Lappeenranta University of Technology
School of Business and Management (LUT)
Erasmus Mundus Master's Program in Pervasive Computing & Communications for sustainable Development PERCCOM

Oyedeji Abdullateef Shola

EARLY INVESTIGATION TOWARDS DEFINING AND MEASURING SUSTAINABILITY AS A QUALITY ATTRIBUTE IN SOFTWARE SYSTEMS

2016

Supervisor(s) : Professor Ahmed Seffah (Lappeenranta University of Technology)
Professor Jari Porras (Lappeenranta University of Technology)

Examiners: *Professor Eric Rondeau* (University of Lorraine)
Professor Jari Porras (Lappeenranta University of Technology)
Professor Karl Anderson (Luleå University of Technology)

This thesis is prepared as part of an European Erasmus Mundus programme PERCCOM - Pervasive Computing & Communications for sustainable development.

Co-funded by the Erasmus+ Programme of the European Union

This thesis has been accepted by partner institutions of the consortium (cf. UDL-DAJ, n°1524, 2012 PERCCOM agreement).

Successful defense of this thesis is obligatory for graduation with the following national diplomas:

- Master in Master in Complex Systems Engineering (University of Lorraine)
- Master of Science in Technology (Lappeenranta University of Technology
- Master in Pervasive Computing and Computers for sustainable development (Luleå University of Technology)

ABSTRACT

Oyedeji Abdullateef Shola

Early Investigation Towards Defining and Measuring Sustainability as a Quality Attribute in Software Systems

Lappeenranta University of Technology
School of Business and Management (LUT)
Computer Science (LUT)
Erasmus Mundus Master's Programme in Pervasive Computing & Communications for Sustainable Development PERCCOM

Master's Thesis, 2016, 81 pages, 18 figures, 20 tables

Examiners: Professor Eric Rondeau (University of Lorraine), Professor Jari Porras (LUT), Professor Karl Anderson (Luleå University of Technology)

Keywords: Sustainability, Software Sustainability, Software Measurement, Software Quality Attribute, Software Development, Sustainability Measurement.

Context: Sustainability in software system is still a new practice that most software developers and companies are trying to incorporate into their software development lifecycle and has been largely discussed in academia. Sustainability is a complex concept viewed from economic, environment and social dimensions with several definitions proposed making sometimes the concept of sustainability very fuzzy and difficult to apply and assess in software systems. This has hindered the adoption of sustainability in the software industry. A little research explores sustainability as a quality property of software products and services to answer questions such as; How to quantify sustainability as a quality construct in the same way as other quality attributes such as security, usability and reliability? How can it be applied to software systems? What are the measures and measurement scale of sustainability? The **Goal** of this research is to investigate the definitions, perceptions and measurement of sustainability from the quality perspective. Grounded in the general theory of software measurement, the aim is to develop a method that decomposes sustainability in factors, criteria and metrics. The **Result** is a method to quantify and access sustainability of software systems while incorporating management and users concern. **Conclusion**: The method will empower the ability of companies to easily adopt sustainability while facilitating its integration to the software development process and tools. It will also help companies to measure sustainability of their software products from economic, environmental, social, individual and technological dimension.

ACKNOWLEDGEMENTS

I want to say big thanks to my supervisors Prof. Ahmed Seffah and Prof. Jari Porras for providing me with good insight on my thesis work, both of you provide a superb balance with your research approaches.

I would also like to thank my parents and family members for their understanding through out the period of my thesis and Diajeng Rahmawati for your patience during all those difficult moments.

The opportunity to carry out this research only came as a result of me been selected in this PERCCOM program, I want to say thank you to Prof. Eric Rondeau, the PERCCOM consortium and EU for choosing me.

Carrying out this research work was not easy, it was made possible with help of my great friends when needed, I will like to thank Dr. Birgit Penzenstadler, all my lovely and awesome friends in Cohort 2, LUT Friends and my cool friend Victoria Palacin. My Grezzli family, thanks for the understanding throughout my thesis period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ICT	Information and Communication Technology
NFR	Non Functional Requirements
SNFR	Sustainability Non Functional Requirement
IT	Information Technology
GHG	Greenhouse Gas
SO	Sustainable Development
CEO	Chief Executive Officer
SC	Software Companies
IEEE	Institute of Electrical and Electronics Engineers
NSC	Non Software Companies
GQM	Goal – Question – Metric
SGQM	Sustainable Goal Question Metric
BMI	Backlog Management Index

1 INTRODUCTION

The concept of sustainability in software systems is still a new practice that most software developers and companies are trying to incorporate into their software system. There is not enough consideration from the research community in investigating sustainability as a quality attribute in software system.

Software Quality is the degree to which software possesses a desired combination of attributes. It can also be defined and viewed from two perspectives; the first been conformance to specification which means software quality whose measurable characteristics satisfy a fixed requirement based on beforehand specification and second meeting customer needs which means software quality capability to meet needs and expectations of customers whether its explicit or not [1]. Exploring software sustainability from software quality viewpoint will have big positive impact on software system design and development because software quality attributes are part of the system nonfunctional requirement covering the overall factors that affect run-time behavior of system design and user experience that represent areas of concern which have the potential for application wide impact across layers [2]. Several factors are important to quantify sustainability as a quality attribute in software system. Some crucial points are the understanding, motivation, and commitment from management and engaged personnel with consideration to the economy, society and environment. Currently, sustainability is not supported by traditional software development process which leads to inefficient efforts to address sustainability or total omission during software development[3] that leads to unsustainable software systems.

1.1 Background

Contrary to the notion that software is environmentally friendly since its virtual, the processes and methods used to develop, maintain and deploy software do have an environmental and social impact that is normally not accounted for by the current software development practices [4]. For example, the lifetime and ability to sustain of software systems would increase if sustainability were taken into consideration as quality attribute during development. Dick et. al [5] stated that software development lifecycle model from the business sense is not appropriate for identifying effects of software system on sustainability. The main focus of the current model is on the business, development and maintenance phase without sustainability consideration.

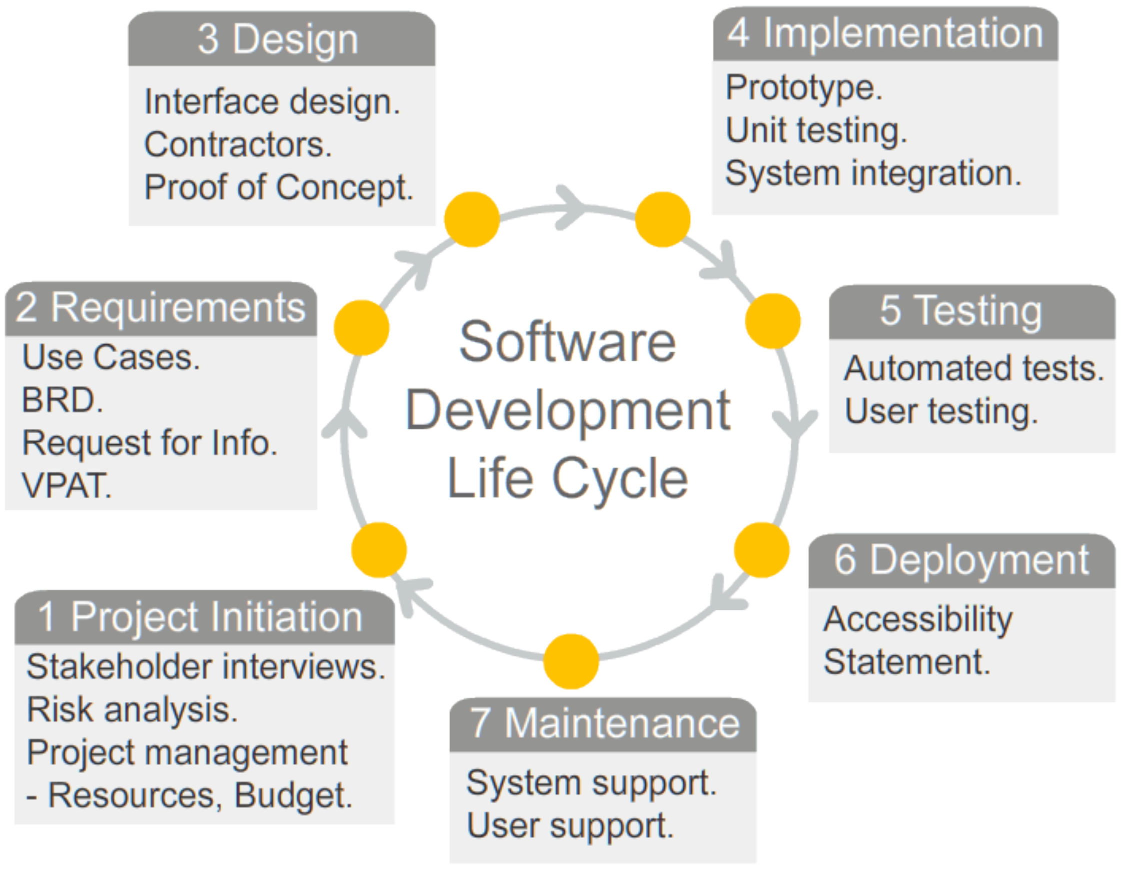


Figure 1. Software Development Life Cycle adopted from [6]

Figure 1 above shows that present software development life cycle doesn’t cover sustainability and this has hinder the ability of companies and developers to incorporate it into the development process in order to create sustainable systems and measure its impacts on the environment. These impacts can be categorized into first, second and third order impacts.

- First order impacts are direct environmental effects created by the physical existence of information and communication technology (ICT) and the processes involved. This includes the design, manufacture, operation and waste disposal of ICT [7] [8].
- Second order impacts are indirect environmental impacts created by the ongoing use, distribution and application of ICT [7] [8].
- Third order impacts though very hard to assess are rebound effects created by the accumulated effects of large numbers of people using ICT over a long period of

time (first and second order impacts) [7] [8].

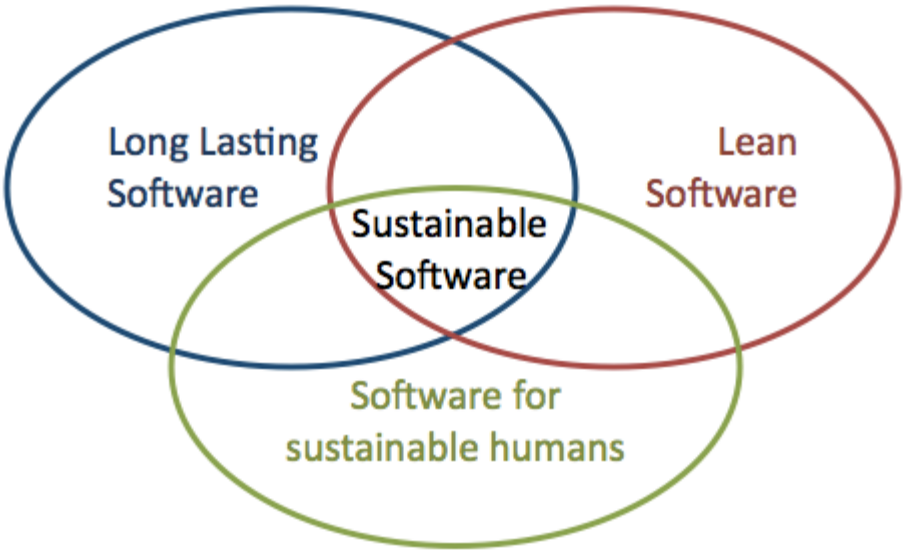


Figure 2. Sustainable Software adopted from [9]

Sustainability is defined as “the capacity to endure”[3] and sustainable software according to the diagram above can be viewed from three angles; Long lasting software which relates to how well a piece of software will be able to cope with changes. Lean software as software, which require less hardware and reduces its own power consumption (energy efficient). Software for sustainable humans as software which induces sustainable human behavior. This definition does not provide enough direction for the different dimensions of sustainability (environment, economy and social) that can be related to software systems. It is therefore very important to clearly define and specify what does sustainability mean in/for software systems with regards to quality attribute. To achieve sustainability as a quality attribute, the development process has to be itself sustainable because it can lead to development of sustainable software. Sustainable software development brings together principles and practices for building software that is technically superior, delivers exceptional business value, and can evolve rapidly to reflect any change to your business or technical environment[10].

Software systems have a major part in ICT, which can help to reduce the negative impacts of ICT on the environment such as the first, second and third, order impacts. The two major areas of information and communication technology (ICT) connected to sustainability are Green by IT (how to encourage sustainability by ICT) and Green IT (How to make ICT more sustainable), what connects these two areas is software: Green by IT that is software base involves tools that help to enhance logistics and to make processes automated in order to save energy; Green IT ensure that software systems uses less resources (energy efficient), this shows importance of software when it comes to sustainability and the huge impact its has in the world.

Dealing with sustainability requirements and systematically supporting their elicitation, analysis, and realization is a problem that has yet to be solved. Decades ago, the discipline of software engineering dealt with similar shortcomings in its processes by including safety and security as new system qualities[11]. Due to the increasing consequential effect of not tackling the issues of sustainability in software system development, developers should use the lesson learned from previous research works as a guide. Considering sustainability as a quality attribute in/for software systems is not only about how energy efficient is the system, which is concerned with the first-order impacts of software systems. Software developers must also take into account sustainability as a nonfunctional requirement (quality attribute) like security, usability, portability that help to cover the second and third-order impacts in the system context, even if they are hard to assess. Through that, developers have the potential to significantly improve software sustainability from economic, social and environmental perspective.

There has been a lot of different Green Software Models to improves sustainability of software systems especially the mostly referenced GREENSOFT Model by Nauman et al. [12] which has the objective to support software developers, administrators and users in creating, maintaining and using software in a more sustainable way but none has proffer the solution of adding sustainability as a quality attribute in software systems. It is therefore important to research how to quantify sustainability as quality attribute and to evaluate the effectiveness of sustainability as a quality attribute in software systems requires measures through metrics that provide concrete results of how well the development process produce sustainable software.

1.2 Goals and Delimitations

Development of **sustainable software** is a major issue in software engineering, what sustainability means in software systems is still not clear because there **is no agreed definition for sustainability** in the field of software engineering and how to integrate it into software system. The lack of clarity has resulted in poor adoption of sustainability during software system development.

The aim of this research is to explore the **emerging definition of sustainability in software system** from the **quality perspective** and provide a method to encourage the adoption sustainability in software system development with a means to use metrics for assessing the software systems.

Main Research Question: How to quantify sustainability as a quality attribute in software systems.

- How to define sustainable software under the context of Main research question.
- What are the measures of sustainability (economic, social, environment)
- How to apply these measures in software system.

In order to answer these research questions, the thesis research will be grounded under **design science research methodology** with **qualitative research method** to have better understanding of all current practices and research relating to sustainability as quality attribute in software systems and proffer solution to support acceptance of sustainability in software system as a quality attribute during system development by companies and ICT experts.

1.3 Structure of the Thesis

This research work is divided into 7 chapters; the following are the breakdown of each chapter.

Chapter 1 titled: Introduction that provides information about the research context covering sustainability in software systems, the most citied definition of sustainability, the goals and delimitation of this research work.

Chapter 2 titled: Start of the Art explores different research work relating to sustainability in software systems, software quality attribute (nonfunctional attribute), software measurements, sustainability in software development to find research gaps.

Chapter 3 titled: Methodology presents procedure of carrying out this research using design science and qualitative research method with the application of grounded theory approach for data collection and analysis and justification for choosing grounded theory. This chapter presents the research organization and categorization of participants in the interview and the interview themes.

Chapter 4 titled: Results shows the result from the interview. The result from this chapter provides information as to why companies and ICT specialist are having hard time adopting sustainability into software systems.

Chapter 5 titled: Sustainability As Quality Attribute explores previous research works in order to proffer holistic way of defining sustainability as a quality attribute in software systems and presents the impact it can have on other existing software quality attributes.

Chapter 6 titled: Grounding Sustainability in the Field of Software Measurement recommends a way to put sustainability into software measurement through the proposed SGQM model that can help encourage companies to have sustainability thinking in software product development process and serve as a measurement platform for software systems.

Chapter 7 is titled: Summary and Future Work summarize the research work and the future research activities.

The thesis ends with References list for all cited work in the thesis work.

2 STATE OF THE ART

There have been a lot of works from different research communities to address software quality attributes (non functional requirements) such as security, usability, dependability, performance and other similar requirements as well. There is very few research related to sustainability as quality attribute in software system. Most research works focus on energy reduction in areas of software for sustainability, Green software and sustainability in software systems. The main objective of this section is to gather information about sustainability, software quality attributes, sustainability in software development, Metrics and measurements of sustainability in software systems, software development life cycle. A quick overview of different definition of sustainability will provide a starting point for this literature review.

2.1 Sustainability-Related Definitions And Descriptions

Table 1. Sustainability Related Descriptions

Source	**Definition**
SustainAbility company view [13]	Sustainability is the capacity to endure
UN WC on Environment & Development [14]	Sustainable development as a development that 'meets the needs of the present generation without compromising the ability of future generations to meet their own needs'
Heinberg R. [15]	Substances introduced into the environment from human activities must be minimized and rendered harmless to biosphere functions. Where pollution from extraction and consumption of nonrenewable resources has proceeded at expanding rates for some time and threatens the viability of ecosystems, reduction in the rates of extraction and consumption of those resources may need to occur at a rate greater than the rate of depletion

Tainter J. [16]	1. Sustainability is an active condition of problem solving, not a passive consequence of consuming less resources. 2. A society or other institution can be destroyed by the cost of sustaining itself. To define sustainability in a specific context, the questions should be to **Sustain what?** , **For whom?** , **How long?** And **At what cost?**
Polese M. and Stren R. [14]	Social sustainability as 'policies and institutions that have the overall effect of integrating diverse groups and cultural practices in a just and equitable fashion.'
Harris J. M. and Goodwin N. R. [14]	A socially sustainable system must achieve fairness in distribution and opportunity, adequate provision of social services, including health and education, gender equity, and political accountability and participation.
Hilty L. M. et al. [14]	In order to evaluate sustainability of ICT systems, the first order , second order and third order effect must be considered.
Seacord et. al., [17]	Software sustainability as the 'ability to modify a software system based on customer needs and deploy these modifications'.

The above descriptions and definitions of sustainability shows that there are many angles to sustainability which makes it very complex to define especially when applied in software systems. Naumann et al. [12] defined sustainable software as software whose direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software are minimal and/or which has a positive effect on sustainable development.

This definition provides basis for starting but still lacks some clarity as to what is considered minimal, how to define sustainable development in regards to software systems, when sustainability is consider in the context of software systems according to Joseph Tainter[16] its is important clearly identify the following; ***Sustain what?, For whom?, How long?*** and ***At what cost?.*** The definition of sustainability relating to software systems must have quantifiable variables in order to be able to access and evaluate how sustainable are software systems.

2.2 Software Quality Attribute and Sustainability as Non-Functional Requirement

Gorton I in his book[18] stated that software quality attributes are the overall factors that affect run-time behavior of system design and user experience which represent areas of concern that have the potential for application wide impact across layers and tiers. For a system development project to be successful its not enough to merely satisfy functional requirements but also satisfy nonfunctional requirements. Quality attribute requirements are part of an application's nonfunctional requirements, which capture the many facets of how the functional requirements of an application are achieved.

Software quality is the degree to which software possesses a desired combination of attributes (example: reliability, interoperability)[19]. Some of these attributes are related to the overall system design, while others are specific to run time, design time, or user centric issues. The extent to which the application possesses a desired combination of quality attributes such as usability, performance, reliability, and security indicates the success of the design and the overall quality of the software application.

Penzenstadler et al. [11] explained that the ubiquity of software systems led to both software security and software safety failures, which necessitated software engineering researchers and practitioners to address security and safety during software development. Security and safety were added to the set of nonfunctional requirements, whereas taxonomies of nonfunctional requirements at that time focused on other qualities such as efficiency, reliability, and usability. The present challenges of sustainability in software systems require sustainability to be added as a quality attribute (nonfunctional requirement) like security and safety. The paper argues that sustainability must be treated as a first- class quality alongside other critical attributes such as safety, security, efficiency, reliability, and usability. Sustainably engineered software systems could help facilitate a more sustainable lifestyle by directly influencing the surrounding system context (including the people, infrastructure, and environment). Software systems have such a significant impact on everyday lives that changes towards more environmental sustainability can ripple to other systems with which they interact and also affect other industries. According to the authors the impacts of software system on sustainability are divided into three orders of magnitude; first, second and third order Impacts.

Observation

The paper has a good proposition to add sustainability as nonfunctional requirement like security and safety but focused mainly on environmental sustainability side in regards to software systems because the authors belief other aspect of sustainability are, in part or implicitly, already supported by established software engineering practices and also touches on sustainable software development but didn’t explicitly explain how to go about incorporating sustainability into software development processes.

Raturi et al. [20] paper focused on how to develop sustainability as a nonfunctional requirement (NFR) using NFR framework informed by sustainability models and how it can be used to correctly obtain and describe sustainability related requirements of the software system to be developed. In order to develop sustainability requirements, it is important to first know what aspect of sustainability can the software support. The paper also touches on the first order effect, second order effect and third order of software systems and the importance for sustainability requirement to consider how far an impact on each requirement and software can and should reach. The authors introduce five dimension of sustainability:

1. **Human Sustainability**: Maintenance of human capital, health, education.
2. **Social Sustainability**: Maintenance of social capital. Focus on community continuance.
3. **Economic Sustainability**: Maintenance of economic capital.
4. **Environmental Sustainability**: Maintenance of environmental capital, including preservation of resource and protection of the environment.
5. **Technical sustainability**: Long-time usage of systems and their adequate evolution with changing surrounding conditions and respective requirements.

The five dimensions (Human, Social, Economic, Environment and Technical) according to the authors appear in the form of things like cost of energy efficient algorithms (environmental), people hours involved in development (human), development team management (social), development (economic) and code maintainability (technical), which are treated separately during software development. In order to create sustainable software it is important that these dimensions are treated with respect to what software is built (requirements) not just how it is built.

The paper proposed a method for reasoning about different decisions while putting together the requirements of a software system called Sustainability nonfunctional requirement (SNFR) framework.

Observation:

The Paper proposes Sustainability Nonfunctional Requirements framework but does not provide detailed specifics on how to evaluate the proposed Sustainability Nonfunctional Requirements framework and also how to integrate the framework into different IEEE standards (ISO/IEC/IEE Standard 29148-2011 addresses requirements engineering processes).

2.3 Sustainability in Software Development and Metrics

Software development project usually treat sustainability as an afterthought as developer are pressured to deliver software in time and mostly are not educated about sustainability according to Durdik et al. [21]. The authors argue that lack of holistic overview of sustainability and several individual techniques and approaches in software engineering is often not sufficiently validated as a result the authors created a catalog of "software sustainability guidelines" to support project managers, software architects, and developers during system design, development, operation, and maintenance

The sustainability guideline was applied to two case studies to improve the systems under study. According to the authors, the case studies were not intended to formally validate the applicability of the guidelines, which would require a longitudinal study to be able to quantify their effect on a software system's evolution. Rather the case studies gave hints on the applicability of selected sustainability- improving methods and helped to mature and refine the guidelines themselves.

Observation

The authors stated that their proposed guideline provide unique perspective as it incorporates all phase of system life cycle but during the application of the guideline for the two case studies there was no sufficient information about how this guideline can be incorporated to system life cycle.

Erdélyi [22] paper studies the lifecycle activities of software development in the point of view as they contribute to environmental protection. Hardware behavior of an IT system is determined by the software installed on it, which can either lead to more or less resources consumption. Software applications designed environmentally consciously consumes 40% less energy than other applications with the same functionality. The author provided definitions and aims for the following terms:

1. **Green IT:** Information Technology (IT) having positive impact on the environments to produce as little waste as possible during the whole IT lifecycle (development operation and disposal).
2. **Green by IT:** IT providing tools for doing task in environmental friendly manner (Using video call instead of travelling), which aims at producing as little waste as possible by means of IT.
3. **Green software:** Software applications that run on environmental friendly manner which aim to produces as little waste as possible during its development and operation.
4. **Green by software**: software applications that help eco sustainability. It indicates methods, tasks and applications that help reach the aim of green by IT.

The paper highlights three ways software engineering can be green such as to produce green software, to produce software to support environmentally consciousness (green by software) and produce less waste during development through three key points.

Table 2. Activities and Task in Green Software System Development adopted from [22]

Activities during software development	**Key points and tasks**
Analysis and Design	1. Computational efficiency, 2. Data efficiency, 3. Context awareness 4. Idle efficiency
Implementation	5. Languages and IDEs have effect to energy consumption of applications.
System requirements	6. Non-functional requirements

	contain demands related to sustainability in ecology
Test	7. Lots of metrics are defined the 'green' level of an application. Most of them have a connection to energy consumption.

Observation

The paper provides an overview of different activities and advice on what to do in order to develop green software but it would have been more interesting if the authors provided different factors and metrics relating to green software during the development process. The formula provided by the authors does not give insight on how the parameters used can be derived during software development.

Kevin Tate [10] explained the meaning of **sustainable software development as mindset (principles) and an accompanying set of practices that enable a team to achieve and maintain an optimal development pace indefinitely.** A very important point highlighted is that unsustainable software development leads to software that is both brittle and fragile as a result of factors such as over- (or under-) design, a code first then fix defects later (code-then-fix) mentality, too many dependencies between code modules, the lack of safeguards such as automated tests, and supposedly temporary patches or workarounds that are never addressed.

According to the author working longer hours has negative impact on the long term efficiency of the software team (declining capability) but rather working smart and being proactive to towards continual improvement in the software ecosystem result in doing work that is of the highest value to customers with high quality and reliability despite the increasing complexity. This also increases ability to respond to changes and keeps cost of change as low as possible.

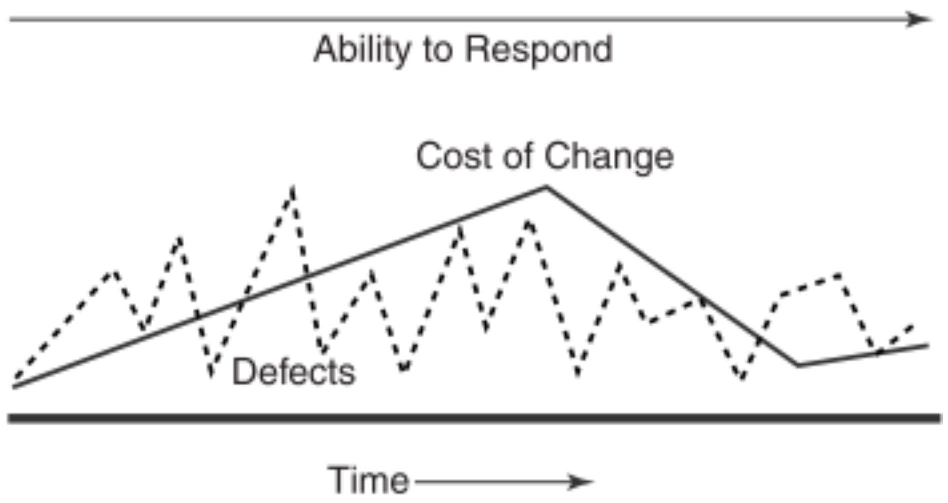


Figure 3. Sustainable Development adopted from [10]

The book chapter presented a case study of Chemical Manufacturing and Sustainable Development where the company is at the verge of closing down; the important lesson from this case studies is that software companies need to strive for continual improvement while resisting the temptation to focus on features and simply working long hours to meet demands. Software development is complex in an environment that is constantly changing with uncertainties, in order for software companies to alleviate this issue, it is important to imbibe the culture of sustainable software development, which is the ability to maintain an optimal pace of development indefinitely.

Observations

The book chapter presented interesting issues with sustainable software development though I disagree that optimal pace of development will lead to sustainable software development but rather incorporating sustainability values in the software development processes with optimal pace can lead to sustainable software development.

Johann et al. [23] proposed a life cycle model that helps to develop green and sustainable software products. The paper mentioned some key challenges Greenhouse gas (GHG) effects, climate change and sustainable development (SO), and Information and Communication Technology (lCT) constitutes a significant component of these complex challenges which until now there is no consensus on whether the energy savings by ICT exceeds the energy consumed by it. In order to develop green and sustainable software the authors adopted techniques known from conventional product design, software engineering, and software development to propose a Reference Model which covers the life cycle of software product providing guidelines and checklist that support developers, users, and other stakeholders involved in the life cycle of the software product.

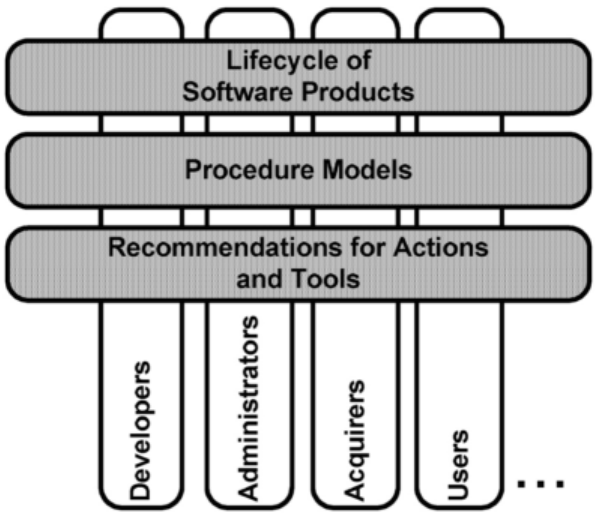

Lifecycle of Software Products
Procedure Models
Recommendations for Actions and Tools
Developers
Administrators
Acquirers
Users
...

Figure 4. Reference Model "Green and Sustainable adopted from [23]

The authors through the proposed model highlighted important factors that affect sustainability throughout the life cycle of software development and suggested some advice on how to handle these factors. The figure below summarizes the factors.

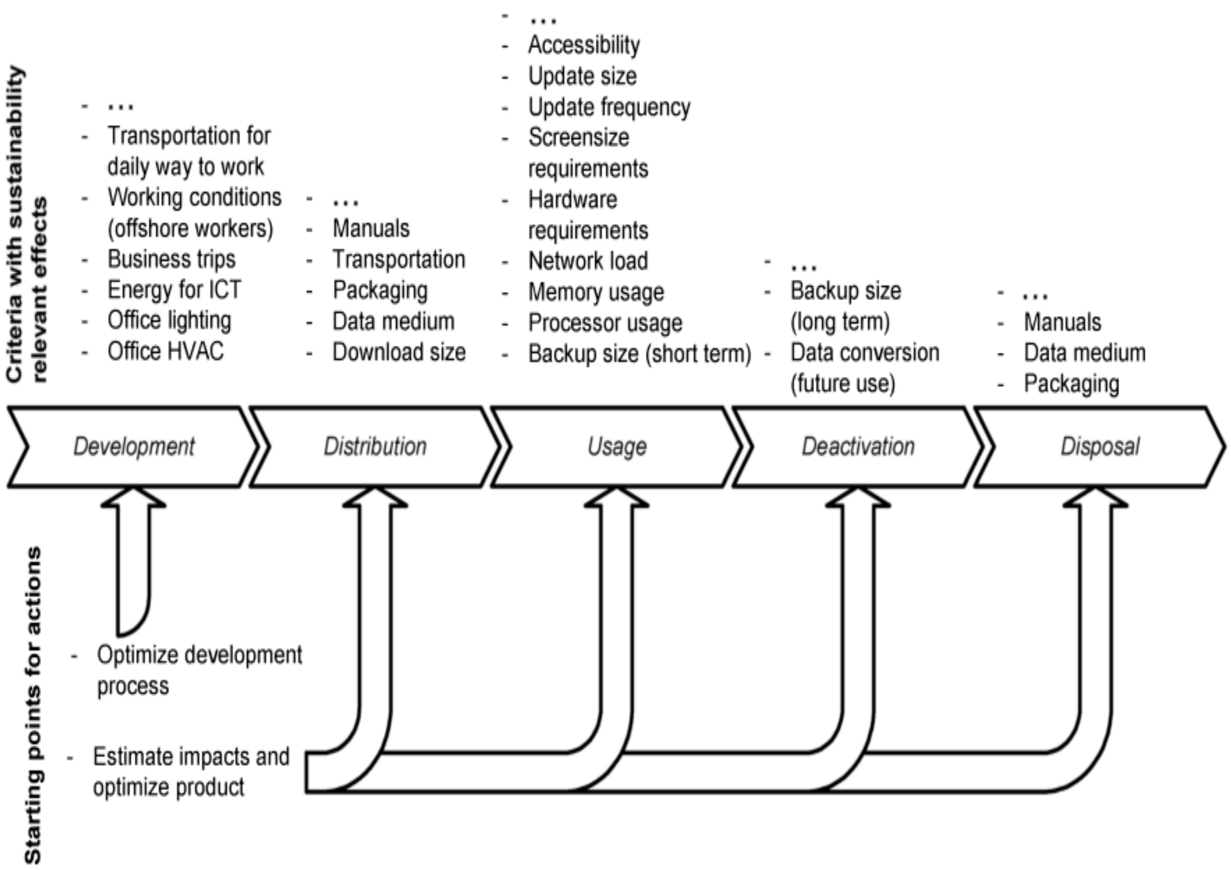


Figure 5. Life Cycle for Software System adopted from [23]

Observation:

The paper provides a starting point for integrating sustainability into software development process with a life cycle approach. The model introduced innovative enhancements such as: Sustainability Reviews & Previews, Process Assessment, Sustainability Journal, and Sustainability Retrospective during software development. Though there is lack of metrics in current model to help stakeholders measure software product during each stage of the product life cycle, which was also highlighted by the authors.

Venters et al. [24] paper goal is to explore emerging definitions of software sustainability from different angles in the field of computational science and engineering in order to contribute to the question, what is software sustainability? The authors stated that in software sustainability, longevity and maintenance are two most important factors for understanding sustainability base on Oxford English dictionary definition for sustainability 'the quality of being sustained', where sustained can be defined as 'capable of being endured' and 'capable of being 'maintained'. Endured is defined as 'continuing to exist' and maintained as ‘being supported.’ The paper also points at different definitions from Johann et al. [23] definition of sustainable software as reducing the negative impact on during development, deployment, usage on the three pillars of sustainability; Amsel et al. [25] considering sustainable software engineering as a process which ‘aims to create reliable, long- lasting software that meets the needs of users while reducing environmental impacts; Venters et al. [24] sustainable software development is defined as the art of developing sustainable software with a sustainable software engineering process so that negative and positive impacts result in and/or are expected to result from the software product over its whole life cycle are continuously assessed, documented, and used for further optimization of the software product. The paper shows that software sustainability is multifaceted with variety of perspective from different dimension. Despite the many definitions of software sustainability, most are limited in scope or too general which are inadequate to provide quantitative pointers that can be used for measuring the performance software. The lack of unanimous definition of this software sustainability will continue to make contributions remain insular and isolated which will ultimately lead to ineffective and inefficient efforts to address the concept. The authors state that a definition aimed at quantitative sustainability which encompass its multi-dimensional nature would result in clear indication to proof a software is sustainable through metrics that can be used in measuring the software.

Observation

The papers show that the starting point of solving the issues of sustainable software is from a unanimous agreement on what it means in software systems, with a clear and quantifiable definition of software sustainability researchers can have more focused and effective impact on how to design and develop sustainable software.

Penzenstadler [26] stated that sustainability view is an emerging concept in software engineering, which usually is still omitted during software development. The author focuses on requirements engineering and quality assurance aspect of software development with aim to support human, social, economic, and environmental pillars of sustainability. The example in the paper shows how to include sustainability concerns during software development. The paper highlights different aspects of sustainability during software lifecycle though the author laid emphasis on system usage aspect, as her hypothesis is that it might have the biggest impact in terms of improvement potential.

Observation:
The paper presents some overview of how to tackle sustainability in software engineering field particularly software development aspect though it didn't present method to tackle sustainability from requirement engineering down to testing (software development life cycle).

Penzenstadler [3] extended abstract provides different definitions of sustainability from software engineering perspective with reference to Mahaux et al. [27] research on the Brundtland definition plus the statement that IT changes behavior and therefore has considerable effect on society and environment and Johann et al. [23] definition. The paper highlights the four aspect of sustainability in software engineering; development process aspect, maintenance process aspect, system production aspect and system usage aspect with the fixed variables.

Observation:
It is an extended abstract with general information on how to tackle sustainability in the field of software engineering.

Naumann et al. [12] pointed out that up to date there is no clarity whether there is balance between the resources and energy savings through ICT and energy consumed by ICT and most research in the past focus more on the environmental aspects of sustainability considering computer hardware. Base on these issues the authors proposed GREENSOFT Model, a model of "Green and Sustainable Software" known as a reference model to address: the reduction of the energy and resource consumption in ICT, as well as the use of ICT to contribute to software development. The proposed model is a reference model, which the authors classified into a new research field called Sustainability Informatics. The Model structures concepts, strategies, activities and processes from software development stage to usage and end of life of the software (Cradle to grave product life cycle). The model classifies effects of software development into first-order (effects of ICT supply), second-order (effects of ICT use) and third-order effects (systemic effects of ICT) and also the use of Sustainability Review and Preview document, Sustainability Journal, Process Assessment for managing sustainability of software product during development.

Observation:

The proposed model provides holistic coverage of different stage in software systems from software development, usage to end of life and provides a general idea of different activities at each stage. The description at each stage tends to focus more on energy efficiency, environmental sustainability and also there is no clear metrics to evaluate software products at each stage. The model is good starting point in building comprehensive model for software sustainability but it is difficult to see how companies can use it in software product development because some parts of the model are too general and others parts are not specifically clear enough for implementation.

Penzenstadler et al. [28] proposed a reference model for sustainability that decomposes sustainability into five dimensions: environmental, individual, social, economic, and technical sustainability. The model is intended for a process engineer who instantiates the model for a software development company or requirements engineer who instantiates it for a specific system under development.

Observation

The model is a good fit for software systems development because it provides useful insights on how to relate sustainability to software products especially in requirement engineering though the examples focus more on environmental sustainability. The introduction of five different sustainability dimensions will also make it easier to measure software products.

Johann et al. [29] presents a generic metric to measure software energy efficiency and a method to apply it in software engineering process. The formula used for energy efficiency is “Useful Work Done/Used Energy”. The authors presented a procedure that allows for energy consumption of specific parts of software to be measured with white box method, which provides programmers the opportunity to find resources intensive code.

Observation

The authors provides useful insight on how to measure software energy efficiency with the examples in the paper and the adoption of whit box measurement of energy efficiency offer system developers opportunities to know which part of software has potential for energy savings.

Becker et al. [14] presents cross-disciplinary initiative to create a common ground and a point of reference for the global community of research and practice in software and sustainability and also highlighted how different research domain from past to present has been trying to tackle the issue of sustainability through collaborative work via conferences and workshop all over the world. The result of the effort is the Karlskrona Manifesto for Sustainability Design, according to the authors "vehicle for a much needed conversation about sustainability within and beyond the software community."

Observation

The paper provides details of a common ground where different stakeholders in software engineering community came together to agree on different aspects of sustainability with regards to software. The initiative of the authors is very important in order to tackle the issues associated with software sustainability because sustainability in this domain is still very unclear and can be interpreted in many ways making it difficult to have a collective effort towards solving the problems of sustainability in software engineering domain.

Bozzelli et al. [30] paper focused on describing and classifying metrics related to software "greenness" present in the software engineering literature through systematic literature review in order to analyze the evolution of those metrics, in terms of type, context, and evaluation methods. The authors highlighted the following metric types; energy, performance, utilization, economic pollution and performance/energy for measuring software energy consumption.

Observation

The metrics presented in the paper mainly focus on software energy consumption and classification provides useful access to the most important software energy consumption metrics in software engineering domain.

Table 3. Analysis of Software Quality Attribute and Sustainability as Non-Functional Requirement

Authors	**Analysis Criteria**					
	Software Quality Attribute	Software Development	Metrics	Sustainability Pillars		
				Economy	Environment	Social
Gorton Ian Chapter 3 [18]	X					
Penzenstadler et al. [11]	X	X			X	
Raturi et al. [20]	X	X		X	X	X

Table 4. Analysis of Sustainability in Software Development and Metrics

Authors	**Analysis Criteria**					
	Software Quality Attribute	Software Development	Metrics	Sustainability Pillars		
				Economy	Environment	Social
Durdik et al. [21]		X				
Erdélyi [22]		X			X	
Kevin Tate [10]		X				X
Johann et al. [23]		X			X	
Venters et. al [24]		X		X	X	X
Penzenstadler[26]		X				
Penzenstadler [3]		X			X	
Naumann et al. [12]		X	X		X	
Penzenstadler et al. [28]		X			X	
Becker et. al [14]		X			X	X

Bozzelli et al. [30]			X			
Johann et al. [29]			X			

2.4 Software Measurement and Sustainability

Measurement of sustainability in software systems will provide useful information for better software management and it will also help companies have better understanding of open ended requirements, uncontrolled changes, product reliability and suitability though currently there are less sustainability metric for software measurement. The proposed software engineering metrics by Albertao et al. [4] can serve as a starting point be used to assess the economic, social and environmental sustainability of software projects based on developments, usage and process related properties using metrics for quality attributes such as modifiability and reusability, portability, supportability, performance, usability, dependability, accessibility, efficiency and project footprint.

Several measurement model used in the assessment of sustainability in software companies are not business driven which has hindered the adoption sustainability during software development. Rini et. al [31] affirmed there are many textbooks available on quality improvement but were surprised by the gap between theory and practice which is due to the fact most literature on software quality lacks the goal-driven nature of business. Business is not just looking for ultimate quality, but for the best quality to be given to other goals, such as timeliness, product features, complexity, or cost. One of the most referenced models when it comes to sustainability metrics for software system is the GREENSOFT Model [12] and for measurement framework in software system the Goal Question Metric (GQM) [32] as an approach to software metrics.

2.4.1 GREENSOFT Model

The GREENSOFT Model is a conceptual reference model for "Green and Sustainable Software", which has the objective to support software developers, administrators, and software users in creating, maintaining, and using software in a more sustainable way [12].

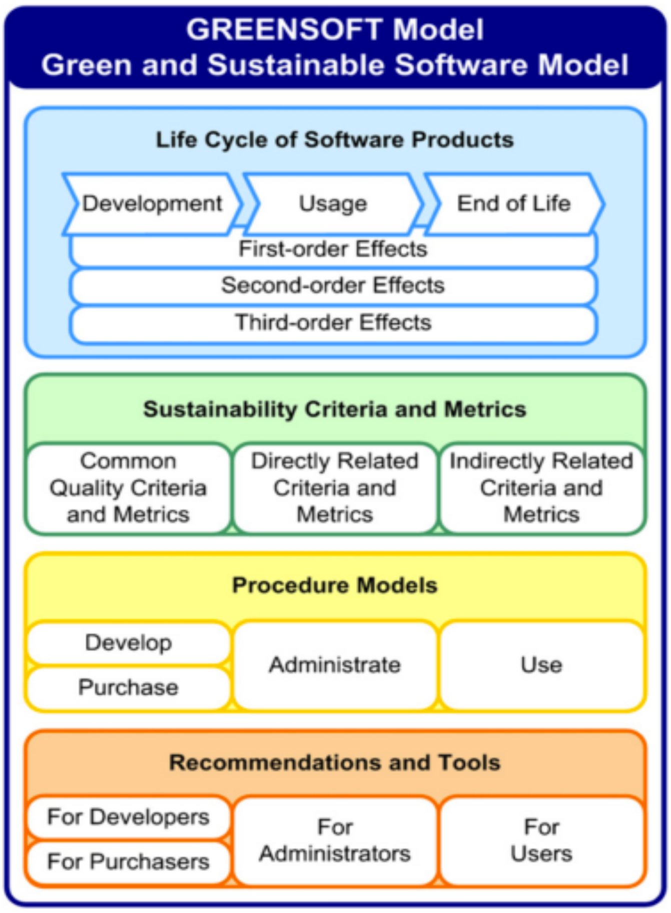
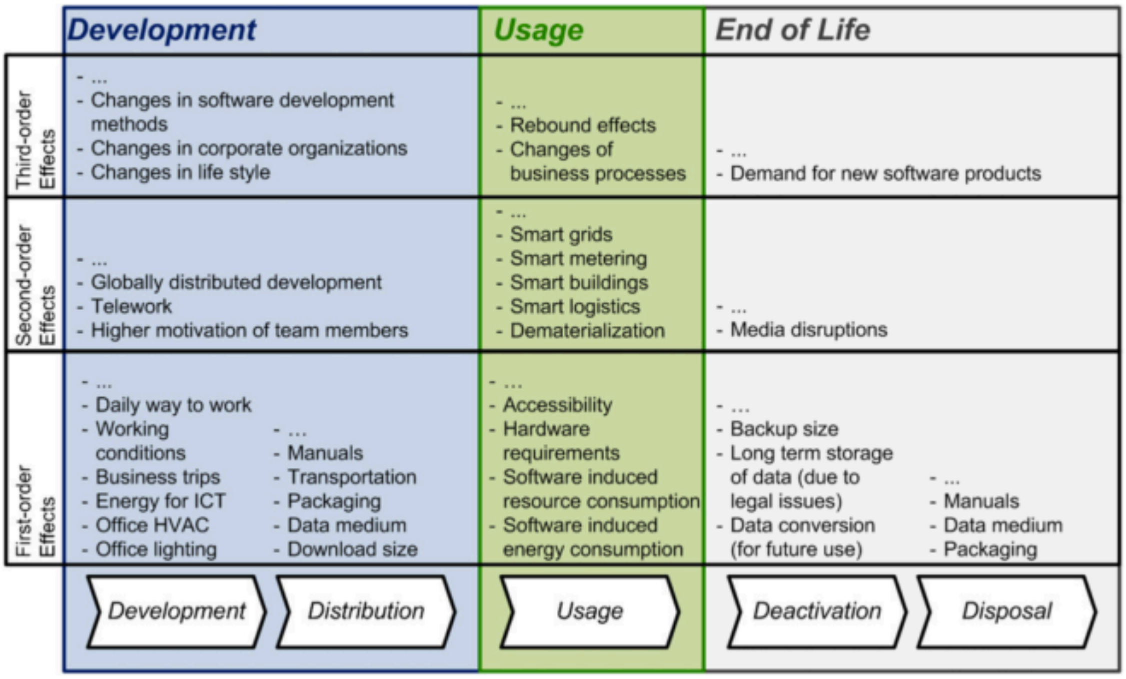

GREENSOFT Model
Green and Sustainable Software Model
Life Cycle of Software Products
Development
Usage
End of Life
First-order Effects
Second-order Effects
Third-order Effects
Sustainability Criteria and Metrics
Common Quality Criteria and Metrics
Directly Related Criteria and Metrics
Indirectly Related Criteria and Metrics
Procedure Models
Develop
Purchase
Administrate
Use
Recommendations and Tools
For Developers
For Purchasers
For Administrators
For Users

Figure 6. **GreenSoft** Model adopted from [12]

The reference model contains a Life Cycle of Software Products with the objective to assess the ecological, social, human, and economic compatibility of a product during its whole life cycle from product development, disposal and recycling.

The second part of the GREENSOFT Model is called Sustainability Criteria and Metrics, which contains common metrics and criteria for software quality measurement, and it allows a classification of criteria and metrics for evaluating software product sustainability.

The last component of the model contains Recommendations and Tools. These support stakeholders in applying sustainable practices when developing, purchasing, administrating or using software products.



	Development	Usage	End of Life
Third-order Effects	- ... - Changes in software development methods - Changes in corporate organizations - Changes in life style	- ... - Rebound effects - Changes of business processes	- ... - Demand for new software products
Second-order Effects	- ... - Globally distributed development - Telework - Higher motivation of team members	- ... - Smart grids - Smart metering - Smart buildings - Smart logistics - Dematerialization	- ... - Media disruptions
First-order Effects	- ... - Daily way to work - Working conditions - Business trips - Energy for ICT - Office HVAC - Office lighting; - ... - Manuals - Transportation - Packaging - Data medium - Download size	- ... - Accessibility - Hardware requirements - Software induced resource consumption - Software induced energy consumption	- ... - Backup size - Long term storage of data (due to legal issues) - Data conversion (for future use); - ... - Manuals - Data medium - Packaging
	Development, Distribution	Usage	Deactivation, Disposal

Figure 7. Software products product life cycle adopted from [12]

Figure 7 highlight the different activities in first, second and third order effects base on software product life cycle to show an example of which activity falls in a particular stage.

2.4.2 GQM Model

Goal, Question, Metric (GQM) is an approach to software metrics for creating and measuring link between goals and strategies across organization with software being developed that allows measurement-based decision-making within the organization [31].

2.4.2.1 Background and Origin of GQM

Differding et al. [33] stated that the Goal Question Metric Model (GQM) was developed in response to the need for a goal-oriented approach that would support the measurement of processes and products in the software engineering domain. The GQM Paradigm (sometimes called the GQM approach) supports a top-down approach to defining the goals behind measuring software processes and products, and using these goals to decide precisely what to measure (choosing metrics).

GQM method was originally created by V. Basili and D. Weiss, and expanded with many other concepts by D. Rombach [31]. It was first developed in 1984 at the University of Maryland and extended as part of the TAME project. Research into using and improving the GQM Paradigm has also been in progress at the University of Kaiserslautern since 1992 and at the Fraunhofer Institute for Experimental Software Engineering since 1996

[33]. Over the years software engineering researchers and practitioners in many different contexts with good success have applied the GQM Paradigm[33]. However, nearly every user of GQM has tailored the paradigm to suit his or her specific needs, resulting in many different views.

2.4.2.2 Phases of GQM

GQM method contains four phases [31]:

1. The Planning phase, during which a project for measurement application is selected, defined, characterized, and planned, resulting in a project plan.
2. The Definition phase, during which the measurement program is defined (goal, questions, metrics, and hypotheses are defined) and documented.
3. The Data Collection phase, during which actual data collection takes place, resulting in collected data.
4. The Interpretation phase, during which collected data is processed with respect to the defined metrics into measurement results, that provide answers to the defined questions, after which goal attainment can be evaluated.

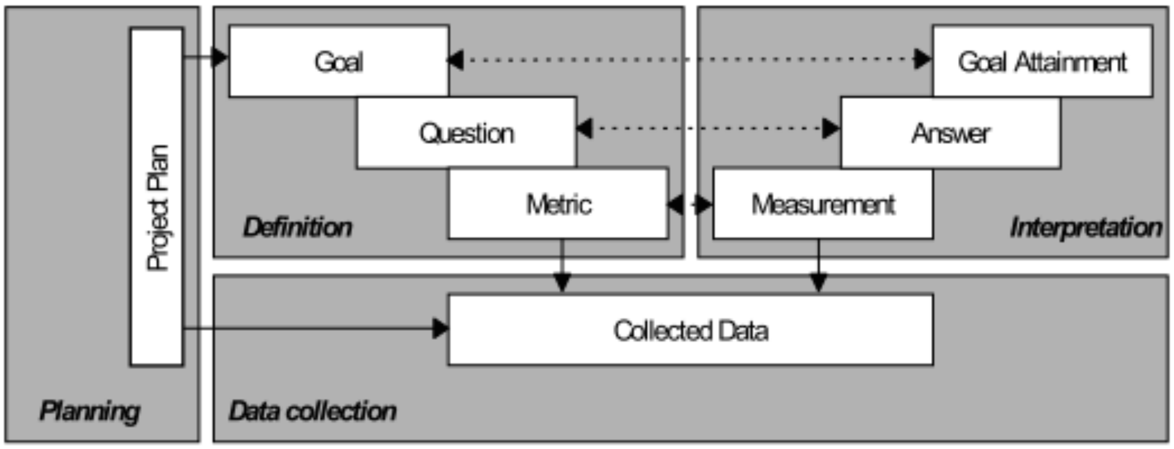


Figure 8. Four phases of GQM Model adopted from [31]

The planning phase ensure fulfillment of all basic requirements in the project to make a GQM measurement a success. In the definition phase all GQM deliverables are developed, mainly based on structured interviews or other knowledge acquisition techniques. The definition phase identifies a goal, all questions, related metrics and expectations (hypotheses) of the measurements. During data collection phase, the data collection forms are defined, filled-in and stored in a measurement database. During the interpretation phase, the measurements are used to answer the stated questions, and these answers are again used to see whether the stated goals have been attained.

2.4.2.3 Activities and Steps in GQM

GQM has seven steps, the three first steps are very important as it affects the outcome of all other steps [34].

1. Develop set of Goals: Develop goals on corporate, division, or project level. These goals can be established from brainstorming sessions involving project team members, or they may be set by organizational goals or from stakeholder's requirements.
2. Develop a set of questions that characterize the goals. From each goal a set of questions is derived which will determine if each goal is being met.
3. Specify the metrics needed to answer the questions. From each question from step two it is determined what needs to be measured to answer each question adequately.
4. Develop Mechanism for data collection and analysis which should determine:
    - Who will collect the data?
    - When will the data be collected?
    - How can accuracy and efficiency be ensured?
    - Who will be the audience?

5. Collect, Validate and Analyze Data: The data may be collected manually or automatically. Metrics data can be portrayed graphically to enhance understanding.
6. Analyze in a post Mortem fashion: Data gathered is analyzed and examined to determine its conformity to the goals. Based on the findings here recommendations are made for future improvements.
7. Provide feedback to stakeholders: The last step, providing feedback to the stakeholders is a crucial step in the measurement process. It is essentially the purpose behind the previous six steps. Feedback is often presented in the form of one goal per page with the questions, metrics and graphed results.

2.4.2.4 GQM Measurement Levels

1. **Conceptual level (Goal):** A goal is defined for an object, for a variety of reasons, with respect to various models of quality, from various points of view and relative to a particular environment [34].
2. **Operational level (Question):** A set of questions is used to define models of the object of study and then focuses on that object to characterize the assessment or achievement of a specific goal [34].
3. **Quantitative level (Metric):** A set of metrics, based on the models, is associated with every question in order to answer it in a measurable way [34]. Victor et al. [32] added that the metric can be :
   - Objective: if they depend only on the object that is being measured and not on the viewpoint from which they are taken, examples: number of versions of a document, staff hours spent on a task, size of a program.
   - Subjective: if they depend on both the object that is being measured and the viewpoint from which they are taken, example readability of text, level of user satisfaction.

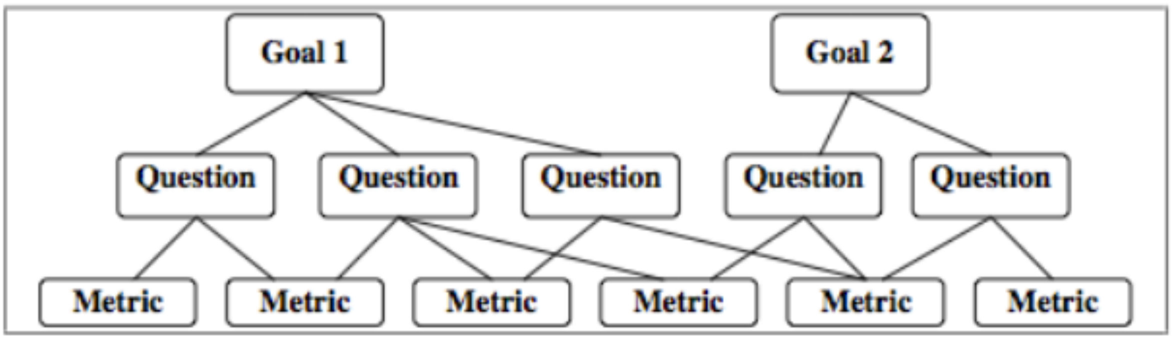


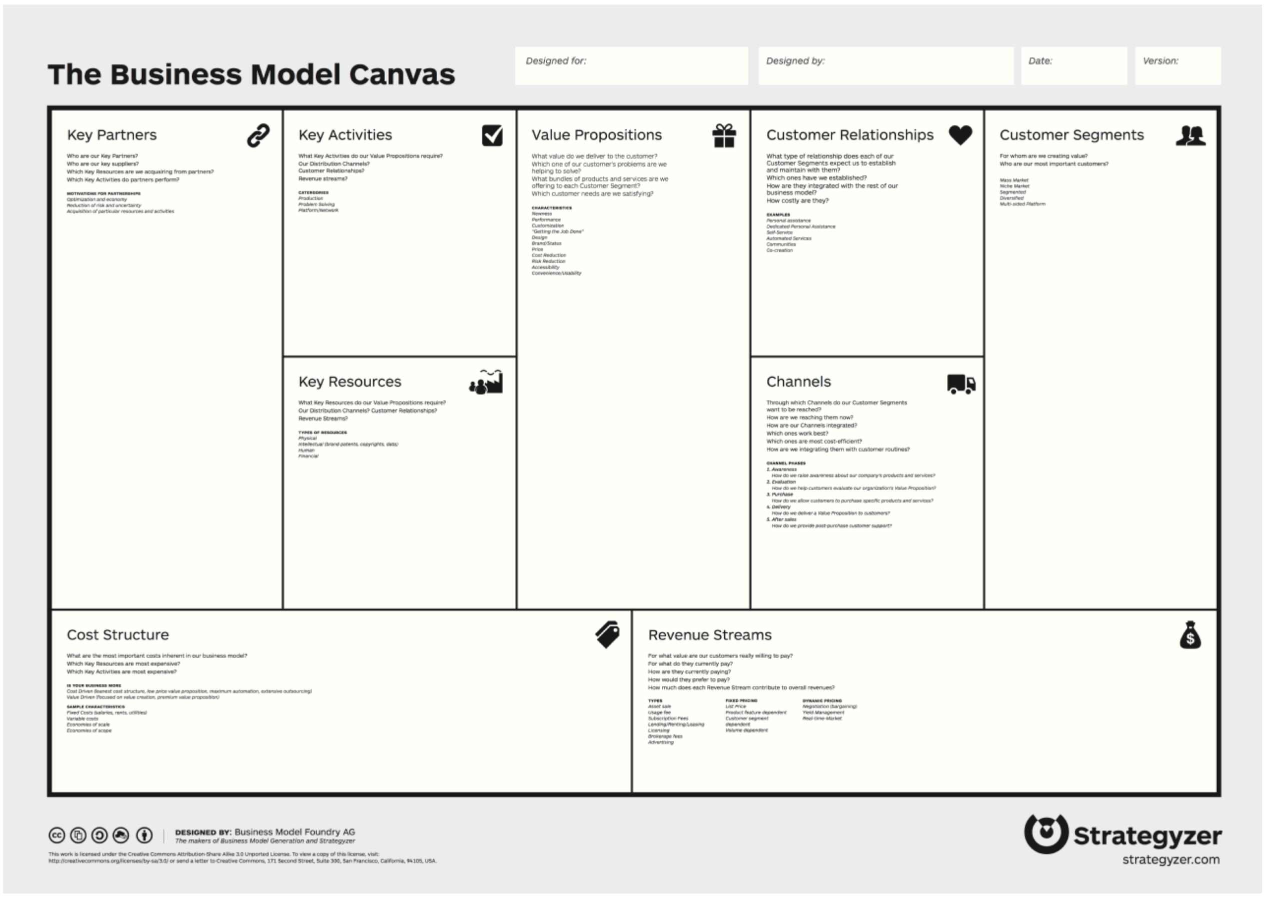
Figure 9. GQM Measurement Model Level adopted from [35]

Each company goals can be linked to one or different set of questions for assessment. Questions try to characterize the object of measurement (Product, process, resource) with respect to selected quality issue and to determine its quality from the selected viewpoint. Questions are then linked to one or more metrics as a way to evaluate if the desired goal is accomplished.

2.5 Business Analysis and Requirement Engineering for Software Sustainability

The goal of this section is to look at different activities that has the potential to support sustainability during development of new or documenting existing business models for software companies and the processes involve during software system development that involve all stakeholders.

- **Business Model Canvas**: This is a strategic management tool that allows businesses and organization to describe, design, challenge, invent and pivot their business model highlighting key partners and activities, value preposition, customer relationship and segments, key resources, revenue stream, cost structure and distribution channel [36].



The Business Model Canvas

Designed for:

Designed by:

Date:

Version:

Key Partners

Who are our Key Partners?
Who are our key suppliers?
Which Key Resources are we acquairing from partners?
Which Key Activities do partners perform?

MOTIVATIONS FOR PARTNERSHIPS
Optimization and economy
Reduction of risk and uncertainty
Acquisition of particular resources and activities

Key Activities

What Key Activities do our Value Propositions require?
Our Distribution Channels?
Customer Relationships?
Revenue streams?

CATEGORIES
Production
Problem Solving
Platform/Network

Key Resources

What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

TYPES OF RESOURCES
Physical
Intellectual (brand patents, copyrights, data)
Human
Financial

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?
What bundles of products and services are we offering to each Customer Segment?
Which customer needs are we satisfying?

CHARACTERISTICS
Newness
Performance
Customization
"Getting the Job Done"
Design
Brand/Status
Price
Cost Reduction
Risk Reduction
Accessibility
Convenience/Usability

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Channels

Through which Channels do our Customer Segments want to be reached?
How are we reaching them now?
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?

CHANNEL PHASES
1. *Awareness*
   How do we raise awareness about our company's products and services?
2. *Evaluation*
   How do we help customers evaluate our organization's Value Proposition?
3. *Purchase*
   How do we allow customers to purchase specific products and services?
4. *Delivery*
   How do we deliver a Value Proposition to customers?
5. *After sales*
   How do we provide post-purchase customer support?

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platform

Cost Structure

What are the most important costs inherent in our business model?
Which Key Resources are most expensive?
Which Key Activities are most expensive?

IS YOUR BUSINESS MORE
Cost Driven (leanest cost structure, low price value proposition, maximum automation, extensive outsourcing)
Value Driven (focused on value creation, premium value proposition)

SAMPLE CHARACTERISTICS
Fixed Costs (salaries, rents, utilities)
Variable costs
Economies of scale
Economies of scope

Revenue Streams

For what value are our customers really willing to pay?
For what do they currently pay?
How are they currently paying?
How would they prefer to pay?
How much does each Revenue Stream contribute to overall revenues?

TYPES	FIXED PRICING	DYNAMIC PRICING
Asset sale	*List Price*	*Negotiation (bargaining)*
Usage fee	*Product feature dependent*	*Yield Management*
Subscription Fees	*Customer segment dependent*	*Real-time-Market*
Lending/Renting/Leasing	*Volume dependent*	
Licensing		
Brokerage fees		
Advertising		

Figure 10. Business Model Canvas adopted from [36].

- **Flourishing Business Canvas**: It is a strategic management tool for developing new or existing business models with the concern about integration of environmental and social sustainability into organization [37]. It serves as a visual chart with elements describing company's or product value proposition, infrastructure, customers and finances.

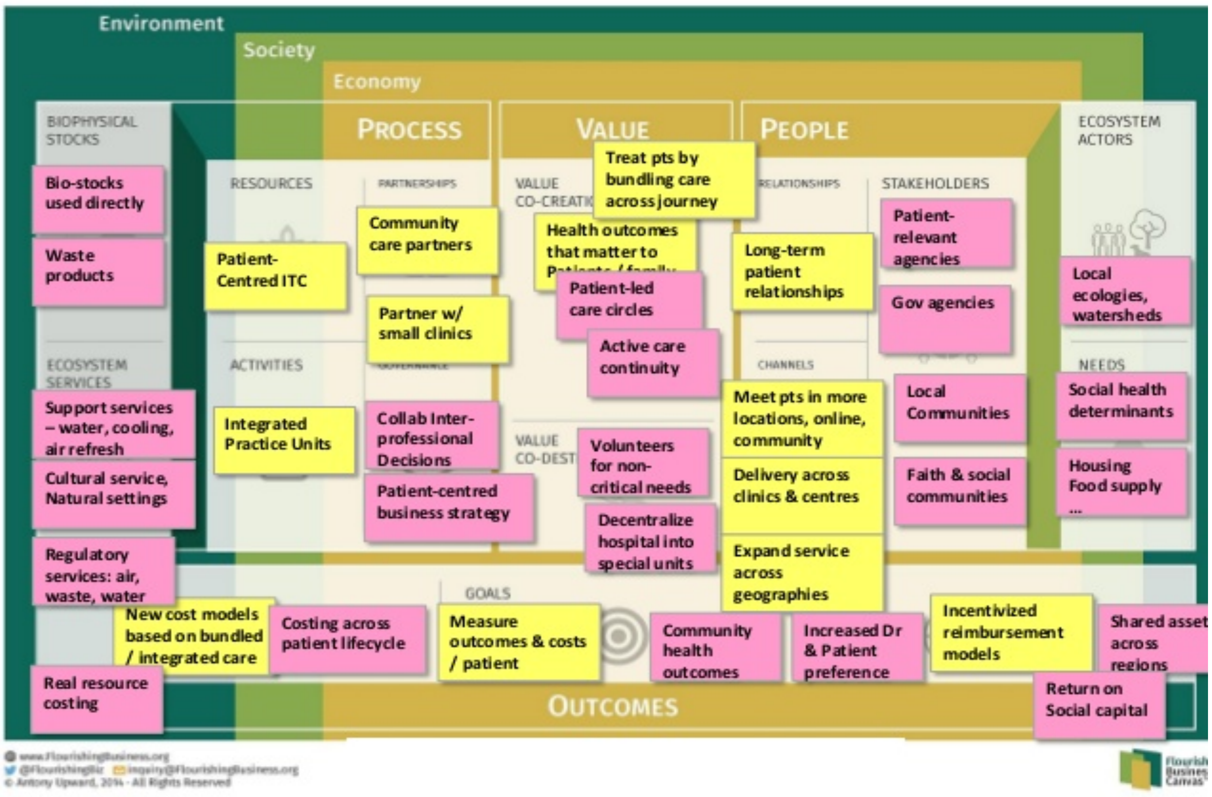

Environment
Society
Economy
BIOPHYSICAL STOCKS
Bio-stocks used directly
Waste products
ECOSYSTEM SERVICES
Support services – water, cooling, air refresh
Cultural service, Natural settings
Regulatory services: air, waste, water
PROCESS
RESOURCES
PARTNERSHIPS
Patient-Centred ITC
Community care partners
Partner w/ small clinics
ACTIVITIES
GOVERNANCE
Integrated Practice Units
Collab Inter-professional Decisions
Patient-centred business strategy
VALUE
VALUE CO-CREATION
Treat pts by bundling care across journey
Health outcomes that matter to Patients / family
Patient-led care circles
Active care continuity
VALUE CO-DESTRUCTION
Volunteers for non-critical needs
Decentralize hospital into special units
PEOPLE
RELATIONSHIPS
STAKEHOLDERS
Long-term patient relationships
Patient-relevant agencies
Gov agencies
CHANNELS
Meet pts in more locations, online, community
Delivery across clinics & centres
Expand service across geographies
Local Communities
Faith & social communities
ECOSYSTEM ACTORS
Local ecologies, watersheds
NEEDS
Social health determinants
Housing Food supply ...
GOALS
New cost models based on bundled / integrated care
Costing across patient lifecycle
Measure outcomes & costs / patient
Community health outcomes
Increased Dr & Patient preference
Incentivized reimbursement models
Shared asset across regions
Real resource costing
Return on Social capital
OUTCOMES
Flourishing Business Canvas

Figure 11. Example of Flourishing Buiness Model Canvas adopted from [37]

- **Goal Model:** It is an component in requirement engineering used in business analysis which include context, scenarios and stakeholder analysis in order to have a rationale for requirements, identify stable information and guide requirement elaboration [38]. The goal model help organization, system developers, users and external regulators to coordinate their actions in order to provide a common result through goal directed view [39].
- **System Vision:** It describes idea or value of a product. It outlines the view of product or services based on stakeholders needs and requirement [40].

- **Sustainability Analysis:** Sustainability analysis examines products or services based on sustainability consideration of the economical, environmental and social consideration in order to evaluate the system impact and identify areas of improvement [41].

Table 5. Summary of Contributions from Researchers in State of the Art Study

Source and Authors	**Contribution**
Penzenstadler et al. [11]	Add sustainability as a nonfunctional requirement
Raturi et al. [20]	Proposed Nonfunctional Requirement Framework (NFR) framework as means to correctly obtain sustainability requirements
Durdik et al. [21]	Use of software sustainability guidelines to help stakeholders develop sustainable software
Erdélyi [22]	Highlights design and analysis, system requirements, implementation and test as key points that affect software sustainability
Kevin Tate [10]	Sustainable software development as mindset (principles) and an accompanying set of practices that enable a team to achieve and maintain an optimal development pace indefinitely
Johann et al. [23]	Life cycle model to develop green software products.
Venters et al. [24]	Defines sustainability from different perspective in the field of computational science and engineering
Penzenstadler[26]	Use requirement engineering to help create sustainable software
Naumann et al. [12]	GREENSOFT Model to help energy reduction and resource consumption in ICT.
Penzenstadler et al. [28]	Propose reference model that decomposes sustainability into five dimensions: environmental, individual, social, economic, and technical sustainability.
Johann et al. [29]	Presents a generic metric to measure software energy efficiency and a method to apply it in software engineering process.

Becker et al. [14]	Propose cross-disciplinary initiative to create a common ground and a point of reference for the global community of research and practice in software and sustainability through Karlskrona Manifesto for Sustainability Design
Bozzelli et al. [30]	Describes and classify metrics related to software “greenness” present in the software engineering
Albertao et al. [4]	Proposed software engineering metrics based on software quality attribute.
Differding et al. [33]	Goal Question Metric as an approach to software metrics

3 METHODOLOGY

The research process is presented in this chapter that shows the research methodology applied, the methodology approach, data collection through interview of experts and companies and different data analysis method. The research is grounded in the overall design science research methodology with qualitative research method used for gathering information through interviews. It also contains the research organization specifying details of companies and individual categories involved in the interview process and justification for choosing grounded theory as an approach used in this research work.

3.1 Design Science Research

Design science is the design and investigation of artifacts in context and design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence [42]. Design science research is centered on the improvement of designed artifacts functional performance.

3.1.1 Design Science Research Methodology

Design science research methodology includes these three elements: conceptual principles to define what is meant by design science research, practice rules, and a process for carrying out and presenting the research [43]. The below diagram presents the research methodology processes used in this research work.

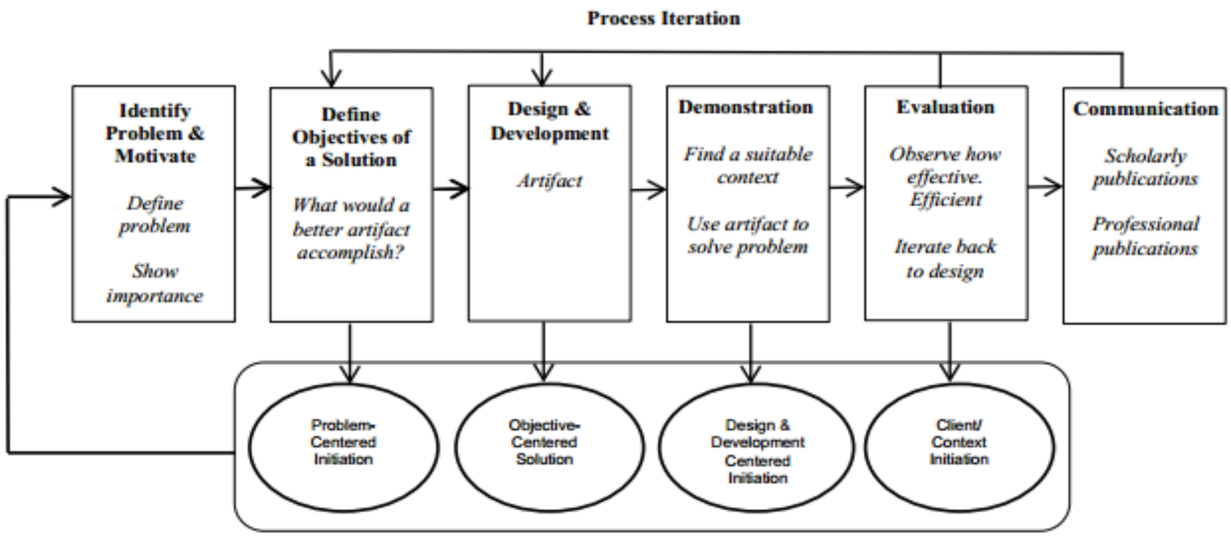

Process Iteration
Identify Problem & Motivate
Define problem
Show importance
Define Objectives of a Solution
What would a better artifact accomplish?
Design & Development
Artifact
Demonstration
Find a suitable context
Use artifact to solve problem
Evaluation
Observe how effective. Efficient
Iterate back to design
Communication
Scholarly publications
Professional publications
Problem-Centered Initiation
Objective-Centered Solution
Design & Development Centered Initiation
Client/ Context Initiation

Figure 12. Design Science Research Methodology adopted from [43]

3.1.2 Design Science Research Guidelines

The table below provides general guidelines for design science research from the beginning of the research to the end [44].

Table 6. Guidelines for Design Science Research adopted from [44]

Guideline	**Description**
1. Design as an Artifact	Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
2. Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
3. Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
4. Research Contributions	Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
5. Research Rigor	Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
6. Design as a search process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
7. Communication of research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

3.2 Qualitative Research

Qualitative Research is primarily an exploratory research that is used to gain an understanding of underlying reasons, opinions, and motivations. It provides insights into the problem or hypotheses for potential quantitative research[45]. Qualitative research is characterized by its aims, which relate to understanding some aspect of social life, and its methods which (in general) generate words, rather than numbers, as data for analysis [46]. It seeks to understand a given research problem or topic from the perspectives of the local population it involves. Qualitative research is especially effective in obtaining culturally specific information about the values, opinions, behaviors, and social contexts of particular populations [45]. The strength of using qualitative research is the fact that it provides complex textual description of how people experience a given research issue because it uncover trends in thought and opinions and dive deeper into the problem.

3.2.1 Qualitative Research Approaches

Research process will reflect the methodological approach that is adopt in the research; different approach involves different set of assumption about what sort of information are important. There are different approaches used in qualitative research [47] such as Ethnography, Grounded Theory, Interpretative Phenomenological analysis (IPA), Discourse analysis, Conversation analysis, Content analysis, Narrative analysis. For the purpose of this thesis, the focus will be on grounded theory.

Grounded Theory: It is define as the discovery of theory from data systematically obtained from social research (Glaser and Strauss 1967) [48]. This methodology originated with the work of Glaser and Strauss on the interactions between health care professionals and dying patients. The main feature is the development of new theory through the collection and analysis of data about a phenomenon. The below are features of grounded theory:

1. Simultaneous collection and analysis of data
2. Creation of analytic codes and categories developed from data and not by pre-existing conceptualizations (theoretical sensitivity)
3. Discovery of basic social processes in the data
4. Inductive construction of abstract categories
5. Theoretical sampling to refine categories
6. Writing analytical memos as the stage between coding and writing
7. The integration of categories into theoretical framework

3.2.2 Data Collection

The three main qualitative methods are participant observation, in-depth interviews and focus groups, each methods is best suitable for obtaining a specific type of data [49].

- Participant observation is suited for gathering data on naturally occurring behaviors in their usual context.
- In-depth interviews are best for collecting data on individual's personal experiences, perspective and history, especially when exploring sensitive research subject.
- Focus groups are appropriate in extracting data on cultural norms of a group and generating broad overviews of issues of concern to the cultural groups or subgroups.

3.2.3 Data Analysis

1. **Coding** is a process for both categorizing qualitative data and for describing the implications and details of these categories. It is initially done considering the data in minute detail while developing some initial categories and later moves to more selective coding where the researcher systematically codes with respect to a core concept [50] [51].

2. **Memoing** is a process for recording the thoughts and ideas of the researcher as they evolve throughout the study. It is considered as extensive marginal notes and comments which early in the process tend to be very open while later on tend to increasingly focus in on the core concept [50] [51].

3. **Integrative diagrams and sessions** are used to pull all of the detail together, to help make sense of the data with respect to the emerging theory. The diagrams can be any form of graphic that is useful at that point in theory development. This integrative work is best-done in-group sessions where different members of the research team are able to interact and share ideas to increase insight [50].

3.2.4 Interview Research Organization

Six companies including software and non software companies from Finland and Russia where interviewed mainly focusing on those from the top management level such as CEOs, CIOs, Operation Managers, Business Development Managers with the aim to find out software companies perception of sustainability from the management side of each companies. 13 experts (Software developers, IT Professors, Masters and PhD students from software engineering department) to find out how ICT specialist or practitioner perceive sustainability of software systems both during after software development.

The categorization of software and non-software companies, small, medium and large companies is explained in the table below.

Table 7. Company Categorization

Categorization			
Company Size	***Small***	***Medium***	***Large***
	1-9	10-50	>50
Company Type	***Software Company***	***Non-Software Company***	
	Main Business is software development	Main Business is not software development but uses software products to offer services	

3.2.4.1 Interview

There were two round interviews for companies and ICT experts. The first round interview theme was "management perception of sustainability" and the second round interview team was "ICT specialist perception of sustainability". Table 2 shows different roles of those involved the first and second round of interview.

Table 8. Themes and Roles

Round	**Theme**	**Role**
1	Management perception of sustainability	CEOs: Owners and responsible for all day-to-day management decisions and for implementing the Company's long and short-term plans.

		Operation Manager: oversee the production of goods and/or provision of services, inventory, purchasing and supplies
		Business Development Manager: Head of a particular branch or business unit, who is in charge of developing the company's business, bringing in new clients and proposing new ways of making the company profitable.
2	ICT specialist perception of sustainability	Software Developers: Programing and creating/testing software applications
		ICT Professors: Educating future software developers through lectures and practical classes.
		ICT Students: Masters and PhDs students working with software systems development

3.3 Justification for choosing Grounded Theory Approach

Grounded theory approach fits best my thesis topic because of its suitability to observe and explore organizations and phenomena, which provides intuitive appeal, forester creativity, provides potential to conceptualize with data depth and richness. A notable advantage of the grounded theory is in its systematic approach to data analysis. Glaser (1978) defined grounded theory as "systematic generating of theory from data that itself is systematically obtained from social research" [52]. Karen Locke [51] also showed the justification of the use of grounded theory in organization and management.

Supporting theorizing of 'new' substantive areas: “The naturalistically oriented data collection methods as well as the approach's theory-building orientation permit the investigation and theoretical development of new substantive areas as they 'arrive' on the organizational scene” [51]
Sustainability in software systems is still a ‘new’ field that has less information; the use of grounded theory will provide useful insight to understand the current research problem and how to address the issue in order to find tangible solutions.

Capturing complexity: The grounded theory style adapts well to capturing complexities of the context in which action unfolds, enabling researchers to better understand all that may be involved in a particular substantive issue [51].
Sustainability in itself is complex to define and adopting it as quality attribute in software system makes it even more difficult but grounded theory approach makes it better to grasp this complex and multi face issue of sustainability.

Enlivening mature theorizing: The grounded theory building approach has been used to bring a new perspective and new theorizing to mature established theoretical areas, enlivening and modifying existing theoretical frameworks
The thesis topic provides new way of thinking and the possibility of new knowledge when it comes to software systems and how it should be measured with consideration of sustainability which is still not well defined in an establish field like software engineering.

4 RESULTS

Chapter four presents the results of the interview from software/non software companies and ICT specialist, the reasons behind companies and ICT specialists perceptions of sustainability and sustainability in software systems. The chapter also covers summary of the outcome from all the interviews.

4.1 Result of Interviews

The results are categorized into sustainability awareness and meaning of sustainability, sustainability consideration during software development and measurement of sustainability in software systems, value of sustainability to company. For categorization software companies will be marked as SC and non-software companies as NSC and using of alphabet as Id for each participant. There are six companies interviewed and twelve ICT specialists.

4.1.1 Sustainability Awareness

Table 9. **Results** for Companies Sustainability Awareness

ID	Company Type	Sustainability Awareness	Meaning of Sustainability
A	SC	Yes	Energy Efficiency
B	SC	Yes	Protect Environment
C	NSC	Yes	Protect Environment
D	SC	No	Not sure
E	SC	No	Not sure
F	NSC	Yes	Protect Nature

The result from the table shows that A, B, C, F companies only think of sustainability from the environmental pillar and they don’t consider it to also involve the economy and social pillar of sustainability. A considers it as energy efficiency because it has some knowledge of sustainability from ICT workshop.

Table 10. Results for ICT Specialist Sustainability Awareness

ID	Sustainability Awareness	Meaning of Sustainability
A	Yes	Protect the planet
B	Yes	Protect the environment
C	Yes	Protect the environment
D	Yes	Protect the environment
E	Yes	Energy Efficiency
F	Yes	Energy Efficiency
G	Yes	Protect the environment
H	No	No Idea
I	No	No Idea
J	No	No Idea
K	No	No Idea

The above result is similar to that of the companies A, B, C, D, G understands sustainability to be mean only protecting the environment, E, F sees it as energy efficiency and H, I, J, K has no idea about sustainability.

4.1.2 Results for Sustainability In Software

Table 11. Results of ICT Specialist for Sustainability in Software

ID	Sustainability in Software Development?	Measure Software Sustainability?
A	Not considered	Don’t know how to measure
B	Not considered	Don’t know how to measure
C	Not considered	Don’t know how to measure
D	Not considered	Don’t know how to measure
E	Not considered	Don’t know how to measure
F	Not considered	Don’t know how to measure
G	Not considered	Don’t know how to measure
H	Not considered	Don’t know how to measure
I	Not considered	No but maybe software footprint

J	Not considered	No, using energy efficiency
K	Not considered	No maybe using performance

The result shows that all the interviewed ICT specialist don't consider sustainability in software development and only I, J, K provides some clue as to what to measure for sustainability in software systems and others don't know how it can be measured.

Table 12. Results of Companies for Sustainability in Software

ID	**Company Type**	**Sustainability in Software Development?**	**Measure Software Sustainability?**
A	SC	Not considered	Don't know how to measure
B	SC	Not considered	Don't know how to measure
C	NSC	Not considered	Don't know how to measure
D	SC	Not considered	Don't know how to measure
E	SC	Not considered	Don't know how to measure
F	NSC	Not considered	Don't know how to measure

All the companies interviewed didn't considered sustainability in software development and don't know how to measure sustainability of software.

4.1.3 Value of Sustainability to Companies

Table 13. **Results** of Sustainability Value to Companies

ID	**Company Type**	**Sustainability Value to your company?**
A	SC	Safe Cost
B	SC	Continuous Growth
C	NSC	Not sure
D	SC	Not sure
E	SC	Not sure
F	NSC	Not sure

Out of all the companies interviewed only A and B sees the value of sustainability to safe cost and company growth. This shows that there is lack of enough information for companies to understand the value of sustainability in their companies especially as companies producing and using software.

4.2 Summary of the Interview Results

The first section on sustainability awareness shows that companies and ICT specialist only view sustainability from the environmental pillar due to lack of understanding that sustainability covers the economy, environment and society. This shows that sustainability is still viewed as a minor component in software system, in order to encourage or promote sustainability as an accepted valid value creating activities for software companies and ICT specialist, there is need to show organizations that it doesn't not only cover energy efficiency but can help save cost, improve company's brand reputation and publicity and aid innovations on news ways of developing software systems and company operations that can aid growth and development for the companies and ICT experts. Government also needs to setup rules and regulations that are connected to sustainability to encourage companies with sustainable products/services and also provide incentives to encourage such activities by companies and ICT experts.

The second section about sustainability in software systems show total absence of sustainability in software development or measurement since the concept of sustainability is not clear, complex and the interviewed participants are not aware of what requirements of sustainability to consider during software development. Software sustainability is very complex, hard to access and measure because it involves different set of processes and stakeholders whose activities impact are interrelated with one another and its hard to trace effects of these interrelated activities, which has an impact on software sustainability that affects the society at macro level. There are metrics that claim to measure sustainability but are too confusing and complicated which makes their application in organizational context unclear and difficult. It is therefore important to highlight starting from small scale what are sustainability requirements for software systems, how can it be added to software development lifecycle with guidelines on how to measure sustainability in software systems. Companies can address complexity in their operating environment by seeking to reduce it and by becoming more resilient by facilitating better alignment about sustainability goals and responsibilities among different stakeholders [53]. The research community also has a very important role to play in this aspect because most of the propose solutions as seen in the state of the art summary focus on different proposed solutions without a simplified holistic solution that brings the technical and nontechnical stakeholders (users, sponsors (management)) in software development together in creating a sustainable software system.

The last sections shows why most software companies don't consider sustainability in their companies because most of them don't see the added value for the company's development or corporate image. Companies are mainly concerned with meeting the needs of customers and other stakeholders today. Sustainability's long-term orientation is challenging to embrace within companies where short-term deadlines, returns, and customer needs take high priority [53]. This sort of mentality will hinder software companies from growing and serving customers who care about company's environmental footprint because these days with the increasing awareness about climate change users are starting to be aware about these things. There are other benefits for companies to adopt sustainability in their business activities such as tax incentives from government, employee retention (Survey research shows that employees would rather work for sustainable firms. And some would even forego higher earnings to do so [54]) and new revenue opportunities through innovation. According to Tima Bansal [54], Organizations that successfully balance their own private interests with the needs of society will consistently attract and keep the best people; generate cheaper and more stable capital; and create innovative and enduring products. Corporations that fail to accommodate public interests through their private actions expose themselves to significant risks: they will lose customer markets; experience-limited access to financial capital; and suffer disruptions to their supply of goods and services.

The above results from the interview summary shows that there is total lack of understanding of what sustainability means in software systems, for companies operations and how it can be measured which warrant a need to research on how to quantify sustainability in software systems and measure it in order to aid the understanding of what sustainability means in software systems. The subsequent chapters covers quantifying sustainability as quality attribute and proffer a starting point of measurement.

5 SUSTAINABILITY AS QUALITY ATTRIBUTE

Sustainability as software quality attribute is not added to current standard in the software engineering domain, examples are ISO/IEC 9126 Software engineering (Product quality) with no reference included for sustainability as non-functional requirement (NFR) and IEEE standard 830-1993 Recommended Practice for Software Requirements Specifications with No reference included for NFR other than performance, design requirements or other requirements. Currently most software companies don't realize the benefit of sustainability to their software products and it is totally ignored during software development process, which is reflected in the interview results in chapter 4.

The adoption of sustainability as a quality attribute into IEEE standard covering the three main dimension (economic, social, environment) of sustainability will help speed up the integration of sustainability into software systems. It will also encourage software companies to work in ways that are environmentally friendly because it will become socially and economically demanding through regulations, customers awareness of sustainability and for their corporate image. The table below shows definitions of most software quality attributes.

5.1 Software Quality Attribute

Table 14. **General** Quality Attributes adopted from Microsoft Application Architecture adopted from [2]

Category	Quality attribute	Description
Design Qualities	Conceptual Integrity	It defines the coherence and consistency of the overall design, which includes how components are designed, coding style and variable naming.
	Maintainability	This is the ability of system to easily undergo changes that can impact services, components, interface when meeting new business requirements or changing functionality and fixing errors.

	Reusability	It defines the ability of system components to be suitable for use in other application or scenarios. It helps to minimize component duplication and reduce implementation time.
Runtime Qualities	Availability	This is the proportion of time the system is working. It can be measured by the percentage of the total system downtime over a predefined period.
	Interoperability	This is the ability of system to function properly by communicating with other systems and exchanging information with other external systems written by external parties.
	Manageability	This is the extent to which a system is easily manageable through sufficient exposed for use in monitoring systems, debugging and performance tuning.
	Performance	This is the extent of system responsiveness during execution of a task within a time interval, which can be measured using latency or throughput.
	Reliability	The ability of system to remain operational over a period of time. This can be measured based on probability that the system will not fail to perform its function over a specific interval.
	Scalability	The ability of system to handle more workload without depletion in performance.
	Security	Ability of system to prevent unauthorized access or accidental or malicious attack outside the designed usage and to prevent disclosure of information.
System Qualities	Supportability	The ability of system to provide useful information for identifying and solving issues when it fails to work correctly.

	Testability	This is the measure of how easy it is to create test criteria for system and its components, execute the test to determine if the criteria are met.
User Qualities	Usability	This defines how well system meets user requirements by being easy to use, localize and globalize, intuitive and providing good user experience.

Venters et al. [24] define software quality attribute as ‘the degree to which a system, component or process meets a stakeholders needs or expectations, which means the only effective way of integrating sustainability into software systems is for it to be defined as a quality attribute like performance and usability though there has been little research in the area of defining sustainability as quality attribute, most work has focus on the energy consumption aspect of sustainability (environmental sustainability) and less has been done to cover all other aspect of sustainability due to the fact that many researchers feel the economical and social aspect of sustainability has gained more attention compared to the environmental aspect. This so far has proof counter productive because most companies are not interested in sustainability since they are not convince it adds value to their company which is reflected by the interview results from companies.

Nonfunctional requirements express desired qualities of the system to be developed and refer to both observable qualities and also to internal characteristics. McCall et al. [55] proposed a classification which differentiate the two level of quality attribute; quality factors and quality criteria. The Quality factors are external qualities that are measured indirectly such as interoperability, maintainability, portability, reliability and reusability. Quality criteria are measured either subjectively or objectively through combination of rating for each individual quality conditions that affects a given quality factor, and then a measure can be obtained to assess the extent to which that quality factor can be satisfied.

One of the biggest issues in defining sustainability as quality attribute is the fact that most proposed solutions lack clarity on implementation. Coral et al [56] define sustainability of a software product as the capacity of developing a software product in a sustainable manner , this is totally unclear and hard to quantify.

5.1.1 Sustainability View on Existing Software Quality Attribute

In order to be able to define sustainability as a quality attribute and also encourage software companies to add sustainability considerations during software development, it is important to have conscious thinking of sustainability on existing nonfunctional requirements and how it relates to the three main dimension of sustainability. The table below shows benefit of having sustainability thinking on existing nonfunctional requirements.

Table 15. Sustainability View on Existing Software Quality Attribute adopted from [57] and [58]

Attribute	**Sustainability View**		
	Economy	**Social**	**Environmental**
Reusability *The extent to which system component can be reused in another system*	Accelerates time to market	Aids the development of new product with less effort	Minimize environmental impact through less effort in system development
Modifiability *The extent to which changes can be applied to a system in cost effective manner.*	Reduce cost of development	Allows system to continuously evolve to meet societal requirements.	Reduce waste through less effort in developing and maintaining existing systems
Portability *The ability of system to function in different environment*	Increase system potential market and lifespan	Reduce user dependency on latest technology which reduce cost for technology adoption	Minimize e-waste by extending system lifespan

Supportability *The extend to which system can be easily configured and maintained*	Reduce cost of support which increase customer base	Increase in product usability due to vendors independence	Minimize resource usage for providing support (Transportation, calls, physical material)
Performance *The extent of systems responsiveness during execution of a task*	Improves Productivity	Minimize dependency on latest technology	Minimizes energy consumption through less computer usage time.
Dependability *The ability of system to function correctly at any time.*	Minimize cost of maintenance and support	Increase societal productivity	Indirect benefit: Minimize energy waste.
Usability *The ability of system to be user friendly (ease of use).*	Customer satisfaction, which lead to more customer and reduce support cost.	Make system more accessible which leads to digital inclusion eliminating barriers	Reduce waste of resources required for training (papers, books, rooms, energy)
Accessibility *The system ability to serve wider range of users irrespective of location, background, experience, and technology used.*	Increase potential market and profit	Leads to equal opportunity and multicultural awareness. Enables Technology to minorities, elderly and disabled people	Indirect benefit ?

Predictability Ability to accurately forecast or estimate effort and system state	Minimize cost and prevent budget overrun	Improves teams working condition (prevent long working time)	Optimize use of resources
Efficiency *The extent to which time is well used for the intended task.*	Maximize Product value	Minimize effort waste	Optimizes use of environmental resources
Projects Footprint *Environmental impact and Natural resources used during system development.*	Indirect benefit	Indirect benefit	Ensure efficient resources usage through reduction in fuel consumption and emissions, office space utilization

The table enables stakeholders in the software industry to understand the relationship between existing quality attributes and how it relates to sustainability.

Calero et al. [56] suggested that sustainability is related to some quality attributes and their sub-characteristics defined in the ISO/IEC 25010 quality model, they considered sustainability in two viewpoint : energy efficiency and perdurability (the degree to which a software product can be modified, adapted and reused in order to perform specified functions under specified conditions for a long period of time. Venters et. al. [59] defined software sustainability as a composite, non-functional requirement (quality attribute) which is ‘a measure of a systems extensibility, interoperability, maintainability, portability, reusability, scalability, and usability,’ this suggest that the concept of sustainability as a quality is strongly coupled with other quality attributes.

The above two definitions don’t cover all the three pillars of sustainability. Base on the above table 13, it is clear that sustainability is a composite quality attribute of software systems and in order to have a holistic definition of sustainability, it has to cover the three

main pillars. Thus, a new proposed definition of sustainability as quality attribute in software systems: "**The degree to which a system is usable, extensible, reusable, maintainable, portable, scalable, interoperable, efficient and provides reasonable balance for economy, society and environment based on the system context"** (working definition).
The definition uses context because sustainability can be viewed from different angles and perspective base on the type of software system being developed, in which sector is it applied and what kind of companies or individuals (users) are involve.

Kocak et al. [60] stated that Software development industry has started getting pressure from regulators to consider green software development. As a result, green attributes of software products are gaining importance as quality attributes. One of the biggest challenges facing companies is how to integrate sustainability into their system due to lack of consensus on what sustainability means in software systems and how it can be define as a quality factor of a system. The above definition of sustainability as a composite quality attributes provides basis for companies and ICT specialist to apply sustainability to software systems and remove the challenge of how to quantify sustainability in software systems.

6 GROUNDING SUSTAINABILITY IN THE FIELD OF SOFTWARE MEASUREMENT

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world [61] in such a way as to describe them according to clearly defined rules. The numerical outcome is called a ‘measurement’. This can be applied to both a software development process and a software product. Rini van Solingen and Egon Berghout in their book [31] state that Software measurement is the continuous process of defining, collecting, and analyzing data on the software development process and its products in order to understand and control the process and its products, and to supply meaningful information to improve that process and its products. Sandro Morasca[61] specified that software measurement is an emerging field of software engineering, since it may provide support for planning, controlling, and improving the software development process, as needed in any industrial development process.

Sustainability measurement is still a new area in the field of software measurement, Lami et al. [62] stated that there are few studies and suggestions about ‘what’ aspects of sustainability to measure and ‘how’ to do it. Calero et al. [63] highlighted that nowadays, sustainability is a key factor that should be considered in the software quality models, though there has less research channeled towards it. Seacord et al. [17] stated that planning and management of software sustainment is impaired by a lack of consistently applied, practical measures. Without these measures, it is impossible to determine the effect of efforts to improve sustainment practices.

6.1 Preconditions for Software Measurement

Magne Jørgensen [64] specified that dependent on how software quality is defined, software quality may be directly measured, indirectly measured or predicted. The below are the preconditions for different measurement types [64]:

1. Direct Measurement of software quality

    Empirical relational system of software quality is established which means there is a common understanding of “same quality” and better quality” that enable anyone to be able to distinguish or identify between “same”, “better” and “worse quality”.

    - Symbol or numerical figure with equivalent formal relations to the empirical quality is established using formal relations like "=" and ">" to show "good" and "bad" quality.
    - Measure mapping from the attribute of software quality to numbers or symbols is defined.
2. Indirect Measurement of software quality
    - The preconditions for measurement of the directly measured software attributes are met.
    - Complete empirical connection between the directly measured attributes and the indirectly measured software quality is established.
    - The connection is accurately translated into the formal relational system (through a formula)
3. Predictions of software quality are similar to indirect measurement. The main difference is that predictions do not require a complete empirical connection or an accurate translation into the formal relational system.

6.2 Basic Elements of software Measurement

In order to ensure that software is properly measured, there is need for five main elements, below are the main elements in software measurements according to William A. Florac [65]:

1. Goals and objectives are set relative to the software product and software management process.
2. Measurements are defined and selected to ascertain the degree to which the goals and objectives are being met.
3. A data collection process and recording mechanisms are defined and used.
4. Measurements and reports are part of a closed loop system that provides current (operational) and historical information to technical staff and management.
5. Data on post-software product life measurement is retained for analysis leading to improvements for future product and process management.

Software sustainability and its measurement is one of the key challenges in the development of software systems but in the industry and academia[66]. Base on the information from the chapter 2 (state of the art) in regards to software measurement and sustainability, the GREENSOFT Model mostly citied for sustainable software model and GQM as an approach to software metrics are identified as key elements in sustainability measurement for software systems.

6.3 GREENSOFT Model

The GREENSOFT Model is a comprehensive life cycle model for software products, sustainability criteria and metrics for software products, procedure models for different stakeholders, and recommendations for action, as well as tools that support stakeholders in developing, purchasing, supplying, and using software in a green and sustainable manner [67]. The issue with the adoption of this model is the fact that its has too many parts and models within it which makes its adoption complex for software engineers and companies because sustainability in software systems is still a new concept that requires an easy model as a starting point for the incorporation of sustainability thinking into software system development though it provides a good ground for creating and developing model or framework to support sustainability in software system development.

6.4 GQM Model

The Goal, Question, Metric (GQM) is a method to software metrics that supports business driven quality improvement very well, though it doesn't include sustainability but can support the adoption of sustainability in software development since it can incorporates business goals linked to metrics to measure the outcome of the software system being developed.

6.5 Proposed Sustainable Goal Question Metric (SGQM)

The proposed method will encourage the incorporation of sustainability thinking during software system development design and engineering processes from the management and software developers with a simplified way of measuring the sustainability of the system based of quality attribute.

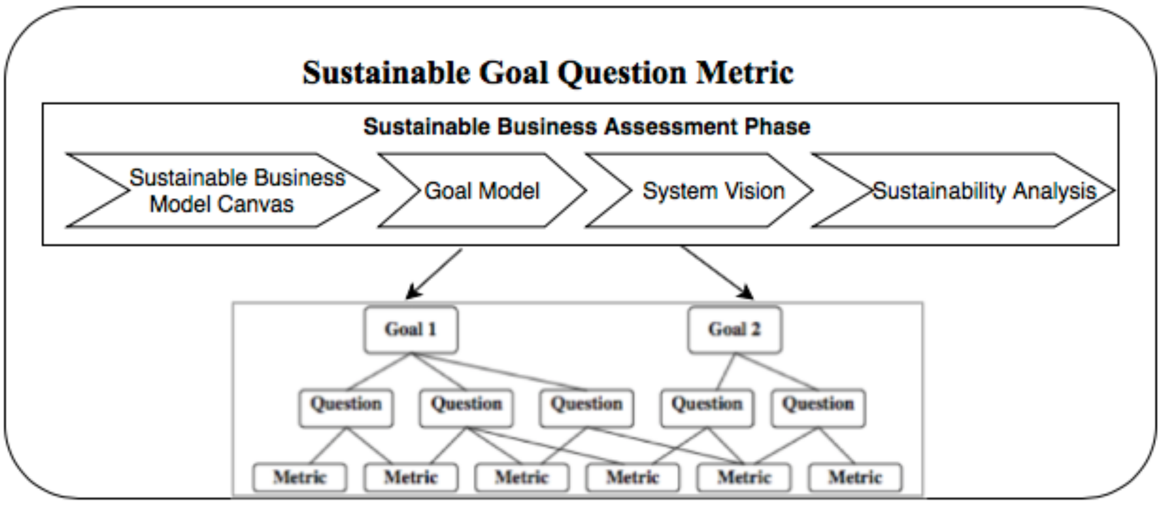


Figure 13. Sustainable Goal Question Metric (SGQM)

The proposed sustainable goal question metric (SGQM) involves two phases; the Sustainable Business Assessment phase and Goal Question Metric Phase.

Sustainable Business Assessment Phase helps facilitate the adoption of sustainability thinking during product innovation and design, this enables easier integration and measurements of sustainability in software systems. It involves the following:

- **Sustainable Business Model Canvas**: The Business Model Canvas incorporates sustainability considerations during business model design. It allows users to describe, design, challenge, invent, and pivot their business model in a sustainable manner [68].
- **Goal Model:** It shows comprehensive and holistic goals of the organization or company in relation to the product under development from the economic, social and environmental dimensions represented in business goal, usage goal and system goal[69].
- **System Vision:** It provides an overview of the whole system and how it interacts with different external components and its potential users based on the agreement of all stakeholders[40].
- **Sustainability Analysis:** Sustainability analysis diagram describe the system from sustainability perspective by considering the sustainability purpose of the system, impact the system have on environment as well as sustainability goal and constraint of the system [41].

- The **Goal Question Metric (GQM)** phase ensure that the companies goals and objectives are considered during system design with sustainability thinking and provides the tracing and measurement of companies goals base on the result from the sustainable business assessment model phase. Examples of metrics that can be used in the phase are Backlog Management Index (BMI), Rework Metric, Defect Density, Energy Efficiency, Gateway Metric and project footprint.

The three biggest issues in adoption of sustainability in software system development as seen in this thesis research are first, the understanding of what sustainability means in software system because different research suggest different definition just like the parable of the blind men and the elephant where six blind men touch the elephant and based on their individual experience suggested it was a wall, spear, snake, tree [66], there is no agreed definition. Second, the proposed solutions are either too complex or focus mainly on the environmental dimension of sustainability (energy efficiency). Third total ignore of management concerns or goals in the adoption of sustainability in software system design and development. In order to solve the aforementioned issues, it is important to first identify what is sustainability as quality attribute? This research work propose the definition of sustainability as quality attribute in software systems as **"a composite quality attribute, which is the degree to which a system is extensible, reusable, maintainable, portable, scalable, interoperable, efficient and provides reasonable balance for economy, society and environment based on the system context"** (working definition). Second, the proposed SGQM provides a simplified solution as a stating point for the aforementioned issues by integrating sustainability into software systems design and development life cycle with the consideration of management goals through sustainable business assessment phase and measurement through GQM.

The SGQM also supports and influences the three aspects of software quality namely functional, structural and process [70] that can support the definition of sustainability as a quality attribute in software systems.

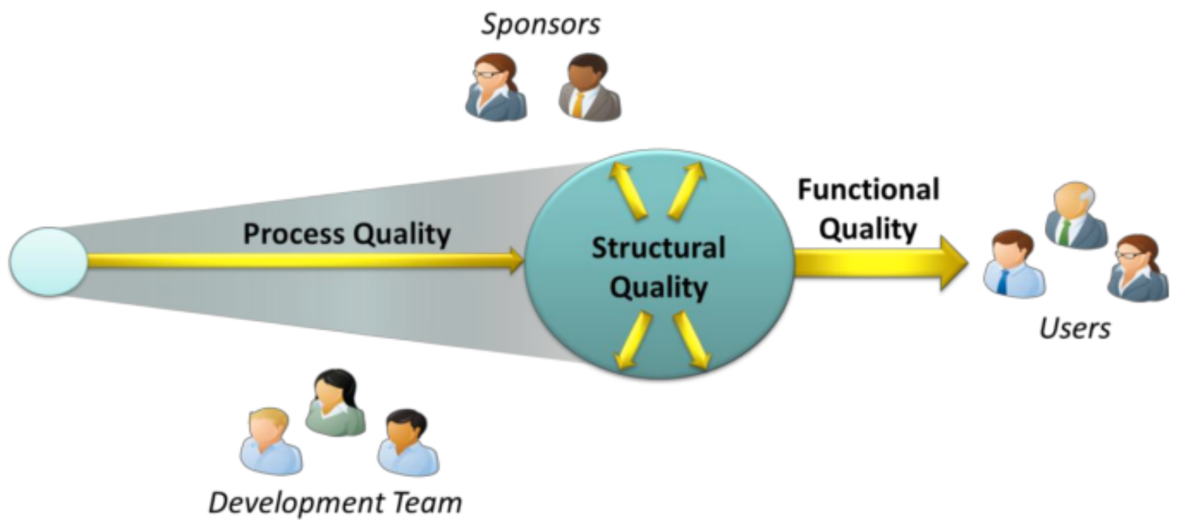


Figure 14. Three Aspect of Software Quality adopted from [70]

The functional quality means that software correctly performs it function based on specified requirements either from project sponsors or intended users [70]. SGQM supports sustainability thinking during requirement gathering which in turn help in design of software system with sustainable functional quality.

The second aspect of software quality is the structural quality means that the code is well structured to ensure code testability, maintainability, understandability, efficiency and security[70]. Sustainability analysis result in SGQM provides a concrete foundation of ensuring the structural quality of software quality incorporate sustainability concerns into software coding which control the behavior, operation of software system and also has an effect on the first, second and third order impacts of software systems on the environment.

Process quality, which at times gets less attention as compared to the functional and structural quality, is very critical and important because it covers the development process which significantly affects the value received by users, development teams, and sponsors [70]. The SGQM process encourages sustainability thinking during software development process to ensure that the process quality produces a software system that is sustainable.

Finally, the procedure and steps in SGQM inspires the three major stakeholders (Sponsors, Development team and Users) to consider sustainability during system development. An example of the application of SGQM in software system development through a use case will demonstrates the advantages of using SGQM in the next section below.

The next step to define sustainability as a quality attribute will require a lot of testing of SGQM using quality attribute scenarios as a means of characterizing sustainable quality attributes.

In order to specify sustainability quality attribute requirement, the use of quality attribute scenarios can serve as a starting pointing which consist of the following [71] [72]:

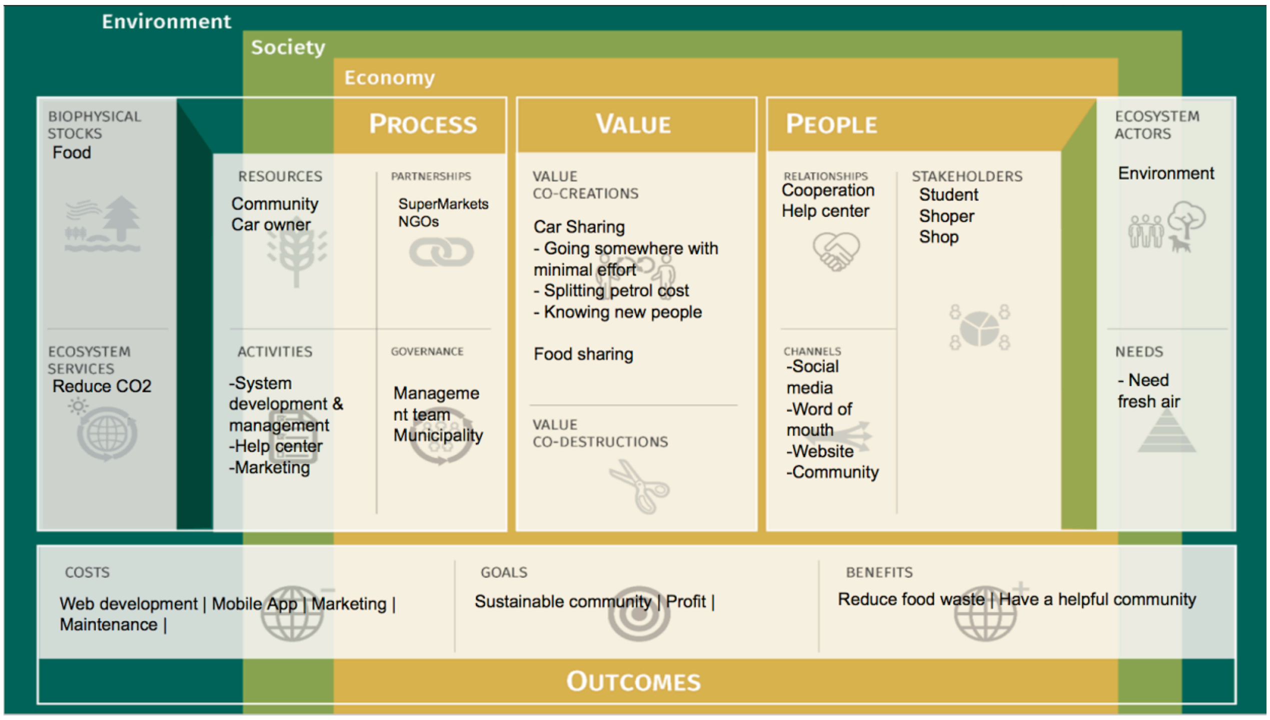
- Source of stimulus: It is an entity, which can be human or computer system that generated the stimulus.
- Stimulus: The stimulus is a condition that needs to be considered when it arrives at a system.
- Environment: The stimulus occurs within certain conditions. The system may be in an overload condition or may be running when the stimulus occurs, or some other condition may be true.
- Artifact: Some artifact is stimulated. This may be the whole system or some pieces of it.
- Response: The response is the activity undertaken after the arrival of the stimulus.
- Response measure: When the response occurs, it should be measurable in some fashion so that the requirement can be tested

6.5.1 Sample Use Case

The sample use case provides an example of how this Model works using a class project where my team proposed development of car sharing system called ***ShareVoyage***; an online web platform designed for the students within a certain community to share ride and food. The system offers users to share their car with nearby people who want to go for group shopping and also share unused foods. The use of this Model involves two phases, the sustainable business assessment model phase and the Goal Question Metric (GQM) phase.

6.5.1.1 Sustainable Business Assessment Model Phase

The first stage involves modeling the team idea using business canvas with sustainable business model canvas to show the impact of the business idea from the economic, social and environmental pillar of sustainability without compromising the ability of the business to function.



BIOPHYSICAL STOCKS
Food

ECOSYSTEM SERVICES
Reduce CO2

RESOURCES
Community
Car owner

PARTNERSHIPS
SuperMarkets
NGOs

ACTIVITIES
-System development & management
-Help center
-Marketing

GOVERNANCE
Manageme nt team
Municipality

VALUE CO-CREATIONS
Car Sharing
- Going somewhere with minimal effort
- Splitting petrol cost
- Knowing new people

Food sharing

VALUE CO-DESTRUCTIONS

RELATIONSHIPS
Cooperation
Help center

STAKEHOLDERS
Student
Shoper
Shop

CHANNELS
-Social media
-Word of mouth
-Website
-Community

ECOSYSTEM ACTORS
Environment

NEEDS
- Need fresh air

COSTS
Web development | Mobile App | Marketing | Maintenance |

GOALS
Sustainable community | Profit |

BENEFITS
Reduce food waste | Have a helpful community

Figure 15. Sustainability Business Canvas adopted from [73]

The second stage involves the goal model that shows the business goal, usage goal and system goal from five dimension economic, environment, social, individual and technical of the system to be developed.

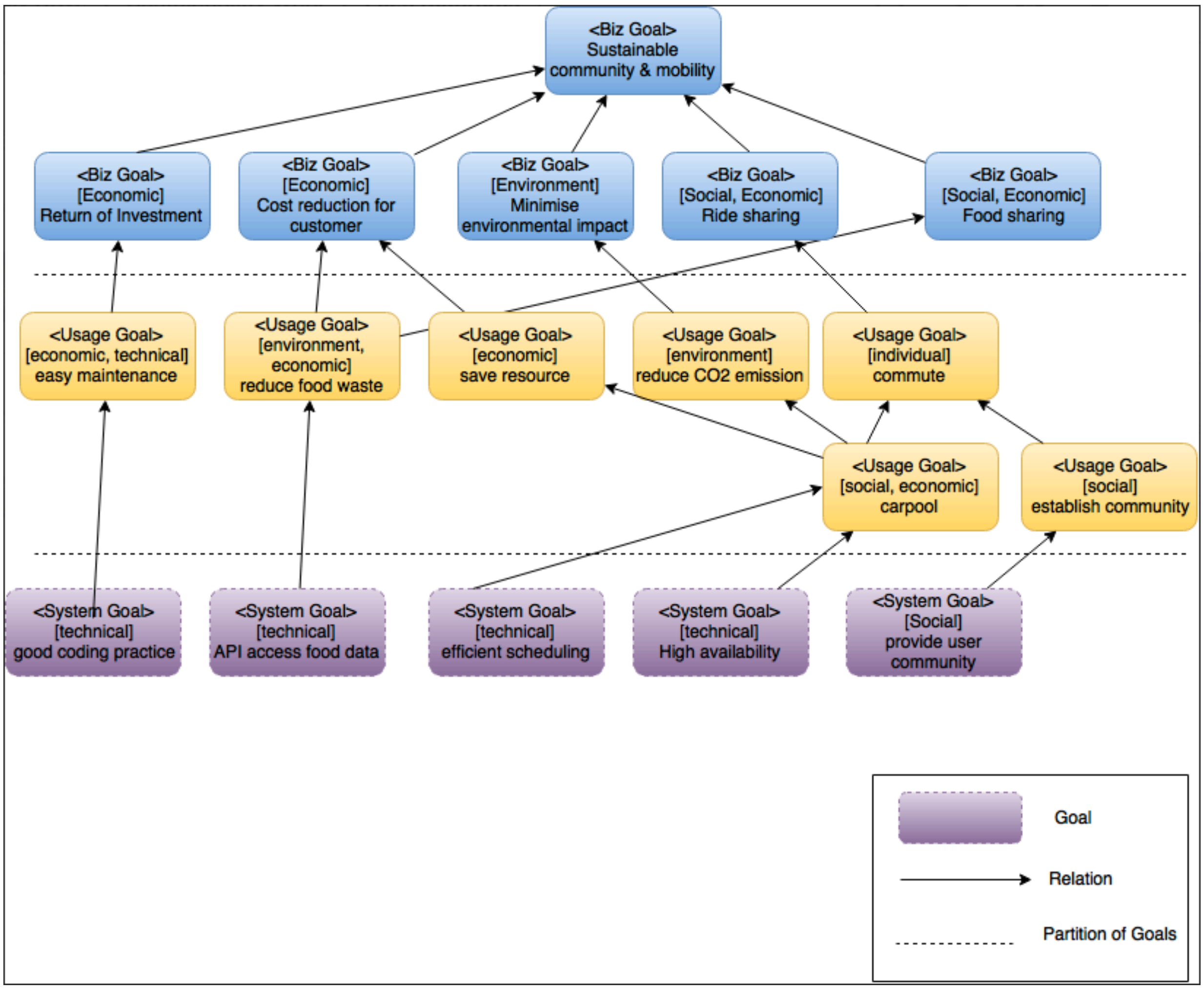

<Biz Goal> Sustainable community & mobility
<Biz Goal> [Economic] Return of Investment
<Biz Goal> [Economic] Cost reduction for customer
<Biz Goal> [Environment] Minimise environmental impact
<Biz Goal> [Social, Economic] Ride sharing
<Biz Goal> [Social, Economic] Food sharing
<Usage Goal> [economic, technical] easy maintenance
<Usage Goal> [environment, economic] reduce food waste
<Usage Goal> [economic] save resource
<Usage Goal> [environment] reduce CO2 emission
<Usage Goal> [individual] commute
<Usage Goal> [social, economic] carpool
<Usage Goal> [social] establish community
<System Goal> [technical] good coding practice
<System Goal> [technical] API access food data
<System Goal> [technical] efficient scheduling
<System Goal> [technical] High availability
<System Goal> [Social] provide user community
Goal
Relation
Partition of Goals

Figure 16. The Goal model adopted from [73]

The third stage presented the system vision in a pictorial format for easy analysis of how different stakeholders interact or relate to each other presented in the business context, system context and operational context.

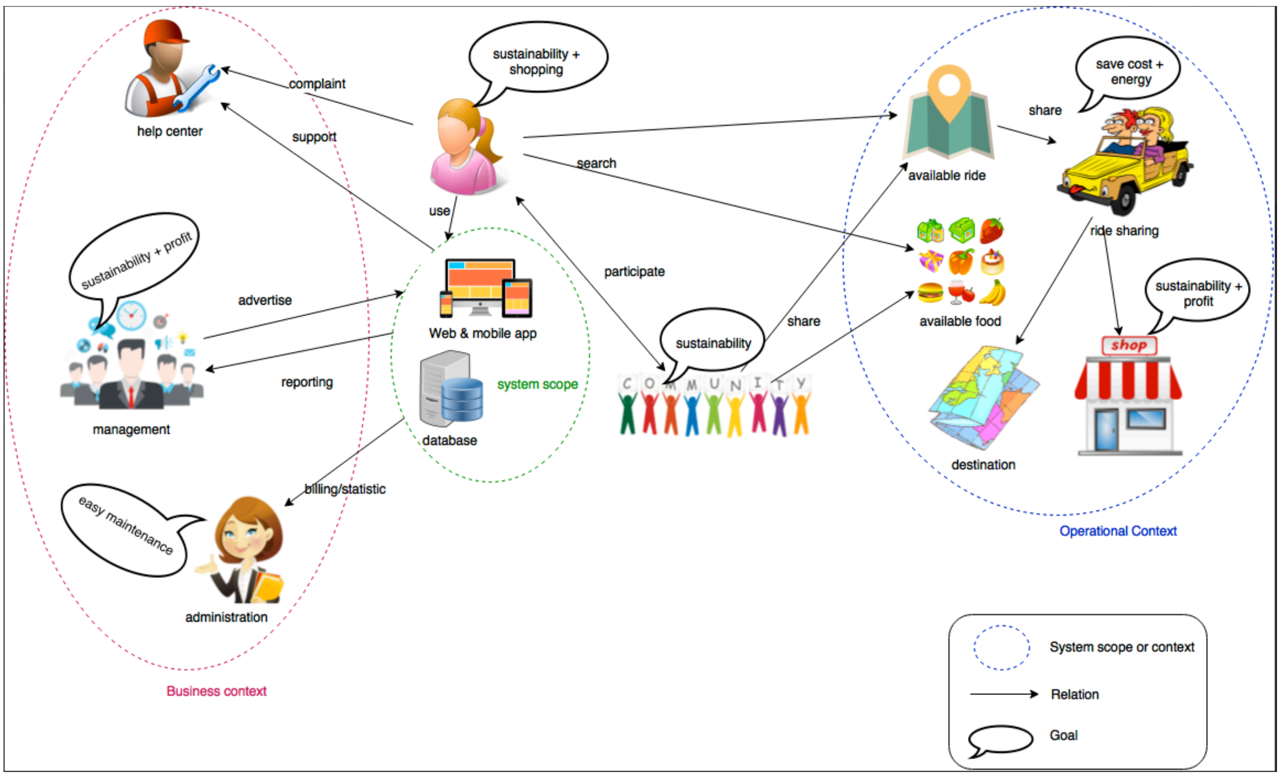


Figure 17. System Vision adopted from [73]

The fourth stage is the sustainability analysis which describe the system from sustainability perspective by considering the sustainability purpose of the system, impact the system have on environment as well as sustainability goal and constraint of the system using the five dimensions of sustainability such as technical, individual, economic, environmental and finally social.

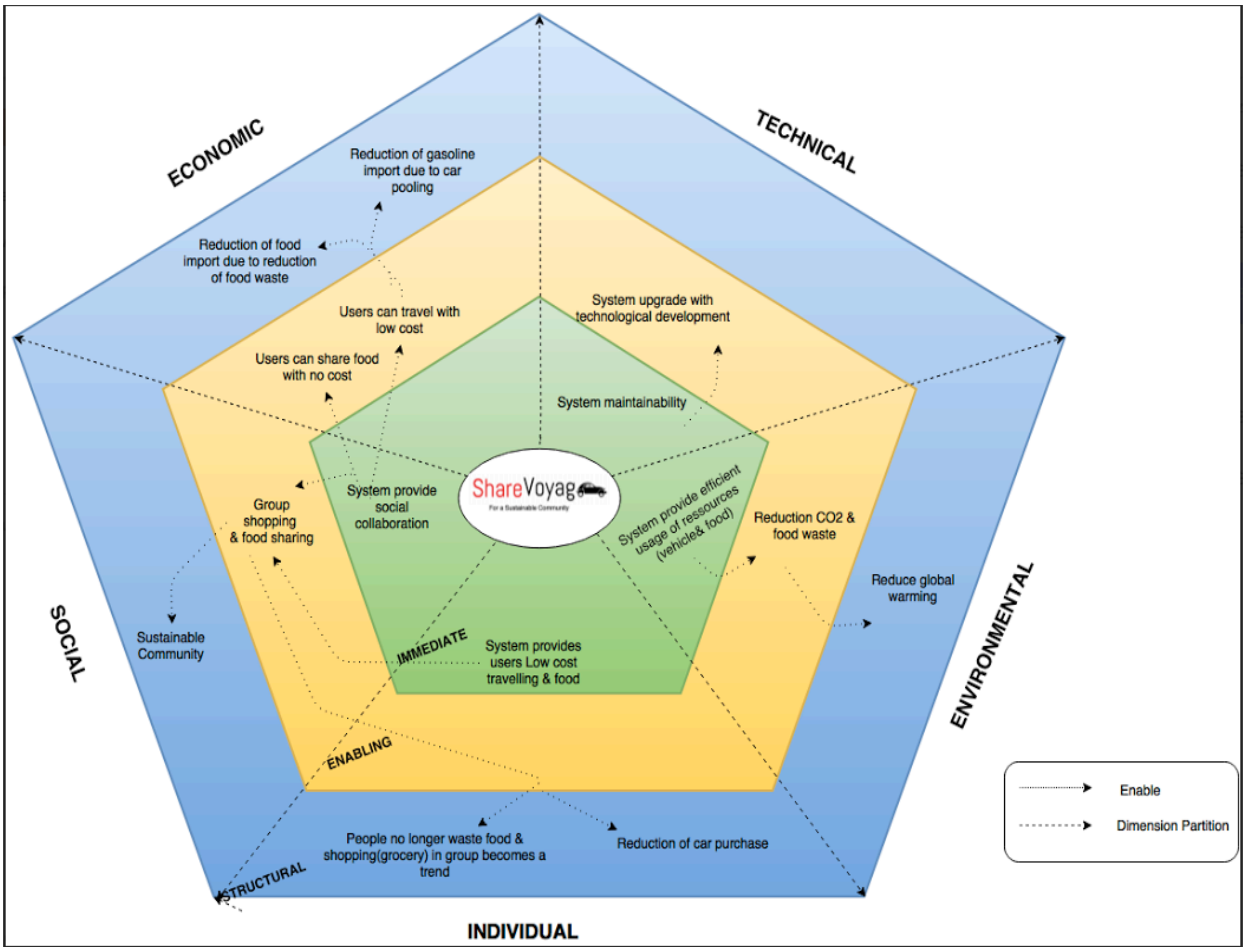

ECONOMIC
TECHNICAL
SOCIAL
ENVIRONMENTAL
INDIVIDUAL
Reduction of gasoline import due to car pooling
Reduction of food import due to reduction of food waste
Users can travel with low cost
Users can share food with no cost
System upgrade with technological development
System maintainability
ShareVoyage
System provide social collaboration
System provide efficient usage of ressources (vehicle& food)
Reduction CO2 & food waste
Reduce global warming
Group shopping & food sharing
Sustainable Community
System provides users Low cost travelling & food
IMMEDIATE
ENABLING
STRUCTURAL
People no longer waste food & shopping(grocery) in group becomes a trend
Reduction of car purchase
Enable
Dimension Partition

Figure 18. Sustainability Analysis adopted from [73]

6.5.1.2 Goal Question Metric Phase

1. The first phase in involves planning on how to measure the software system based on the management goals.

Table 16. Management Goals

Project Name: *ShareVoyage*	
Description:	Car sharing web application.
Goals:	
Reduce Cost of development and maintenance	
Improve customer satisfaction	
Improve product quality	
Ensure product sustainability	

2. The second stage involves setting questions that will be used in assessment of each goal. These questions are usually from the technical team translating the management goals into a set of simple understandable technical questions.

Table 17. Set Questions

Project Name: *ShareVoyage*	
Description:	Car sharing web application.
Goals	**Questions**
Reduce Cost of development and maintenance	What is the backlog Management Index (BMI)?
Improve customer satisfaction	Can users complete task?
Good product quality	What is the defect density? What is the amount of rework?
Ensure product sustainability	Does the actual cost outweigh the budgeted cost? Are the project teams happy? What is the project footprint?

3. The Third stage specifies the metrics that would be used in assessing all the management goals and also provides information on how to translate the results base on management regulations on acceptable level for all measurement results.

Table 18. Metric Worksheet

Category	**Question**	**Metric**	**Positive Marks**
Technical	What is the Backlog Management Index (BMI)?	$\text{BIMI=}\dfrac{\text{Number of problems closed during the month}}{\text{Number of problems arrivals during the month}}\text{X 100}$	0 or 100
	What is the amount of rework?	Rework Metric (Total Number of function modified)	0
Economy	What is the BMI?	$\text{BIMI=}\dfrac{\text{Number of problems closed during the month}}{\text{Number of problems arrivals during the month}}\text{X 100}$	0
	What is the defect density?	Defect Density= Total defects/Size	$\leq$10.46
	Does the actual project cost outweigh budgeted cost?	Budgeted Capital - Total Capital Spent	+ Number
Environment	What is the BMI?	$\text{BIMI=}\dfrac{\text{Number of problems closed during the month}}{\text{Number of problems arrivals during the month}}\text{X 100}$	0 or 100
	What is the defect density?	Defect Density= Total defects/Size	$\leq$ 10.46
	What is the software energy efficiency?	Useful work done/Used Energy	
Individual	Can users successfully complete task?	Gateway metric (1=Task success and 0= Task failure)	7
	What is the defect density?	Defect Density= Total defects/Size	$\leq$ 10.46
Social	Can users successfully complete task?	Gateway metric (1=Task success and 0= Task failure)	7
	What is the defect density?	Defect Density= Total defects/Size	$\leq$ 10.46
	Are the project teams happy?	Budgeted hours - Total working hours	+ Number

4. The fourth stage is the data collection and interpretation stage which involves using the metric to evaluate the system based on the questions associated to each metric.

Table 19. Data Collection and Interpretation

Category	**Question**	**Metric**	**Result**	**Positive Marks**
Technical	What is the Backlog Management Index (BMI)?	$BIMI=\frac{\text{Number of problems closed during the month}}{\text{Number of problems arrivals during the month}} X\ 100$	100	0 or 100
	What is the amount of rework?	Rework Metric (Total Number of function modified)	0	0
Economy	What is the BMI?	$BIMI=\frac{\text{Number of problems closed during the month}}{\text{Number of problems arrivals during the month}} X\ 100$	100	0
	What is the defect density?	Defect Density= Total defects/Size	10.4	$\leq$ 10.46
	Does the actual project cost outweigh budgeted cost?	Budgeted Capital - Total Capital Spent	+ 1000 Euros	+ Number
Environment	What is the BMI?	$BIMI=\frac{\text{Number of problems closed during the month}}{\text{Number of problems arrivals during the month}} X\ 100$	100	0 or 100
	What is the defect density?	Defect Density= Total defects/Size	10.4	$\leq$ 10.46
	What is the software energy efficiency?	Useful work done/Used Energy		
Individual	Can users successfully complete task?	Gateway metric (1=Task success and 0= Task failure)	7	7
	What is the defect density?	Defect Density= Total defects/Size	10.4	$\leq$ 10.46

Social	Can users successfully complete task?	Gateway metric (1=Task success and 0= Task failure)	7	7
	What is the defect density?	Defect Density= Total defects/Size	10.4	≤ 10.46
	Are the project teams happy?	Budgeted hours - Total working hours	+ 40 hours	+ Number

The Above use case demonstrates how Sustainable Goal Question Metric (SGQM) facilitates sustainability thinking in a company or organization management process and provides a model that can be use to evaluate the outcome (system) using quantifiable values to ensure the system meets their entire requirements. SQGM also ease the problem of adopting sustainability during software development because it ensures that that the companies from the beginning consider sustainability through the Sustainable Business Assessment Model Phase.

6.6 Comparison between Models

Table 20. Comparison Between Models

Evaluation Areas	**GREENSOFT Model**	**GQM**	**SGQM**
Sustainability integration in software systems	Yes	No	Yes
Sustainability measurement of software system	Yes but no practical example found	Yes	Yes
Integration of business goals to software system	No	Yes	Yes
Sustainability Integration in Business Culture	No	No	Yes
Support sustainability pillars (economy, social, environment)	Environment	No	Yes
Ease of adoption	Complicated	Easy	Easy
Direct or Indirect awareness of sustainability to users and companies	Mostly academicians	Not Used	Target Companies, Users, Academicians

The above table shows SGQM incorporates management concerns in software systems and provides measurement in an easy way compared to GREENSOFT Model and GQM.

7 SUMMARY AND FUTURE WORK

The research on how to define and measure sustainability in software systems involves a broad study (start of the art), interviews and review of the interview results to analysis and identify the challenges of defining sustainability as a quality attribute in software systems. Sustainability in software systems is a concept that has multi-dimensional angles with different researchers describing it from their own perspective base on their area of specialization. This has led to difficult in having a consensus agreement of what sustainability really means and how it relates to software systems.

Analysis of the interview results shows that there is still lack of complete understanding about software sustainability in companies and also in academia as seen in Table. 7 and 8 where the participants responded that sustainability is only about saving or protecting the environment/planet and its about energy efficiency but in reality sustainability covers economy, environment and society. Table 9 and 10 shows that companies and ICT specialist still don't know how to implement and measure sustainability in software systems. Companies still don't know the value of sustainability as shown in Table 11.

In order to tackle all these challenges, the first step is to quantify sustainability as quality attribute in software systems. It will provide clear and measureable definitions that can be applied in ICT industry which will lead to easier understanding and adoption of sustainability during software development and also encourage software companies to think in a sustainable manner. Table 13 shows the impact of quantifying sustainability can have on existing quality attributes as researchers can start looking at news ways of defining them with the consideration of sustainability.

The proposed solution Sustainability Goal Question Metric (SGQM) will assist software companies and ICT specialist during software innovation and idea development through the sustainability business assessment phase to imbibe sustainability thinking into their management processes and using GQM provide a means to measure their software product in a way that incorporate management concerns into different aspects of the software systems. It can also serve as a starting point for defining sustainability as a quality attribute because it supports and influences the three aspect of software quality as stated in chapter 6 and also the integration of sustainability into software development life cycle. The sample use case demonstrates how SGQM work and can be applied in software companies.

7.1 Future Work

The current proposal is a starting point that requires further study to explore more areas so as to have an improve definition of sustainability as a quality attribute and its measurement. The results can lead to development of an SGQM measurement system to assist companies in incorporating sustainability into their management processes.

Testing SQGM in diverse companies on different software systems will provide good feedback on how to improve the measurement model to ensure better adoption. One of the major challenges during the current research work was finding sustainability metrics to evaluate software systems. This can lead to new research on defining new sustainability metrics in software engineering domain.

REFERENCES

[1] P. Berander, L. Damm, J. Eriksson, T. Gorschek, K. Henningsson, P. Jönsson, S. Kågström, D. Milicic, F. Mårtensson, K. Rönkkö, P. Tomaszewski, L. Lundberg, M. Mattsson, and C. Wohlin, "Software quality attributes and trade-offs," no. June, pp. 1–100, 2005.

[2] Microsoft MSDN, "Chapter 16: quality attributes," vol. 658094, pp. 1–9, 2014.

[3] B. Penzenstadler, "What does Sustainability mean in and for Software Engineering ?"

[4] F. Albertao, J. Xiao, C. Tian, Y. Lu, K. Q. Zhang, and C. Liu, "Measuring the Sustainability Performance of Software Projects," *2010 IEEE 7th Int. Conf. E-bus. Eng.*, pp. 369–373, 2010.

[5] M. Dick and S. Naumann, "Enhancing software engineering processes towards sustainable software product design," *24th Int. Conf. Informatics Environ. Prot. (EnviroInfo 2010)*, vol. 2010, pp. 706–715, 2010.

[6] H. unimelb. edu. au/accessibility/users/developmen. Computer Science, "Software Development Lifecycle Software Development Lifecycle," 2011.

[7] E. I. T. O. ·. EITO, "The impact of ICT on sustainable development x x x," pp. 250–283, 2002.

[8] H. J. Berkhout Frans, "Impacts of Information and Communication Technologies on Environmental Sustainability : speculations and evidence," vol. 5.

[9] T. U. M. https://sustainability. wiki. tum. de/Sustainable+Software+Developmen. Wiki, "Sustainable Software Engineering," pp. 366–367, 2010.

[10] K. Tate, "Sustainable Software Development book," pp. 1–12.

[11] B. Penzenstadler, A. Raturi, D. Richardson, and B. Tomlinson, "Safety, security, now sustainability: The nonfunctional requirement for the 21st century," *IEEE Softw.*, vol. 31, no. 3, pp. 40–47, 2014.

[12] S. Naumann, M. Dick, E. Kern, and T. Johann, "The GREENSOFT Model: A reference model for green and sustainable software and its engineering," *Sustain. Comput. Informatics Syst.*, vol. 1, no. 4, pp. 294–304, 2011.

[13] H. sustainability. com/sustainabilit. SustainAbility, "Sustainability: Can our society endure?," 2010. .

[14] C. Becker, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, and C. C. Venters, “Sustainability Design and Software: The Karlskrona Manifesto,” *Proc. - Int. Conf. Softw. Eng.*, vol. 2, pp. 467–476, 2015.

[15] A. . Fallis, *PEAK EVERYTHING waking up to the century of declines*, vol. 53, no. 9. 2013.

[16] J. A. Tainter, “Social complexity and sustainability,” *Ecol. Complex.*, vol. 3, no. 2, pp. 91–103, 2006.

[17] L. W. Robert Seacord, Joseph Elm, Wolf Goethert, Grace Lewis, Dan Plakosh, John Robert, “Measuring Software Sustainability,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[18] I. Gorton, *Essential Software Architecture full book*, vol. 1. 2015.

[19] S. E. S. C. IEEE, *IEEE Standard for a Software Quality Metrics Methodology*. 1998.

[20] A. Raturi, B. Penzenstadler, B. Tomlinson, and D. Richardson, “Developing a sustainability non-functional requirements framework,” *Proc. 3rd Int. Work. Green Sustain. Softw. - GREENS 2014*, pp. 1–8, 2014.

[21] Z. Durdik, B. Klatt, H. Koziolek, K. Krogmann, J. Stammel, and R. Weiss, “Sustainability guidelines for long-living software systems,” *IEEE Int. Conf. Softw. Maintenance, ICSM*, pp. 517–526, 2012.

[22] K. Erdélyi, “Special factors of development of green software supporting eco sustainability,” *SISY 2013 - IEEE 11th Int. Symp. Intell. Syst. Informatics, Proc.*, pp. 337–340, 2013.

[23] T. Johann, M. Dick, E. Kern, and S. Naumann, “Sustainable development, sustainable software, and sustainable software engineering: An integrated approach,” *2011 Int. Symp. Humanit. Sci. Eng. Res.*, pp. 34–39, 2011.

[24] C. C. Venters, C. Jay, L. M. S. Lau, M. K. Griffiths, V. Holmes, R. R. Ward, J. Austin, C. E. Dibsdale, and J. Xu, “Software sustainability: The modern tower of babel,” *CEUR Workshop Proc.*, vol. 1216, pp. 7–12, 2014.

[25] N. Amsel, Z. Ibrahim, A. Malik, and B. Tomlinson, “Toward sustainable software engineering: NIER track,” *2011 33rd Int. Conf. Softw. Eng.*, pp. 976–979, 2011.

[26] B. Penzenstadler, “Supporting Sustainability Aspects in Software Engineering,” *3rd Int. Conf. Comput. Sustain.*, pp. 1–4, 2013.

[27] germain saval Martin, mahaux, patrick heymans, “Requirements Engineering: Foundation for Software Quality,” *Requir. Eng. Found. Softw. Qual.*, vol. 4542, no.

January, pp. 247 – 261, 2007.

[28] B. Penzenstadler and H. Femmer, “A generic model for sustainability with process- and product-specific instances,” *GIBSE 2013 - Proc. 2013 Work. Green Softw. Eng. Green by Softw. Eng.*, no. June 2015, pp. 3–7, 2013.

[29] T. Johann, M. Dick, S. Naumann, and E. Kern, “How to measure energy-efficiency of software: Metrics and measurement results,” *2012 1st Int. Work. Green Sustain. Software, GREENS 2012 - Proc.*, pp. 51–54, 2012.

[30] P. Bozzelli, Q. Gu, and P. Lago, “A systematic literature review on green software metrics,” *Sis.Uta.Fi*, 2013.

[31] R. Solingen and E. Berghout, “The goal/question/metric method,” *A Pract. Guid. Qual. Improv. Softw. Dev. New York, McCraw-Hill Publ.*, p. 216, 1999.

[32] H. D. R. Basili, Victor, “GQM Metric Paradigm.pdf.” .

[33] C. Differding, Hoisl, Barbara, and C. M. Lott, “Technology Package for the Goal Question Metric Paradigm,” *Fachbereich Inform. Interner Bericht*, no. 281/1996, p. 27, 1996.

[34] B. Ahmed, “Universal Design Principles,” 1997.

[35] V. R. Basili and J. Calvo-villagran, “The Goal Question Metric Approach.”

[36] Https://canvanizer.com/new/business-model-canvas, “The Business Model Canvas.”

[37] Http://www.slideshare.net/peterjones/ocadu-research-flourishing-business, “Flourishing Business model Canvas.”

[38] B. H. C. Cheng, “Goal Modeling,” 2006.

[39] E. Kavakli and P. Loucopoulos, “Goal Modelling in Requirements Engineering,” *Inf. Model. Methods Methodol. Adv. Top. Database Res.*, pp. 1–27, 2005.

[40] B. Penzenstadler, “System Vision Requirements Engineering for Sustainability Timeline.”

[41] B. Penzenstadler, “Requirements Engineering for Sustainability - Sustainability Analysis Timeline.”

[42] C. S. Hevner Alan, *Design Research in Information Systems*. .

[43] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–78, 2007.

[44] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information Systems Research,” *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.

[45] Q. R. M. A. D. C. F. Guide, M. 1 Q. R. M. Overview, and O. F. A. Y. H. T. I. TIONAL, “Qualitative Research Methods Overview,” p. 12, 2015.

[46] N. Brikci and J. Green, “A Guide to Using Qualitative Research Methodology,” pp. 1–36, 2007.

[47] B. Hancock, “An Introduction to Qualitative Research Au t hors,” *Qual. Res.*, vol. 4th, p. 504, 2006.

[48] B. G. Glaser, “What is Grounded Theory?,” *Grounded Theory Online; Support. GT Res.*, 2015.

[49] C. B. Seaman, “Qualitative methods in empirical studies of software engineering,” *IEEE Trans. Softw. Eng.*, vol. 25, no. 4, pp. 557–572, 1999.

[50] W. M. K. http://www. socialresearchmethods. net/kb/qualapp. ph. Trochim, “Qualitative Approaches.” .

[51] K. Locke, “Grounded Theory in Management Research,” *Account. Bus. Res.*, vol. 32, no. 1, pp. 57–58, 2002.

[52] M. El Hussein, S. Hirst, V. Salyers, and J. Osuji, “Using Grounded Theory as a Method of Inquiry: Advantages and Disadvantages.,” *Qual. Rep.*, vol. 19, no. 27, pp. 1–15, 2014.

[53] R. Bunch, “Simplifying complexity. The 8 Sustainability Challenges for Canadian Business in 2014,” *NBS’s 2014 Challenges Rep.*, 2014.

[54] T. Bansal, “canadian business sustainability priorities 2011.”

[55] J. a. Mccall, P. K. Richards, and G. F. Walters, “Factors in software quality: Concept and Definitions of Software Quality,” vol. I, no. November, p. 188, 1977.

[56] C. Calero, M. A. Moraga, and M. F. Bertoa, “Towards a Software Product Sustainability Model,” *J. Sustain.*, vol. 25010, p. 4, 2013.

[57] M. L. Sanz, “Sustainability Modelling,” 2011.

[58] F. Albertao, “SUSTAINABLE SOFTWARE ENGINEERING,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[59] C. C. Venters, L. Lau, M. K. Griffiths, V. Holmes, R. R. Ward, C. Jay, C. E. Dibsdale, and J. Xu, “The Blind Men and the Elephant Towards an Empirical Evaluation Framework for Software Sustainability,” vol. 2, no. 1, pp. 1–6, 2014.

[60] S. A. . Koçak, G. I. . Alptekin, and A. B. . Bener, “Evaluation of software product quality attributes and environmental attributes using ANP decision framework,” *CEUR Workshop Proc.*, vol. 1216, pp. 37–44, 2014.

[61] S. MORASCA, “Software measurement,” 1996.

[62] G. Lami and L. Buglione, “Measuring software sustainability from a process-centric perspective,” *Proc. 2012 Jt. Conf. 22nd Int. Work. Softw. Meas. 2012 7th Int. Conf. Softw. Process Prod. Meas. IWSM-MENSURA 2012*, pp. 53–59, 2012.

[63] C. Calero, M. F. Bertoa, and M. A. Moraga, “A systematic literature review for software sustainability measures,” *2013 2nd Int. Work. Green Sustain. Software, GREENS 2013 - Proc.*, pp. 46–53, 2013.

[64] M. Jørgensen, “Software quality measurement,” vol. 30, pp. 907–912, 1999.

[65] Q. Subgroup, S. Metrics, D. Working, S. Process, and M. Project, “with the Quality Subgroup of the Software Metrics Definition Working Group and the Software Process Measurement Project Team.”

[66] “Defining ’ and ' Measuring ' Software‘ Sustainability :’ Towards ’ An ' Empirical ' Framework ' for ' Evaluation ' at ' the ' Architectural ' Level ' Abstract ' Keywords ' Introduction ' Software‘ Sustainability ?’”

[67] S. Naumann, M. Dick, E. Kern, and T. Johann, “The GREENSOFT Model: A reference model for green and sustainable software and its engineering,” *Sustain. Comput. Informatics Syst.*, vol. 1, no. 4, pp. 294–304, 2011.

[68] B. Penzenstadler, “Sustainability Business Canvass Requirements Engineering for Sustainability Timeline.”

[69] B. Penzenstadler, “Goal Model Requirements Engineering for Sustainability Timeline.”

[70] D. Chappell, “THE THREE ASPECTS OF SOFTWARE QUALITY : FUNCTIONAL , STRUCTURAL , AND PROCESS Sponsored by Microsoft Corporation,” *David chappel Assoc.*, vol. 1.0, 2012.

[71] F. Bachmann and M. Klein, “Chapter 4 . Understanding Quality Attributes,” *Quality*, pp. 1–30, 2011.

[72] R. K. Len Bass, Paul Clements, “Software Architecture in Practice, 3rd Edition,” 2013.

[73] R. Oyedeji shola, Chandara, Mustaqim, “Technical Report on Requirements Engineering for Sustainability.”