ***Use Good Requirement Gathering Techniques:*** Requirement gathering techniques can be a good strategy to overcome the challenge of lack of practices for getting sustainable requirements. There are different requirements techniques which can be very useful in requirements gathering such as Brainstorming, interviews, One to One Meetings, Observations, Focus Group and through survey or questionnaire.

III. ***Constant Communication among Stakeholders:*** For better requirements, there should be constant communication among stakeholders. Conduct one-to-one meetings or group meetings with stakeholders to get proper and clear requirements without any confusion. This also can be a good strategy for requirements gathering and better system understanding.

IV. ***Agile Methodology:*** Agile methodology is a practice that in which continuous iteration of development and testing happens throughout the software development. Similarly, in agile developers keep updating requirements between the iterations, keep prioritizing the requirements while gathering the requirements.

V. ***Learn more knowledge about the Field:*** Learn more knowledge about the field means do complete and thorough research regarding the software you want to develop and get detailed knowledge about the product that what are the main requirements for the product you want to develop.

VI. ***Requirement Prioritization:*** Prioritizing requirements means ordering the requirements based upon their importance, the most important requirements should be ordered higher in the list and less important at the later part of the list.

VII. ***Brainstorming, Observations, and Interviews:*** These are requirement gathering techniques, requirements can be gathered using these techniques depending upon the nature of the product. Brainstorming is used in requirement gathering to get as many ideas or requirements as possible from a group of people. And interviews are in which you get requirements by interviewing the stakeholder.

- **High Energy Consumption Challenge**

2. “While developing the sustainable software, do you face the challenge of high energy consumption (process, resources, and the product itself)”.

- **Strategies to Overcome High Energy Consumption Challenge**

High energy consumption is among one of the emerging challenges in "Sustainable Software Development" which causing huge impacts on the environment and is very necessary to overcome this challenge. To overcome this software industry should monitor and follow power-saving software strategies. There are some strategies suggested by respondents of the survey we conducted. Which are as follows:

I. ***Green Software Development Life Cycle:***

Green software development life cycle is an approach used to develop green software which can be helpful in saving energy consumption or can be helpful for low energy consumption. GREENSOFT Model is a model which is presented in recent research which can be helpful in developing green software [32].

II. ***Follow Software Development Life Cycle for Green IT:*** As mentioned in the above strategy similar software development life cycles can be used for sustainable software development to overcome the high energy consumption challenge while developing the software.

III. ***Agile Methodology:*** As the agile methodology is a well-established software development life cycle which is software developers are successfully using for software development. Similarly, agile methodology can be used by keeping in mind the green software requirements or keeping in mind the green software requirements just like GREENSOFT.

- **High Carbon Emission Challenge**

- **Strategies to Overcome High Carbon Emission Challenge**

High carbon emission is causing a huge impact on the environment during the software development process.

I. ***Use Green Software Development Life Cycle/Green SDLC:*** The software is developed designed and deployed can have a major impact on energy consumption which causes carbon emission. Due to a lot of carbon emission environment is the among the most affected things. So, by using a green software development life cycle we can manage less carbon emission and can lessen the effects on the environment.

II. ***Green Cloud Technique:*** Most of the recent applications are mostly deployed on the cloud, the rapid growth in the cloud has resulted in the expansion of data centers. Data centers almost consume 2% of the world's total electricity and by 2030 it can reach 8% of the total energy. So, by making cloud green the carbon emission challenge can be dealt with. As we first were focusing on making hardware green but now the focus has been shifted to green software so following the green software development processes and greener server architecture will likely improve the less carbon emission and carbon footprints can be reduced in the environment [33].

4.8.2 Strategies against Economic Sustainability Challenges

Table 4.41 shows the strategies to overcome the economic sustainability challenges, it comprises two columns namely ‘R_ID’ and ‘Strategies to Overcome the Challenges’.

Table 4.41: Strategies to Overcome Economic Sustainability Challenges

R_ID	**Strategies to Overcome “Poor software design (architectural, logical, physical and user interface) Challenge”**	**R_ID**	**Strategies to Overcome “Lack of ICTs for Coordination and Communication Challenge”**	**R_ID**	**Strategies to Overcome “High Resource Requirement Challenge”**
R3	UML Diagram	R12	Use managerial skills and communicate online with stakeholders	R2	Project Management and Planning

R8	Hire Professional Software Design Engineer	R76	Make group in any free online software to communicate	R3	Proper Resource Management and Planning
R12	SDLC	R91	Proper communication	R50	Planning and Management
R15	Detail Explanation system through UML diagrams	R93	Communication with stakeholders through proper channels and in understandable technical terms	R79	HR Planning
R16	UML Diagram	R97	Use video Conferencing tools	R89	Project Manager
R49	Agile Methodology			R100	Project resource management
R52	Agile Methodology during UI			R105	Hire Professional
R53	Follow Proper SDLC				
R79	Follow Designing Standards				
R83	Follow Designing Principle				
R92	SDLC				
R97	UML Diagrams				
98	SDLC				

- **“Poor Software Design (architectural, logical, physical and user interface)” Challenge**

1. To have sustainable development, do you face the challenge of poor software design (architectural, logical, physical, and user interface).

- **Strategies to Overcome “Poor software Design (architectural, logical, physical and user interface)” Challenge**

Design is a very important stage of software development; a project's success is also big-time depends on the design of the software. To deal with the challenge of poor software design certain strategies have been suggested by the respondents of the survey, which are as follows:

I. ***UML Diagram:*** Unified Modeling Language is a diagram based on the UML with visually representing a system with its main artifacts, roles or character, users, actors to make the system understand better and clear. To deal with the challenge of poor understanding of the system a UML diagram can be helpful as a system can be clearly described at the early stage and can be discussed and can be shown to the stakeholders and clients of the system that what you are going to build. So, it will save money as you won't have to change the system requirements at a later stage and build the entire system or change the design at the later stage of the development process and spend more money.

II. ***Hire a Professional Software Design Engineer & Follow Designing Principles:*** A professional software design engineer can help deal with the design challenge. As he can design the system properly by following the proper designing guidelines and designing modules more efficiently and properly. A professional designer can design the software more efficiently and can follow principles and guidelines and develop a system successfully within a limited budget.

III. ***SDLC & Agile Approach:*** Software development life cycle or agile development approach can be used. As agile is an iterative developing process so u can keep updating design changes at every iteration if needed. It can also save money and time and can make the project successful.

- **"Lack of ICTs for Coordination and Communication" Challenge**

2. To have sustainable software, do you face the challenge of "lack of ICTs for coordination and communication".

- **Strategies to Overcome “Lack of ICTs for Coordination and Communication” Challenge**

It has been observed that face to face meeting and traveling during software development process has adverse impacts on economy and environment. So, some strategies have been discussed to deal with this challenge. Which are as follows:

I. ***Use Modern Communication Technologies:*** Instead of using means of traveling for face-to-face meetings again and again use online communication technologies for communication between the stakeholders. so that impacts on money and the economy can be reduced by following this methodology. Groups can be made on any software and communication can be performed between teams, clients, and management. Proper and regular communication can be continuing among stakeholders to reduce confusion related to software. Or we can say the technique of crowdsourcing can be used to overcome this challenge very efficiently.

- **High Resource Requirement Challenge**

3. While developing sustainable software, do you face the challenge of high resource requirements.

- **Strategies to Overcome High Resource Requirement Challenge**

High resource requirement is also one of the enlisted challenges which software developers face during sustainable software development. To deal with or overcome this challenge some of the strategies have been given by survey respondents which are as follows:

I. ***Project Management and Planning:*** To overcome the challenge of high resource requirements one of the offered strategies is to properly do project management and planning. Use previous project knowledge for planning and estimation of resources required for a certain project. Prioritize the deliverable and keep in continue communication with stakeholders and inform them if any resource is needed. To control the lack of resource problem you need to keep track of all resources throughout the project. You can track the availability of resources, manage workload, manage work schedule. For this, you can use project management software to keep track of all the resources and timelines.

II. ***Hire Professional:*** Hire a professional project manager who can plan and hire professional and skilled employees relevant to the project and field and who can plan and manage the resource properly. And who will manage the project progress, workload, schedules, tools required, and other resources within the limited budget in the best way possible.

4.8.3 Strategies against Technical Sustainability Challenges

Table 4.42 shows the strategies to overcome the technical sustainability challenges during sustainable software development which developers face. It comprises two columns namely ‘R_ID’ and ‘Strategies to Overcome Challenges.

Table 4.42: Strategies to Overcome Technical Sustainability Challenges

R_ID	**Strategies to Overcome Lack of coding standards Challenge**	**R_ID**	**Strategies to Overcome Lack of green software development knowledge Challenge**	**R_ID**	**Strategies to Overcome Throw away prototyping strategy Challenge**	**R_ID**	**Strategies to Overcome Redundant copies of data Challenge**	**R_ID**	**Strategies to Overcome Poor documentation Challenge**
R1	Better Coding Standards	R3	Increase knowledge of developers by enrolling them in green software development courses.	R4	Delete the Previous Prototype	R1	Better code optimization	R8	Follow Proper SDLC
R5	Train and teach better coding standards to developers	R5	Conduct Seminars	R20	Use Versioning System	R3	Optimize the Code	R10	SDLC

R8	Knowledge of upcoming technology and coding standards	R 50	Seminars and Conferences	R 38	Use Github	R 6	Data Optimization/Code Optimization	R 14	Agile Methodology
R9	Guidelines should be provided across the organization for coding standards	R 75	Training	R 87	Use Github	R 9	Use Normalization techniques	R 35	Detailed FRD before execution of software /starting development.
R 15	Documentation	R 79	Give proper knowledge to developers about green software development	R 100	Delete the Previous Prototype	R 12	Use normalization technique	R 48	Should be following standards for proper documentation
R 16	Proper Documentation should be done	R 83	Training	R 108	Use Versioning System	R 35	Code Optimization	R 49	Agile approach
R 35	Use best practices and standards	R 89	Tools and Techniques/models/processes			R 83	Data Normalization	R 58	Each Functionality should be properly documented
R 49	Documentation	R 100	Seminars/ Conferences			R 100	Use 1NF, 2NF and BCNF	R 81	Follow SDLC
R 52	Extreme Programming							R 97	Document Source Code Properly
R 83	Use proper Coding Standards								
R 89	Iterative behavior should be avoided								
R 100	Agile/ Extreme coding Approach								

- **Lack of Coding Standards Challenge**

1. To have sustainable software development, do you face the challenge of lack of coding standards.

➢ Strategies to Overcome "Lack of Coding Standards" Challenge

"Lack of coding standards" is also one of the leading challenges in sustainable software development. Redundancy, nested loops, and uncontrolled data flow are poor coding constructs and consume more energy. Therefore, to overcome the sustainability challenge these should be avoided. To deal with this challenge certain strategies have been suggested by the survey respondents which are as follow:

I. ***Better Coding Standards and Documentation:*** To overcome the lack of coding standard challenge one of the strategies is to follow the coding standard and do proper documentation of the written code. Some of the coding standards are as follow: write comments and documentation, write readable yet efficient code, limited use of global variables, use less nested loops, follow a naming convention, indentation avoid lengthy functions, follow language-specific conventions, reusability, portability, scalability and keep the code simple. The code should be well documented and easy to untestable for other developers to changes if required in the future.

II. ***Use Extreme Programming Technique:*** Extreme programming is an "Agile Methodology" or "Agile software development framework" which aims to create a high-quality product and higher quality life for the development team. Values of extreme programming are as follows: Communication, Feedback, Respect, Simplicity, and courage. Some practices are used in extreme programming. Some of those practices are weekly cycle, team planning, quarterly cycle, test-first programming, pair programming, etc.

➢ "Lack of Green Software Development Knowledge" Challenge

2. Do you face the challenge of "lack of green software development knowledge for sustainable software development"?

- **Strategies to Overcome “Lack of Green Software Development” Challenge**

There is still a lack of knowledge among software developers regarding sustainable software development. To overcome this challenge strategies are discussed below.

I. ***Conduct Seminars or Conferences and Training:*** To overcome the challenge of lack of sustainable development knowledge among the software developers' seminars and conferences can be conducted to provide the proper knowledge and guide them about sustainable software development and its importance. Organizations can also train their employees by enrolling them in green software development-related courses or by training them to follow the green development life cycle.

- **Throw Away Prototyping Challenge**

3. While developing sustainable software, do you face the challenge of the use of a throw-away prototyping strategy.

- **Strategies to Overcome Throw Away Prototyping Challenge**

I. ***Delete the Previous Prototype:*** Throw away prototyping is used when you are completely unknown to the requirements and you use prototypes to gather requirements and develop a system. So, their multiple prototypes involve against each requirement which can create confusion and problem. This prototype is built, tested, and reworked until accepted. So, while a prototype got accepted delete its previous one.

II. ***Use GitHub/ Versioning System:*** Github is used for versioning control in software development. So, to deal with the challenge of throw-away prototyping git-hub can be used by creating versions of every prototype. It will become easy to track them without any confusion or redundancy.

- **Redundant Copies of Data Challenge**

4. For developing sustainable software, do you face the challenge of redundant copies of data.

- **Strategies to Overcome Redundant Copies of Data Challenge**

I. ***Better Code Optimization:*** It is a method of increasing code quality and code efficiency through code modification. While project progress and the number of developers working on the project increases which causes repetitions and overlaps starts to take place for which code optimization is the method that can be used to overcome this problem. Code optimization can result in less memory usage and fast processing, it also means the short time required to load a page. It also makes code easy to read and easy to maintain.

II. ***Use Normalization Techniques:*** It's a database normalization technique that is used to lessen redundancy and remove characteristics like 'insertion', 'update', and 'deletion' anomalies. The normal forms which are used for normalization are "1NF (First Normal Form)", "2NF (Second Normal Form)", "3NF (Third Normal Form)", 4NF, 5NF, 6NF, and "BCNF (Boyce-Codd Normal Form)". Normalization helps in developing databases that have less cost and provide better security of data.

- **Poor Documentation Challenge**

5. For developing sustainable software, do you face the challenge of poor documentation.

- **Strategies to Overcome Poor Documentation Challenge**

Documentation is also a challenge developers face during sustainable software development. To deal with these certain strategies are suggested by the software developers we got through survey conduction.

I. ***Document every SDLC Level/ Agile Approach:*** Documentation is considered a necessary evil in software development but it can be a blessing in disguise for follow-on work. For example, if requirements are properly documented you can use that to clear your confusion and to better understand the system by reading the document. Similarly, all other development stages can also be helpful while needed. So SDLC can be helpful to document every level clearly and the project manager should oversee the documentation. The source code used in software development can be well documented by following coding standards which include commenting and others.

4.8.4 Strategies against Social Sustainability Challenges

Table 4.43: Strategies to Overcome Social Sustainability Challenges

R_ID	Strategies to Overcome "Poor understanding of System" Challenge	R_ID	Strategies to Overcome "Lack of social and ethical responsibility" Challenge	R_ID	Strategies to Overcome "Lack of government support towards green software development" Challenge
R8	The requirement should be defined clearly and the language barrier should be omitted	R5	Ethical values are most important	R3	Provide awareness related to green software
R12	Research and gather domain knowledge	R9	The software developer must be aware of social and ethical responsibilities (Use of information, protection of data, intellectual property, Copyrights, license agreement)	R7	For Green, Software development take agreement and proper document from the government side
R15	Read requirement documentation again	R50	Software should be secure to secure user information	R34	Govt needs to provide a platform or provide government-defined strategies
R41	Schedule meeting with the client for better system understanding	R87	Define Organization Policy	R62	Government defined policies
R52	Better requirement gathering & understanding	R91	Keep in mind all the legal requirements and human safety	R73	Provide Guidelines / Policies
R75	Properly get requirements from user	R97	Follow international copyrights standards	R95	Conduct conferences related to this

R77	Better requirement gathering and explaining	R100	Define Policies	R104	Follow International Defined Policies
R100	Agile Methodology	R108	Define Organization Policy		

- **Poor Understanding of System Challenge**

1. For developing sustainable software, do you face the challenge of poor understanding of the system.

- **Strategies to Overcome Poor Understanding of System Challenge**

While developing sustainable software developers do face the challenge of poor understanding of the system. To overcome this challenge following strategies have been suggested by the software developers while conducting the survey.

I. ***Define Clear Requirements:*** Requirements are a very important part of every project; the success and failure of a project depend on the requirements. If the requirements are correct the system will be developed according to the wish of the client and stakeholders. So, to overcome the challenge of poor system understanding requirements should be clear and every language barrier should be omitted. Requirements should be defined, easy to understand, to the point, and shouldn’t be ambiguous.

II. ***Research and Gather Domain Knowledge:*** Before starting the software development proper research related to the system should be performed to fully understand the requirements of the system.

III. ***Better Requirements gathering And Understanding & Better Communication with Client:*** Use better requirements gathering techniques to gather requirements, interviews, brainstorming, face to face meeting, focus groups, survey and questionnaires, research and observation is among those techniques which can be used to gather better requirements and understanding the system in a better way and develop it.

IV. ***Agile Approach:*** Agile Methodology can also be a successful strategy to deal with the challenge of poor understanding of the system. Agile methodology is an iterative process so at every iteration you can add new requirements if needed or required and can be added easily. You can show the system to stakeholder after each iteration and can discuss it weather any change required or not.

- **Lack of Social and Ethical Responsibilities Challenge**

2. For developing sustainable software, do you face the challenge of lack of social and ethical responsibility.

- **Strategies to Overcome Lack of Social and Ethical Responsibilities Challenge**

I. ***Must be Aware of Ethical and Social Values:*** The software developer who is developing the software must be aware of the ethical and social values of the society and he should keep in mind those values during development. Developers should keep in mind the safety of humans and also all the legal regularities should be considered while developing the software. The developer should keep in mind the issues related to privacy, accuracy, property, accessibility, and effects on quality of life. So, by keeping these things in mind this challenge can be overcome.

II. ***Define Policies/ Follow International Defined Policies:*** The organization should define its policies against the ethical and social responsibilities or can include international ethical and social policies in their company's or organization's charter.

- **“Lack of Government Support Towards Green Software Development” Challenge**

3. For developing sustainable software, do you face the challenge of “lack of government support towards green software development”.

- **Strategies to Overcome “Lack of Government Support Towards Green Software Development” Challenge**

I. ***Government Defined Policies/ Conferences/ Seminars:*** to overcome this challenge government can play its role by defining policies or rules and regulations against sustainable software development and order to all the organizations to follow the rule and regulations and policies defined by the government for sustainable software development. Government can also conduct awareness conferences and seminars related to green IT or sustainable software development.

4.9 Mapping of Strategies (in numbers)/ Level of strategy from various Respondents

By combining all the strategies provided by the respondents of the survey against the challenges which software developers face during sustainable software development the mapping is done as shown in the figure below.

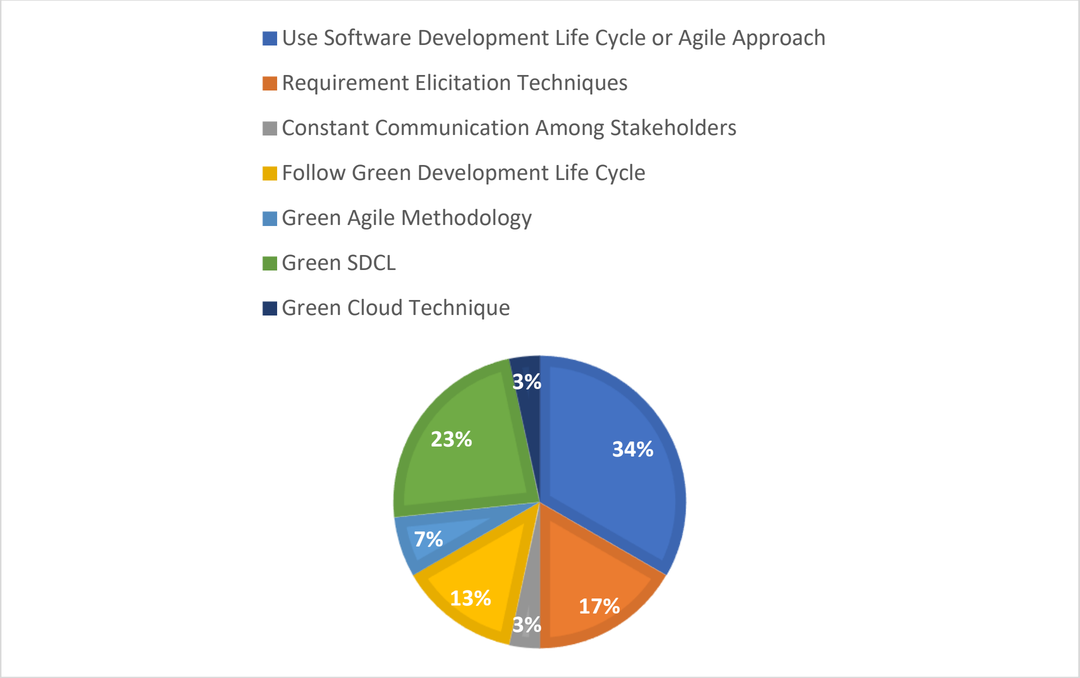

Figure 4.2: Mapping for Environmental Sustainability Strategies

Figure 4.2 shows mapping of strategies against environmental sustainability. It highlights the strategies to overcome the environmental sustainability challenges developers face during sustainable software development.

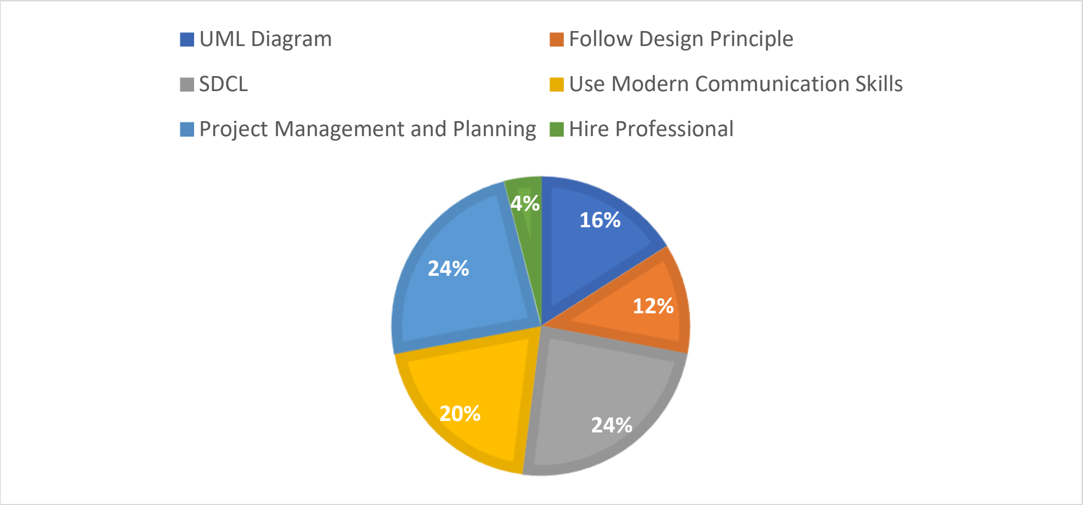

Figure 4.3: Mapping Strategies for Economic Sustainability

Figure 4.3 shows the mapping of strategies for economic sustainability. The strategies /solution to overcome the economic sustainability challenges which were provided by software developers during survey are mapped and showed in Figure 4.3.

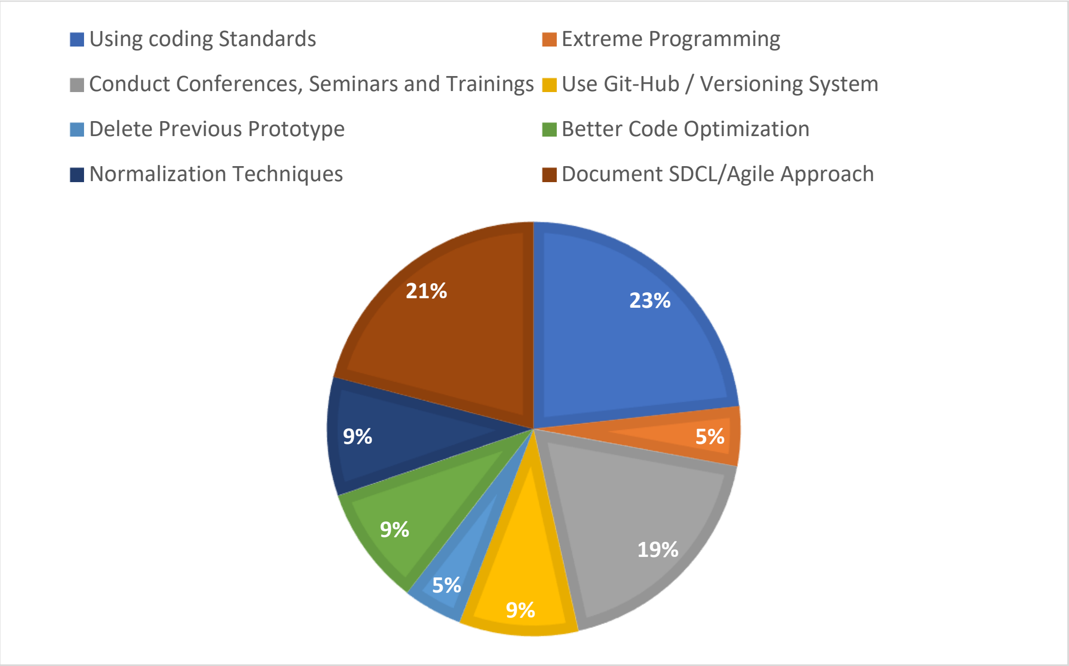

Figure 4.4: Mapping for Technical Sustainability Strategies

Figure 4.4 shows the mapping for technical sustainability strategies. It shows the list of strategies to overcome the technical sustainability challenges software developers face during sustainable software development.

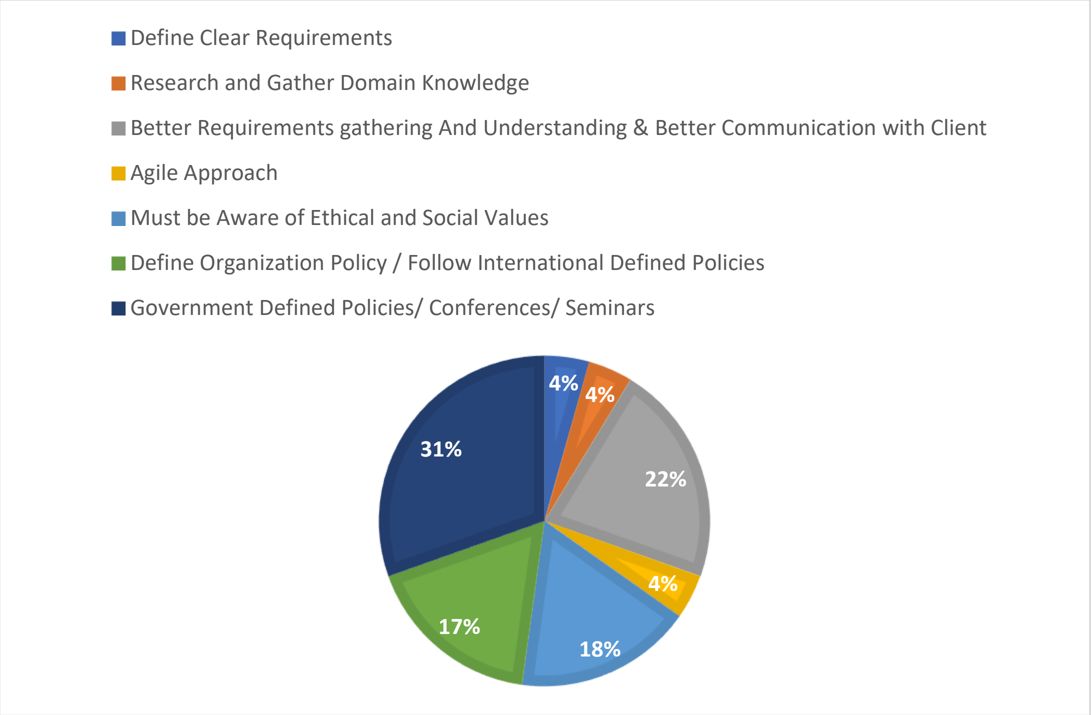

Figure 4.5: Mapping for Social Sustainability Strategies

Figure 4.5 shows the mapping for social sustainability strategies. It highlights the solutions required to deal with or to overcome the challenges software developers face during sustainable software development.

4.10 Towards a Sustainable Software Development Model

By combining all the strategies provided by the respondents of the survey against environmental, economic, social, and technical challenges which software developers face during sustainable software development the mapping is done which leads us towards a sustainable software development model as shown in Figure 4.6.

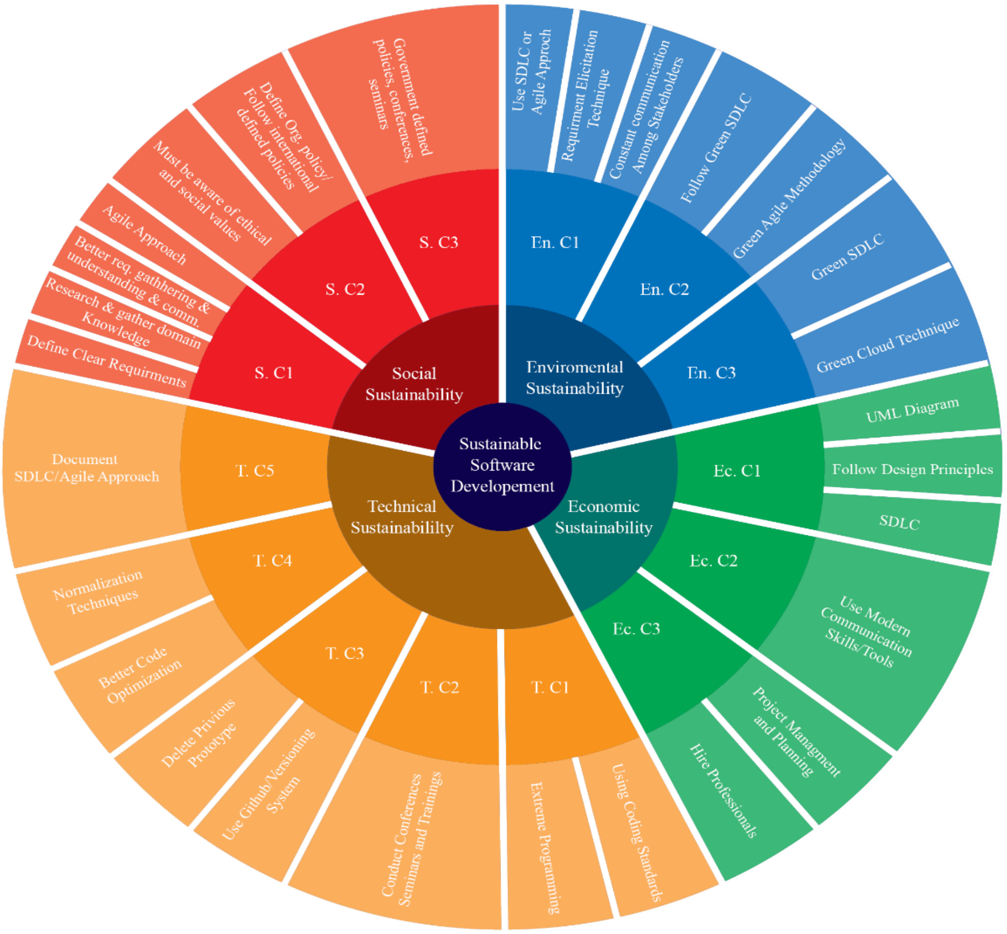

Figure 4.6: Towards a Sustainable Software Development Model

4.11 Chapter Summary

This Chapter has presented the results of survey conduction. The objective of the survey was to highlight the awareness level of software developers regarding software sustainability and check either developer face challenges during sustainable software development and if they face getting solutions or strategies against those challenges. This chapter also discusses the strategies software developers provide to overcome the challenges software developers face during sustainable software development. The mapping of strategies also done in this chapter was provided by the software developers during survey conduction. A sustainable software development model is also provided in this chapter. Chapter 5 discuss the conclusion of the research.

CHAPTER 5

CONCLUSION

5.1 Introduction

Chapter 4 illustrates the results of the survey. This Chapter provides the strategies against the challenges provided by the software developers, developers face during sustainable software development.

The previous chapter also discusses strategies to overcome the challenges software developers face during sustainable software development. Chapter 6 discuss the conclusion of the research.

5.2 Research Summary

This study was set out to explore sustainable software development. The importance of sustainable software development is increasing day by day, and it becoming one of the leading research areas in software engineering. A lot of effort is being inserted into sustainable software development. Our study was also related to sustainable software development, to do so we identified the awareness level of software developers regarding sustainable software development. Enlisted the challenges developers face while developing sustainable software. This study also provides the strategies against those challenges developers face during sustainable software development.

This research sought to answer three research questions.

1. What is the awareness level of sustainability among software developers?
2. What challenges software developers to face for sustainable software development.?
3. What solutions/strategies software developers need to overcome the challenges?

The first research question was answered by highlighting the awareness level of software developers regarding sustainable software development. A survey was conducted to highlight the awareness level among software developers regarding sustainable software development. To do this questionnaire was created and was distributed among software developers. As a result, we got a response from 109 respondents.

The second question was answered by enlisting the challenges software developers face during sustainable software development. The survey was conducted to check that weather software developer faces these enlisted challenges during sustainable software development or not. To do so the questionnaire was created which was consist of the enlisted challenges and was distributed among the software developers working in the industry. As a result, we got a response from 109 respondents during the survey conduction.

The third and last question was answered by giving the solutions/ strategies against the challenge's software developers face during sustainable software development. The survey was conducted and a questionnaire was created and under each challenge, space was given to answer the strategies against those challenges. Similarly, in the result, we got a response from 109 respondents.

5.3 Fulfillment of Research Objectives

Three objectives of this research have been achieved.

This study highlights the awareness level of software developers regarding sustainability that either the software developers who are working in this domain how much they are aware of software sustainability. The results are reported in Chapter 4. By achieving objective one we managed to overcome the gap between the existing studies regarding work on software developer’s awareness levels.

Similarly, this study also enlisted the challenges software developers face and check that software developers face those challenges or not during sustainable software development through survey conduction. The results are reported in Chapter 4. By achieving this objective, we managed to overcome the gap between studies regarding work on challenges which developers face.

This study provided the solutions or strategies against the challenges software developers face during sustainable software development. The results of solution/ strategies are reported in Chapter 4. By achieving this objective, we managed to overcome the gap between studies regarding work on challenges and solutions/ strategies against those challenges which developers face during sustainable software development.

5.4 Limitations of Research

The target audience for surveying this research was software developers working in the industry. Because of the study’s explicit focus on software developers working in the industry, it was found extremely difficult to get an appointment from them. Which might be resulted in a small sample size. Finding respondents extended data collection and other processes. As the online response rate was very low, due to which another effort was done by physically visiting the organizations to get responses from developers. Although it was found that there was still difficulty in getting a response from software developers. Which resulted in smaller sample size, a bigger size might have increased the survey accuracy and reliability. Similarly, in existing studies mostly didn’t conducted survey to conduct the research which is also a limitation of this study.

5.5 Future Work

Sustainable software development is the need of the present world and a lot of work is being done in this domain. The survey was conducted to perform this study so in future work the finding of the survey can be compared with the literature. The survey can also be conducted in multiple industries so that the results can be generalizable and can be validated.

REFRENCES

[1] C. Portela, A. Vasconcelos, S. Oliveira, and M. Souza, "The Use of Industry Training Strategies in a Software Engineering Course: An Experience Report," *Proc. - 30th IEEE Conf. Softw. Eng. Educ. Training, CSEE T 2017*, vol. 2017, pp. 29–36, January 2017.

[2] S. Citation, "Measuring and Sustaining the New Economy, Software, Growth, and the Future of the U.S Economy," *Measures Sustainable New Economies Software, Growth, Future U.S Economy*, 2006.

[3] F. Gurcan and C. Kose, "Analysis of software engineering industry needs and trends: Implications for education," *International Journal Engineering Education.*, vol. 33, no. 4, pp. 1361–1368, 2017.

[4] I. Pearson Education, "The Technical and Social History of Software Engineering," *Technical Society Hist. Software Engineering*, p. 456、453、603, 2014.

[5] P. Lago, "Challenges and Opportunities for Sustainable Software," *Proc. - 5th International Work. Prod. Line Approaches Software Engineering PLEASE 2015*, pp. 1–2, 2015.

[6] M. N. Malik and H. H. Khan, "Investigating Software Standards: A Lens of Sustainability for Software Crowdsourcing," *IEEE Access*, vol. 6, pp. 5139–5150, 2018.

[7] R. Chitchyan, J. Noppen, and I. Groher, "Sustainability in software product lines," *ACM International Conference Proceeding Serial*, vol. 1, no. ii, p. 367, 2014.

[8] R. Chitchyan *et al.*, "Sustainability design in requirements engineering: State of practice," *Proceeding - International Conference Software Engineering*, pp. 533–

542, 2016.

[9] L. Karita, B. C. Mourão, and I. Machado, "Software industry awareness on green and sustainable software engineering: A state-of-the-practice survey," *ACM International Conference Proceeding Series*, pp. 501–510, 2019.

[10] N. Condori-Fernandez and P. Lago, "Towards a software sustainability-quality model: Insights from a multi-case study," *Proceeding - International Conference Res. Challenges Inf. Sci*, pp. 1–11,vol- May 2019.

[11] D. Khara and D. P. Mahapatra, "Organizational behavior engineering for sustainable global software development," *Proceeding - 11th IEEE International Conference Global Software Engineering ICGSE 2016*, pp. 11–15, 2016.

[12] B. Combemale, B. H. C. Cheng, A. Moreira, J. M. Bruel, and J. Gray, "Modeling for sustainability," *Proc. - 8th International Work. Model. Software Engineering MiSE 2016*, pp. 62–66, 2016.

[13] O. Badreddin, "Powering Software Sustainability with Blockchain," *Proc. 28th Annual International Conf.erence Computer Science Software Engineering*, pp. 315–322, 2018.

[14] I. S. Brito, J. M. Conejero, A. Moreira, and J. Araújo, "A concern-oriented sustainability approach," *Proceeding - International Conference Res. Challenges Inf. Sci.*, vol. 2018-May, pp. 1–12, May 2018.

[15] D. Renzel, I. Koren, R. Klamma, and M. Jarke, "Preparing research projects for sustainable software engineering in society," *Proc. - 2017 IEEE/ACM 39th International Conference Software Engineering Software Engineering Society Track, ICSE-SEIS 2017*, pp. 23–32, 2017.

[16] H. S. Zhu, C. Lin, and Y. D. Liu, "A programming model for sustainable software," *Proceeding. - International Conference Software Engineering*, vol. 1, pp. 767–777, 2015.

[17] L. Lefevre, "Full life cycle in sustainable ICT for reaching energy reduction, energy efficiency and energy proportionality," pp. 1–1, 2016.

[18] M. Jagemar and G. Dodig-Crnkovic, “Cognitively Sustainable ICT with Ubiquitous Mobile Services - Challenges and Opportunities,” *Proceeding - International Conference Software Engineering*, vol. 2, pp. 531–540, 2015.

[19] M. Salam and S. U. Khan, “Challenges in the development of green and sustainable software for software multisourcing vendors: Findings from a systematic literature review and industrial survey,” *J. Software Evolution Process,* vol. 30, no. 8, 2018.

[20] B. Penzenstadler *et al.*, “The SusA Workshop - improving sustainability awareness to inform future business process and systems design,” 2020.

[21] J. S. Molléri, K. Petersen, and E. Mendes, “Survey Guidelines in Software Engineering: An Annotated Review,” *International Symp. Empirical Software Engineering Measures*, vol. 08-09-September, 2016.

[22] M. Linåker, Johan; Sulaman, Sardar Muhammad; Maiani de Mello, Rafael; Höst, “Guidelines for conducting surveys in software engineering v. 1.1,” vol. 08, no. pp. 0–63, May 2015.

[23] B. Penzenstadler *et al.*, “Raising awareness for potential sustainability effects in Uganda: A survey-based empirical study,” *CEUR Workshop Proceeding*, vol. 2541, 2019.

[24] S. Nazir, N. Fatima, S. Chuprat, H. Sarkan, F. Nurulhuda, and N. N. A. Sjarif, “Sustainable software engineering: A perspective of individual sustainability,” *Int. J. Advance Science Engineering Information Technology*, vol. 10, no. 2, pp. 676–683, 2020.

[25] S. Oyedeji, A. Seffah, and B. Penzenstadler, “A catalogue supporting software sustainability design,” *Sustainable*, vol. 10, no. 7, pp. 1–30, 2018.

[26] J. A. García-Berna, J. M. C. de Gea, B. Moros, J. L. Fernández-Alemán, J. Nicolás, and A. Toval, “Surveying the environmental and technical dimensions of sustainability in software development companies,” *Applied Science.*, vol. 8, no. 11, pp. 1–16, 2018.

[27] B. Penzenstadler *et al.*, “The SusA Workshop - improving sustainability

awareness to inform future business process and systems design.", *Empirical Software Engineering Community*, January 2020.

[28] R. Amri and N. Bellamine Ben Saoud, “Towards a generic sustainable software model,” *Proc. - 2014 4th International Conference Advance Computer Commun.ications ICACC 2014*, pp. 231–234, 2014.

[29] C. Igwenagu, “Fundamentals of Research Methodology and Data Collection,” *L. Lambert Acad. Publ.*, no. June, p. 4, June 2016.

[30] S. Gounder, “Chapter 3 - Research methodology and research questions,” *Research Methodology Research Methods*, no. March 2012, pp. 84–193,March, 2012.

[31] M. Kasunic, “Designing An Effective Survey.,” *Software Engineering Institute*, p. 140,September 2005.

[32] M. Muthu, K. Banuroopa, and S. Arunadevi, “Green and Sustainability in Software Development Lifecycle Process,” *Sustainable Assessing 21st century*, pp. 1–16, 2020.

[33] Adam burden Sanjay podder, “How green your software is,” *How green your Software is?, "publication Harvard Bussiness Review"*, 2020.

APPENDIX A

Survey Questionnaire

A Survey on Sustainable Software Development

Purpose: The aim of this questionnaire is to gauge the awareness and attitude of the software developer regarding sustainability in software development, catalog the challenges and solutions against those challenges developers face during sustainable software development. Sustainability is the most fast-growing and important area of software engineering. Sustainability has four dimensions which are social sustainability, environmental sustainability, economic sustainability, and technical sustainability. The results of this survey will be used to improve the software development and the developer attitude towards sustainability by thorough sustainability-related engagement and initiatives.

Important Note: *The respondent data will be confidential and will be available upon your request.*

Section 1: Respondent Background/Bio:
Name: ___________________________ **Qualification:** ___________________________ **Designation:** ________________________ **Organization:** __________________________

Section2: Social Sustainability:

Social sustainability refers to protecting common aim of the society through long lasting software to remove negative impact.

Questions	Agree	Disagree	Don’ t Know
1. Long lasting software can remove negative impact on society.	□	□	□
2. Software can change the trust between users and the business that owns the software.	□	□	□
3. Software application should be used by different backgrounds, age groups and education level people.	□	□	□
4. Software should help people take the actions to achieve the goals, projects and tasks of a group.	□	□	□

Section3: Environmental Sustainability:
Environmental sustainability refers to satisfaction of human needs through ensuring that there are no harm effects on environment during the development and deployment of software.

Questions	Agree	Disagree	Don’ t Know
1. Software should generate less waste and emission.	□	□	□

2. Software should be produced that can influence institutions to generate less waste or emission.	□	□	□
3. Software should not impact the plants or animals around it.	□	□	□
4. Software should encourage less energy consumption.	□	□	□

Section4: Economic Sustainability:
Economic sustainability refers to cost involved in development and stakeholder's long-term investment in terms of software in its value towards society.

Questions	**Agree**	**Disagree**	**Don' t Know**
1. Software should create monetary value.	□	□	□
2. Software should build a positive relationship between the business and its customers.	□	□	□
3. Developed software should positively impact the financial situation of their customers and their people/institutions.	□	□	□

Section5: Technical Sustainability:
Technical sustainability possesses the characteristics that keep the software alive for long time and flexible enough to respond the changing technological advancement.

Questions	**Agree**	**Disagree**	**Don' t Know**
1. Software should be used for long time.	□	□	□
2. Software can easily be used by the first-time users (technical/non-technical).	□	□	□

3. It should be easy to add substantial new features.	□	□	□
4. Software should be adapted to fit new usage scenarios.	□	□	□
5. Assets controlled by the software are not easy to be attacked.	□	□	□

Section6: Environmental Sustainability:
Environmental sustainability refers to satisfaction of human needs through ensuring that there are no harm effects on environment during the development and deployment of software.

Questions	**Strongly Agree**	**Agree**	**Strongly Disagree**	**Disagree**	**Don’ t Know**
3. While developing the software, do you face the challenge of lack of practices for getting the sustainable requirements. **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________ ______________________________ ______________________________	□	□	□	□	□
4. While developing the sustainable software, do you face the challenge of high energy consumption (process, resources, and the product itself). **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________	□	□	□	□	□

______________________________ ______________________________					
5. While developing the software, do you face the challenge of high carbon emission throughout the software development. **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________ ______________________________ ______________________________	□	□	□	□	□

Section7: Economic Sustainability:
Economic sustainability refers to cost involved in development and stakeholder’s long-term investment in terms of software in its value towards society.

Questionnaire	**Strongly Agree**	**Agree**	**Strongly Disagree**	**Disagree**	**Don’ t Know**
1. To have sustainable development, do you face the challenge of poor software design (architectural, logical, physical, and user interface). **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________ ______________________________ ______________________________	□	□	□	□	□
2. To have sustainable software, do you face the challenge of lack of ICTs for coordination and communication. **What strategy can be followed to overcome this challenge?**	□	□	□	□	□

______________________________ ______________________________ ______________________________ ______________________________					
3. While developing the sustainable software, do you face the challenge of high resource requirements. **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________ ______________________________ ______________________________	□	□	□	□	□

Section8: Technical Sustainability:

Technical sustainability possesses the characteristics that keep the software alive for long time and flexible enough to respond the changing technological advancement.

Questionnaire	**Strongly Agree**	**Agree**	**Strongly Disagree**	**Disagree**	**Don’ t Know**
1. To have sustainable software development, do you face the challenge of lack of coding standards. **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________ ______________________________ ______________________________	□	□	□	□	□
2. Do you face the challenge of lack of green software development knowledge for sustainable software development?	□	□	□	□	□

What strategy can be followed to overcome this challenge? ________________________________ ________________________________ ________________________________ ________________________________	
3. While developing the sustainable software, do you face the challenge of use of throw - away prototyping strategy. *Note: Throwaway prototyping refers to the creation of a model that will eventually be discarded rather than becoming part of the final delivered software*. **What strategy can be followed to overcome this challenge?** ________________________________ ________________________________ ________________________________ ________________________________	□ □ □ □ □
4. For developing the sustainable software, do you face the challenge of redundant copies of data. **What strategy can be followed to overcome this challenge?** ________________________________ ________________________________ ________________________________ ________________________________	□ □ □ □ □
5. For developing the sustainable software, do you face the challenge of poor documentation. **What strategy can be followed to overcome this challenge?** ________________________________ ________________________________ ________________________________ ________________________________	□ □ □ □ □

Section9: Social Sustainability:					
Social sustainability refers to protecting common aim of the society through long lasting software to remove negative impact.					
Questionnaire	**Strongly Agree**	**Agree**	**Strongly Disagree**	**Disagree**	**Don’ t Know**
1. For developing the sustainable software, do you face the challenge of poor understanding of the system. **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________ ______________________________ ______________________________	□	□	□	□	□
2. For developing the sustainable software, do you face the challenge of lack of social and ethical responsibility. **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________ ______________________________ ______________________________	□	□	□	□	□
3. For developing the sustainable software, do you face the challenge of lack of government support towards green software development. **What strategy can be followed to overcome this challenge?** ______________________________ ______________________________ ______________________________ ______________________________	□	□	□	□	□

Note: If you face more challenges other than listed above please do mention those challenges below:

APPENDIX B

Respondent Bio Data / Detail

ID	Qualification	Designation	Organization
R1	BE(IT)	PROJECT MANAGER	WEBHIVE TECHNOLOGIES
R2	BSCS	SOFTWARE DEVELOPER	PLANDSTUDIOS
R3	BS CYBER TECHNOLOGY	BUSSINESS DEVELOPER	PLANDSTUDIOS
R4	BSCS	FLUTTER DEVELOPER	PLANDSTUDIOS
R5	BSCS	FLUTTER DEVELOPER	PLANDSTUDIOS
R6	BSCS	SENIOR DEVELOPER	PLANDSTUDIOS
R7	BSIT	IOS DEVELOPER	SMART CODERS PVT LTD.
R8	BSCS	.NET DEVELOPER	SMART CODERS PVT LTD.
R9	BSIT	IOS DEVELOPER	SMART CODERS PVT LTD.
R10	BSCS	SOFTWARE DEVELOPER	TECHNOLOGY JUNCTION
R11	BSCS	JUNIOR SOFTWARE ENGINEER	TECHNOLOGY JUNCTION
R12	BSIT	SOFTWARE ENGINEER	TECHNOLOGY JUNCTION
R13	MSCS	SOFTWARE ENGINEER	TECHNOLOGY JUNCTION
R14	BSCS	SOFTWARE ENGINEER	TECHNOLOGY JUNCTION
R15	BSSE	NODE JS	ITZ SOLUTIONS
R16	BSSE	.NET DEVELOPER	F3TECHNOLOGY
R17	BSIT	ANDROID DEVELOPER	ITZ SOLUTIONS
R18	BSSE	.NET DEVELOPER	F3TECHNOLOGY
R19	BSCS	WEB DEVELOPER	11VALUES
R20	BSCS	WEB DEVELOPER	11VALUES
R21	BSCS	SOFTWARE ENGINEER	11VALUES
R22	BSCS	SEO EXPERT	FAZAL TECHNOLOGIES
R23	MASTER	SENIOR WEB DEVELOPER	FAZAL TECHNOLOGIES
R24	BSIT	MOST SENIOR UI/UX DESIGNER	FAZAL TECHNOLOGIES
R25	BSCS	UI/UX DESIGNER	FAZAL TECHNOLOGIES
R26	BSSE	ANDROID DEVELOPER	FREELANER
R27	BSSE	.NET DEVELOPER	PEP
R28	BSSE	SEO	TECHTINO.TK
R29	BSIT	ANDROID DEVELOPER	UNRAVELWORKS
R30	BSSE	ANDROID DEVELOPER	FREELANER
R31	BSSE	UNITY DEVELOPER	TECH360
R32	BSCS	ARCHITECT	PEP
R33	MSCS	MANAGER	TMRC
R34	BSCS	FUNTIONAL CONSULTING	TMRC
R35	BSIT	TECHNICAL CONSULTANT	TMRC
R36	BSSE	ERP	TMRC
R37	BSCS	ENTERNY	TMRC
R38	BSIT	WEB DEVELOPER	BTECH GROUP

R39	BSSE	WEB DEVELOPER	BTECH GROUP
R40	BSCS	MOBILE APP DEVELOPER	BTECH GROUP
R41	MS PROJECT MANAGEMENT	MANAGER	BTECH GROUP
R42	BSIT	ANDROID DEVELOPER	BTECH GROUP
R43	BSCS	SOFTWARE ENGINEER	SKY SOFT
R44	BSCS	CRM DEVELOPER	SKY SOFT
R45	MSSE	SOFTWARE ENGINEER	SKY SOFT
R46	BSCS	DYNAMIC 365 DEVELOPER	SKY SOFT
R47	BSCS	ASP.NET MVC DEVELOPER	SKY SOFT
R48	BSCS	MOBILE APP DEVELOPER	SIGNUP SOLUTIONS
R49	BSCS	REACT-NATIVE DEVELOPER	SIGNUP SOLUTIONS
R50	BSCS	SOFTWARE DEVELOPER	SIGNUP SOLUTIONS
R51	BSCS	WORDPESS DEVELOPER	TECHEEKS SOLUTIONS
R52	BSSE	TEAM LEAD	SIGNUP SOLUTIONS
R53	BSCS	WEB DEVELOPER	SIGNUP SOLUTIONS
R54	BSIT	PHP DEVELOPER	TECHEEKS SOLUTIONS
R55	BSCS	SR. SOFTWARE ENGINEER	WISEMAN INNOVATIONS
R56	BSSE	SR. ETL DEVELOPER	WISEMAN INNOVATIONS
R57	BSCS	SR. SOFTWARE ENGINEER	WISEMAN INNOVATIONS
R58	BSCS	SOFTWARE ENGINEER	SECURE TECH CONSULTING(PVT)
R59	BSCS	ASSISTANT MANAGER S/W DEV	SECURE TECH CONSULTING(PVT)
R60	BSCS	SOFTWARE ENGINEER	MFSYS
R61	BSCS	SOFTWARE ENGINEER	MFSYS
R62	BSCS	SOFTWARE ENGINEER	MFSYS
R63	MSSE	SOFTWARE ENGINEER	SECURE TECH CONSULTING(PVT)
R64	BSSE	PHP DEVELOPER	FREELANCER
R65	MSSE	IT ADMINISTRATOR	SELTEQ
R66	BSCS	PHP DEVELOPER	SELTEQ
R67	BCE	SERVER ADMIN	SELTEQ
R68	BSCS	PHP/WORDPRESS DEVELOPER	SELTEQ
R69	MSSE	SR. SOFTWARE ENGINEER	JIN TECHNOLOGIES
R70	BSSE	SOFTWARE ENGINEER	JIN TECHNOLOGIES
R71	BSSE	SOFTWARE ENGINEER	JIN TECHNOLOGIES
R72	BSCS	SOFTWARE DEVELOPER	NET TECH
R73	BSCS	SOFTWARE ENGINEER	NET TECH
R74	BSCS	SOFTWARE ENGINEER	NET TECH
R75	BSCS	SR. SOFTWARE ENGINEER	NET TECH
R76	BSCS	GAME DEVELOPER	SPARTANS GLOBAL
R77	BSSE	GAME DEVELOPER	SPARTANS GLOBAL
R78	BSCS	GAME DEVELOPER	SPARTANS GLOBAL
R79	BSCS	GAME DEVELOPER	SPARTANS GLOBAL
R80	BSCS	UNITY DEVELOPER	SPARTANS GLOBAL
R81	BSCS	GAME DEVELOPER	SPARTANS GLOBAL
R82	BSCS	UNITY DEVELOPER	SPARTANS GLOBAL
R83	BSCS	GAME DEVELOPER	SPARTANS GLOBAL

R84	MS	MANAGER	SPARTANS GLOBAL
R85	BSSE	GAME DEVELOPER	SPARTANS GLOBAL
R86	BSSE	FREELANCER	FIVER
R87	BSSE	BUSSINESS ANALYST	PUCIT
R88	BSSE	SOFTWARE ENGINEER	FJWU
R89	BSSE	WEB DEVELOPER	AMFAH
R90	BSSE	MANAGER	DAHAR TECH
R91	MSIT	UNITY DEVELOPER	SPARTANS GLOBAL
R92	MSSE	DEVELOPER	FREELANE
R93	BSSE	SOFTWARE DEVELOPER	PMAS ARID
R94	BSSE	INSTRUCTOR	SCHOL
R95	MASTER	DEVELOPER	FREELANCER
R96	MSSE	TEAM LEAD	RED APES
R97	BSSE	GAME DESIGNER	SPARTANS GLOBAL
R98	BSCS	GAME DEVELOPER	SPARTANS GLOBAL
R99	BSCS	UNITY DEVELOPER	SPARTANS GLOBAL
R100	MCS	REAL NATIVE ENGINEER	BITSOL TECH
R101	BSCS	SOFTWARE ENGINEER	PRAL
R102	BSSE	SENIOR DEVELOPER	MOD
R103	MCS	SOFTWARE ENGINEER	PRAL
R104	BSSE	IOS DEVELOPER	PRAL
R105	MS	WEB DEVELOPER	BITSOL TECH
R106	MCS	LEAD SOFTWARE ENGINEER	BITSOL TECH
R107	MS	WEB DEVELOPER	BITSOL TECH
R108	BSCS	WEB DEVELOPER	BITSOL TECH
R109	BSSE	DB DEVELOPER	PRAL

APPENDIX C

Respondent Responses

Responses against Awareness Level

ID	SQ1.A	SQ2.A	SQ3.A	$Q4.A	EnQ1.A	EnQ2.A	EnQ3.A	EnQ4.A	EcQ1.A	EcQ2.A	EcQ3.A	TQ1.A	TQ2.A	TQ3.A	TQ4.A	TQ5.A
R1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R3	2	1	2	1	1	1	1	2	1	1	1	2	2	2	1	2
R4	0	1	2	1	1	1	1	1	0	1	1	1	1	1	2	1
R5	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1
R6	2	2	1	1	1	1	1	2	1	1	1	2	1	2	1	2
R7	2	1	1	1	1	1	2	1	1	1	1	0	2	1	1	1
R8	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1
R9	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
R10	1	1	1	1	2	2	2	1	1	1	1	2	1	2	1	2
R11	2	2	1	1	0	1	2	1	1	1	1	1	1	1	1	1
R12	2	1	1	1	1	1	1	1	0	1	1	1	1	0	1	0
R13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
R14	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
R15	0	1	1	1	1	1	2	2	0	1	1	1	1	1	2	1
R16	0	1	1	1	1	1	2	2	0	1	1	1	1	1	2	1
R17	1	1	1	1	0	2	1	1	2	2	1	2	1	1	2	2
R18	1	1	1	1	0	2	1	1	1	1	1	2	1	1	2	1
R19	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R20	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	2
R21	0	1	2	1	1	0	1	1	1	1	1	1	1	1	1	1
R22	2	1	1	1	0	0	0	1	1	1	1	1	1	0	2	1

R 23	2	1	1	1	0	0	0	1	1	1	1	1	1	0	2	1
R 24	2	1	1	1	1	2	1	1	1	1	1	1	1	1	2	0
R 25	2	1	1	1	1	2	1	1	1	1	1	1	1	1	2	0
R 26	2	1	1	1	1	1	2	1	1	1	1	0	1	1	1	1
R 27	0	1	2	1	2	1	0	1	1	2	0	1	1	1	2	1
R 28	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1
R 29	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	2
R 30	2	1	1	2	1	2	1	0	1	1	1	1	1	1	1	1
R 31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 32	2	1	1	2	1	2	1	0	1	1	1	1	1	0	1	2
R 33	2	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2
R 34	2	2	0	1	1	1	1	1	1	1	0	2	2	1	1	2
R 35	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
R 36	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
R 37	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 38	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1
R 39	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 41	2	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1
R 42	2	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1
R 43	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 44	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 45	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 46	0	2	1	1	1	1	1	1	1	0	1	1	1	1	1	1
R 47	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 48	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 49	0	1	2	1	1	1	0	1	0	1	1	1	1	1	1	1
R 50	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1
R 51	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
R 52	2	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1

R 53	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1
R 54	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	2
R 55	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
R 56	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 57	2	1	1	1	1	1	1	0	1	1	1	2	2	1	1	2
R 58	1	1	1	1	1	0	2	2	0	1	1	1	1	1	1	1
R 59	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 60	1	1	1	1	1	1	1	1	0	1	1	2	1	1	1	2
R 61	0	1	1	1	1	1	1	1	1	2	1	1	1	2	0	1
R 62	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2
R 63	1	2	1	1	1	1	1	2	1	1	1	1	2	2	1	2
R 64	2	2	1	1	2	2	1	1	0	1	2	2	2	1	2	1
R 65	1	1	1	1	0	1	2	1	2	2	1	1	1	1	1	1
R 66	1	1	1	1	0	1	2	1	1	2	1	1	1	1	1	1
R 67	2	2	2	1	2	1	2	2	1	2	1	2	2	1	0	2
R 68	2	2	2	1	2	1	2	2	1	2	1	2	2	1	0	1
R 69	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 70	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
R 71	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
R 72	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
R 73	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R 74	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R 75	1	2	1	1	1	1	1	1	1	1	1	1	1	1	0	1
R 76	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1	2
R 77	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
R 78	2	1	1	1	1	1	1	1	2	1	1	2	1	1	1	1
R 79	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R 80	0	1	1	1	1	1	0	1	1	1	1	1	2	1	1	1
R 81	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R 82	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1

R 83	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R 84	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R 85	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 86	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 87	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
R 88	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 89	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 90	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 91	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 92	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 93	1	1	1	1	1	1	1	1	0	1	1	1	2	1	1	0
R 94	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	2
R 95	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R 96	2	1	1	1	1	1	2	1	1	1	1	2	0	2	1	2
R 97	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R 98	2	1	1	1	1	1	2	1	1	1	1	1	1	1	2	1
R 99	2	1	1	1	1	1	2	1	1	1	1	1	1	1	0	1
R 100	2	1	1	1	1	1	0	1	1	1	1	1	0	1	2	1
R 101	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R 102	1	1	1	1	1	1	2	1	0	1	1	1	1	1	2	1
R 103	2	1	1	1	1	1	2	1	1	1	0	1	1	1	1	1
R 104	2	1	1	1	1	1	2	1	1	1	1	0	2	1	1	1
R 105	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
R 106	0	1	1	1	3	1	1	1	1	1	1	1	1	1	1	1
R 107	1	2	1	1	2	1	0	1	1	1	1	1	2	1	1	2

R108	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
R109	2	1	1	1	1	1	2	1	1	1	1	1	1	1	0	1

Reponses against Challenges

ID	EnQ 1.C	EnQ 2.C	EnQ 3.C	EcQ 1.C	EcQ 2.C	EcQ 3.C	TQ 1.C	TQ 2.C	TQ 3.C	TQ 4.C	TQ 5.C	SQ 1.C	SQ 2.C	SQ 3.C
R1	2	2	0	2	2	1	2	0	2	2	2	2	3	4
R2	2	1	1	1	1	1	2	2	1	1	1	1	2	1
R3	4	2	4	4	3	2	3	2	4	4	2	4	4	2
R4	2	3	1	4	0	4	4	4	4	4	4	4	4	4
R5	2	2	4	4	2	2	4	4	4	2	2	2	2	2
R6	4	1	4	1	3	2	3	2	4	4	2	4	4	2
R7	2	1	0	1	4	4	3	1	0	2	1	2	3	2
R8	2	0	0	2	0	4	4	0	2	4	4	2	0	1
R9	2	2	1	2	2	2	1	2	3	3	2	1	3	4
R10	1	2	1	4	1	2	4	4	0	1	1	1	1	1
R11	0	4	0	4	4	4	4	0	0	4	3	3	3	0
R12	0	2	0	2	2	0	2	0	0	2	2	2	2	0
R13	4	2	4	4	3	2	4	4	2	2	4	3	3	0
R14	2	4	1	2	4	2	4	4	4	3	2	4	3	2
R15	1	2	4	1	0	2	2	0	0	2	2	2	0	1
R16	1	2	4	1	0	2	1	2	0	2	2	1	4	1
R17	1	1	0	2	1	2	2	0	3	1	1	2	1	1
R18	1	1	0	2	1	2	2	4	4	1	1	1	2	0
R19	1	2	0	1	2	1	1	2	2	2	2	1	0	4
R20	2	2	4	1	2	1	1	1	1	2	1	1	4	3
R21	2	2	4	2	2	1	1	2	1	1	1	1	2	4
R22	2	1	4	0	0	2	1	2	1	1	1	2	1	4
R23	2	1	4	0	0	1	2	4	1	1	1	2	2	3
R24	1	1	2	3	1	3	1	2	0	2	4	1	1	3
R25	1	2	4	3	1	3	1	0	0	2	0	1	1	3
R26	1	2	4	1	4	4	2	1	0	2	1	1	4	2

R27	1	1	1	4	4	2	2	0	2	2	2	0	3	3
R28	0	2	2	2	2	2	2	2	2	2	0	0	0	0
R29	1	4	4	4	1	1	1	1	0	4	2	0	1	2
R30	1	2	1	2	1	2	1	4	2	1	1	1	1	1
R31	2	1	0	2	2	1	2	1	1	1	1	1	0	3
R32	0	1	3	4	2	4	2	1	1	1	1	1	3	2
R33	2	2	3	2	3	3	2	3	2	3	3	3	3	3
R34	2	2	3	2	0	2	4	0	1	0	2	2	4	1
R35	2	2	3	2	2	1	2	3	3	3	1	2	3	3
R36	2	2	4	2	2	2	2	2	2	1	2	2	1	2
R37	2	2	0	2	0	2	4	4	4	4	4	4	4	4
R38	1	1	0	2	0	2	1	0	0	2	2	2	0	4
R39	2	4	4	2	4	4	1	0	4	4	1	4	2	4
R40	1	1	4	1	0	1	2	1	4	2	2	2	4	2
R41	2	1	0	4	4	2	4	0	0	4	2	2	4	0
R42	2	1	0	2	0	2	4	0	2	3	2	2	4	0
R43	2	1	2	2	2	1	1	2	1	2	2	1	0	3
R44	4	2	0	1	1	1	4	2	1	1	2	1	1	4
R45	3	1	3	1	1	1	1	1	1	1	1	1	1	3
R46	2	1	4	1	2	1	2	1	2	1	1	1	2	3
R47	3	1	3	1	1	1	1	1	1	1	1	1	1	3
R48	2	4	4	2	4	4	1	0	4	4	1	4	2	4
R49	1	1	4	1	2	1	1	1	1	1	1	2	3	3
R50	1	1	0	2	0	2	2	0	0	2	1	1	1	0
R51	2	2	4	1	0	1	2	1	4	2	2	2	4	2
R52	2	2	0	1	1	2	2	0	0	1	2	2	1	4
R53	2	4	4	2	4	2	2	0	0	4	1	2	4	0
R54	1	2	4	0	4	4	4	4	0	2	2	2	2	4

R55	2	2	0	2	2	2	2	2	2	2	1	4	4	2
R56	4	2	4	2	2	2	4	2	3	2	2	4	2	2
R57	2	1	4	2	4	4	4	0	4	4	2	4	2	1
R58	2	2	0	4	3	0	4	0	4	0	4	4	4	2
R59	2	2	4	3	3	2	4	4	4	4	4	4	4	4
R60	1	2	1	2	1	1	2	1	1	2	1	1	2	2
R61	1	2	1	4	3	2	2	2	1	1	3	1	3	2
R62	1	2	1	2	1	1	2	1	1	2	1	4	2	3
R63	4	2	4	4	4	2	2	2	4	4	4	4	4	4
R64	2	4	3	2	0	3	2	2	2	2	1	2	1	2
R65	1	1	1	2	2	1	1	2	2	2	2	2	2	1
R66	1	1	1	2	2	1	2	1	2	2	2	1	1	2
R67	3	2	2	2	2	0	0	0	2	3	4	2	3	3
R68	3	2	2	2	2	1	2	0	1	3	4	2	3	3
R69	2	0	2	1	2	1	1	4	2	2	1	2	4	3
R70	2	4	3	1	2	1	1	4	2	2	1	1	2	3
R71	1	3	3	1	2	1	1	4	1	2	1	1	2	3
R72	1	2	2	1	1	1	1	1	1	1	1	1	2	4
R73	2	2	4	1	2	2	1	4	1	1	1	2	2	3
R74	2	1	4	2	2	1	1	3	1	2	1	1	2	3
R75	1	1	2	2	2	1	1	2	2	1	2	1	4	3
R76	2	2	2	2	3	1	1	2	4	2	2	1	1	1
R77	2	4	4	2	2	2	2	2	4	3	3	2	2	2
R78	1	2	0	1	2	2	1	1	2	1	1	2	2	1
R79	1	2	0	2	2	2	2	2	2	2	1	2	4	4
R80	1	1	0	2	2	1	2	2	4	2	1	1	2	4
R81	1	2	0	2	2	2	2	2	4	2	2	2	3	3
R82	1	1	1	1	2	2	2	1	2	2	2	2	0	4

R83	2	2	4	2	2	2	2	2	4	2	2	1	4	4
R84	1	1	1	1	2	2	2	1	2	2	2	1	3	2
R85	2	2	2	2	2	2	2	0	2	2	2	2	2	2
R86	1	2	2	1	1	2	1	0	2	2	2	2	2	2
R87	1	2	0	2	1	2	1	0	0	0	0	2	2	2
R88	2	2	2	2	2	2	2	1	2	2	2	2	2	2
R89	2	2	2	2	2	2	2	2	2	2	2	2	2	2
R90	4	2	0	2	4	4	4	0	4	2	3	4	4	1
R91	4	4	0	2	4	2	2	4	0	2	2	4	2	4
R92	2	2	0	2	2	2	2	2	2	2	2	2	2	2
R93	2	1	2	1	1	1	1	2	4	4	2	1	1	3
R94	2	2	4	4	2	1	2	1	2	4	2	2	2	1
R95	1	1	1	1	2	2	4	2	2	2	2	4	2	2
R96	2	2	4	2	3	2	2	0	0	0	0	2	0	0
R97	2	2	4	2	2	2	2	2	2	2	2	2	2	2
R98	2	2	4	2	2	2	2	2	0	2	1	2	0	2
R99	1	2	4	2	1	2	2	2	0	2	1	2	2	4
R100	1	2	3	2	2	1	1	2	0	1	1	1	2	4
R101	2	2	4	2	2	1	2	2	2	1	1	2	2	2
R102	1	2	4	1	2	1	1	2	1	1	1	1	2	4
R103	1	1	1	2	2	1	1	1	4	1	1	1	2	4
R104	1	1	3	2	2	2	2	2	0	1	2	1	2	3
R105	1	1	3	2	2	1	1	1	4	1	1	4	4	2
R106	2	4	4	4	4	4	4	4	4	4	4	0	4	4
R107	2	2	0	0	2	2	2	1	4	2	2	2	2	3
R108	2	2	2	2	2	1	1	1	2	1	1	1	2	3
R109	1	1	3	1	2	1	2	2	0	1	1	2	0	4