Sustainability Quantification in Requirements Informing Design

Shola Oyedeji
Department of Software Engineering
Lappeenranta University of Technology
Lappeenranta, Finland
shola.oyedeji@lut.fi

Ahmed Seffah
Department of Software Engineering
Lappeenranta University of Technology
Lappeenranta, Finland
ahmed.seffah@lut.fi

Birgit Penzenstadler
Department of Computer Engineering and Computer Science
California State University Long Beach
Long Beach, California, USA
birgit.penzenstadler@csulb.edu

Abstract— **Sustainability has been defined with different perceptions and from diverse dimensions making it an ambiguous concept to objectively engineer and integrate into software development lifecycle. Although a large body of knowledge already exists on what sustainability is and isn't, little research has explored how to quantify sustainability. How can the definitions and perceptions of sustainability from software engineering and other fields be turned into requirements, effective measures that quantify sustainability and most importantly can inform a "sustainability by design" approach? What are the measures and measurement scale of sustainability? Our long-term research goal is to answer such questions and similar ones. In this position paper, we summarize our investigations and pave the road for a theoretical ground of sustainability quantification in software development and measurement. The goal is to foster research and standardization initiatives on sustainability as a quality attribute and sustainability by design.**

Keywords: *Software Sustainability, Sustainability Requirements, Software Measurement, Software Development, Sustainability Metrics, Software Design*

I. INTRODUCTION

In a broad sense, sustainability is "the capacity to endure" [1]. In software engineering, sustainability has been introduced from different dimensions with diverse perceptions and definitions. Sustainability can be differentiated into several dimensions including environmental, human, social, and economic. According to Becker et al. [2] sustainability dimensions are interdependent and cumulative - first, second and third order effects from each dimension will bleed into each other. Sustainability consideration as a non-functional requirement like security, usability, reliability can help reduce a software system's first order impacts which will also aid reduction of second and third-order impacts of software systems. By doing so, developers have the potential to considerably improve software systems sustainability from the requirement engineering stage onwards [3]. This also requires measures informing how well the development process produces sustainable software [4].

The fundamental question is how to quantify sustainability not only for software products, systems and services but also for the entire digital ecosystem created by the system of software systems? This research aims to serve different communities, though there is still need to conduct empirical studies to validate these benefits. Quantifying sustainability in software systems will encourage software engineering community to develop processes, tools and new metrics to assess sustainability of software system like the other quality attributes. It will help companies, organizations and managers to easily adopt and institutionalize sustainability in their mainstream software development and management processes, assess objectively the cost-benefit while creating a business model associated with sustainability of their software system.

Furthermore, it will guide standardization bodies like ISO and governmental agencies to enact standards and policies for software system sustainability. For example, what is the minimum sustainability level of a software system to get certain accreditation like we do with security today? It will also make the society and people more aware about the impact of software systems when developing and using it; one example is the categorization of a fridge based on its level of greenness (energy usage) such as A+, A++. Shall we adopt the same approach in software engineering?

Kocak et al. [5] stated that software development industry is now getting pressure from regulators to consider green certification. As an answer to this pressure, green attributes of software products should be defined as quality factor. Then, the biggest challenge facing companies is how to integrate sustainability into their engineering practices when knowing the lack of consensus on what sustainability means in software systems and how it can be quantified and measured.

Quantification of sustainability requires that it should be considered among the six divisions in ISO standards SQuaRE Model such as: Quality Management Division, Quality Model Division, Quality Measurement Division, Quality Requirements Division, Quality Evaluation Division, SQuaRE Extension Division [6]. By including sustainability in such standardization framework, sustainability may be considered more effectively in the industry. This is not really the case today. One starting point towards this, is to turn the current meanings, perceptions, and beliefs into requirements, factors, measurable criteria and tangible measures.

This paper presents the early results of an ongoing research that aims to build a theoretical ground for sustainability requirement quantification in software development. Hopefully, the paper can stimulate a discussion as a means of getting feedbacks for further investigations.

The remainder of this paper is as follow. The next section provides various sustainability definitions for requirements. Section III traces the research trends and outcomes from requirement engineering domain. Section IV discusses sustainability in software measurement and propose an approach for it. Section V details the proposed approach with an example. Section VI contains the conclusion with remarks for future work.

II. SUSTAINABILITY DEFINITIONS FOR REQUIREMENT

The varying definitions of sustainability show there are diverse opinions about what is sustainability. This makes it harder to define especially when applied to software systems. Still, these definitions provide a basis to start grounding sustainability in software engineering research and practices. Some clarity is needed as to how to quantify sustainability in software systems in term of quantifiable variables in order to be able to access and evaluate sustainability of software systems.

Sustainable software has been viewed from three angles [7] as:

(1) Long lasting software which relates to how well a piece of software will be able to cope with changes;
(2) Lean software that require less hardware and reduces its own power consumption (energy efficient);
(3) Software for sustainable humans as software that induces sustainable human behavior.

This definition leads to three measurable concerns that we should consider during requirement: energy efficiency, longevity and user experiences.

Venters et al. [8] 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? They stated that in software engineering, longevity and maintenance are the two most important factors for understanding sustainability. Their perception is based on the 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'.

This work highlighted the importance of longevity and maintenance for the requirement of sustainability.

Heiko Koziolek [9] define sustainability of software systems from the perspective of software architecture as long living system that should last for more than 15 years and can be cost-efficiently maintained and evolved over its entire life-cycle. This also supports the requirement of longevity and maintainability.

Tainter [10] introduces sustainability as an active condition of problem solving, not a passive consequence of consuming less resources. To define sustainability in specific context the questions should be to sustain what, for whom, how long and at what cost? Applying Tainter's definition to software systems, it will help frame definition of sustainability into context in order to understand what the boundaries are in a system.

Seacord et al. [11] defined software sustainability as the 'ability to modify a software system based on customer needs and deploy these modifications,' which means sustainability is the quality of conforming to user specification. Modifiability is the key requirement from this definition.

Harris and Goodwin [2] describe sustainability as system that must achieve fairness in distribution and opportunity, adequate provision of social services, including health and education, gender equity, and political accountability and participation. Their definition focus on social sustainability relating to how well a system can cater for different user needs irrespective of their condition. The definition highlights the requirement for accessibility.

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. Base on this definition the main requirements for sustainability can be derived from the economic, environment, social and individual dimensions of sustainability.

Table 1 summarizes the most cited definitions and identifies the key requirements.

TABLE I. DEFINITION SUMMARY AND REQUIRMENTS

Author	Definition	Requirement
M. R. Idio [7]	Long lasting and Lean software, Software for sustainable humans	Energy efficiency, Longevity and User Experiences.
Venters et al. [8]	Sustainability is the quality of being sustained. Longevity and maintenance are the two most important factors for understanding sustainability	Longevity and Maintenance
Heiko Koziolek [9]	Long living system that should last for more than 15 years and can be cost-efficiently maintained and evolved over its entire life-cycle.	Longevity and Maintenance
Seacord et al. [11]	Ability to modify a software system based on customer needs and deploy these modifications	Modifiability
Harris and Goodwin [2]	Sustainability as system that must achieve fairness in distribution and opportunity, adequate provision of social services	Accessibility
Naumann et al. [12]	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	Economic, environment, social and individual

Author	Definition	Requirement
Tainter [10]	To define sustainability in specific context the questions should be to sustain what, for whom, how long and at what cost?	Sustainability is a requirement within a certain context. It requires the specification of the context

III. SUSTAINABILITY IN REQUIREMENT ENGINEERING

The following are some of the research work in the domain of requirement engineering for sustainability in software systems.

Raturi et al. [13] focused on how to develop sustainability as a non-functional 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. The sustainability model has five dimensions (Human, Social, Economic, Environmental and Technical sustainability).

Penzenstadler et al. [14] also support the consideration of sustainability as a nonfunctional requirement like safety and security that are considered as a system quality attribute.

Mahaux et al. [15] highlights the fact that requirements engineering has a major role to play for making software last long by reducing the impact of development and disposal phase.

Roher et al. [16] concerned with the lack of software engineering teams including environmental sustainability during software development proposed the use of sustainability requirement patterns (SRPs) as a guide for software engineers to elicit sustainability requirements.

Becker et al [3] explains the crucial role of requirements not only for software systems but also for how requirement for sustainability can also impact on the social-economic and natural environment. The two case studies presented by the authors' shows the importance of requirement in sustainability design.

Based on the above research, there are three major issues for quantifying sustainability during the requirement stage as seen in the summary in Table I and section III:

- First, different research suggests different definitions, so there is no consensus definition.
- Second, the proposed definitions are either too complex or focus mainly a particular dimension of sustainability.
- Third, there is no central framework that is pivotal to the quantification of sustainability.

This shows there is need for discussing and coming to a consensus by researchers interested in sustainability of software systems. This can enable development of a central formwork that would support the addition of sustainability into the SQuaRE Model [6]. We believe this will foster a focused research towards better quantification of sustainability for software system. It will also encourage research on how best to incorporate management goals and requirements in the adoption of sustainability for software system design and development.

IV. SUSTAINABILITY MEASURES

Sustainability is still not fully explored in the field of software measurement. These are the different works on quantifying sustainability that have been done so far and also attempts to measure sustainability.

Lami et al. [17] stated there are few studies on 'what' aspects of sustainability to measure and 'how' to do it. Calero et al. [18] highlighted that nowadays, sustainability is a key factor that should be considered in the software quality models, though there is less research channeled towards it. Seacord et al. [11] indicated that planning and management of software sustainment is impaired by a lack of consistently applied, practical measures. Without these measures, it is difficult to determine the effect of efforts to improve sustainment practices.

Johann et al. [19] presents a generic metric to measure software energy efficiency and a method to apply it in software engineering process using the formula "Useful Work Done/Used Energy."

Krisztina Erdélyi [20] studies the lifecycle activities of software development with focus on environmental protection by proposing a formula to calculate software waste to encourage the development of green software.

Albertao et al. [4] proposed software engineering metrics based on software quality like reusability, portability, supportability, performance as a way for measuring the sustainability performance of software projects.

Bozzelli et al. [21] 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 highlighting metric types like energy, performance, utilization, software energy consumption.

One of the most referenced model for developing and measuring sustainable software is the Greensoft Model by Naumann et al. [22]. It is a conceptual reference model for "Green Software." The Greensoft model has the objective to support software developers, administrators, and software users in creating, maintaining, and using software in a more sustainable way but lacks the clarity and practical examples of how this model can be implemented for software system development. The key to measuring sustainability of software system requires quantifiable variables that can be applied to all sustainability dimensions in relation to software system development.

Thus, a new proposed approach; Sustainable Business Goal Question Metric (S-BGQM) is introduced here. It encourages the incorporation of sustainability during the entire software system development engineering processes.

S-BGQM is influenced by work from [23] and [24]. It combines results from the software requirement engineering process [23] into the design and development process. It is formed by two major components; the Sustainable Business Assessment and the Goal Question Metric. Figure 1 portrays

S-BGQM. All artefacts in the sustainable business assessment component provides support for all activities in the Goal Question Metric component of S-BGQM.

In the Sustainable Business Assessment, analysis of information in the sustainable business canvas leads to creation of sustainability goals. These goals are categorized into business, usage and system goals with consideration of sustainability that serve as a requirement for measurement. Based on this categorization, a set of questions are generated to characterize all those goals.

System vision and sustainability analysis provide a quick overview of the software system first, second and third order impacts based on those goals. And it provides information useful for specifying the right metric to evaluate the software system.

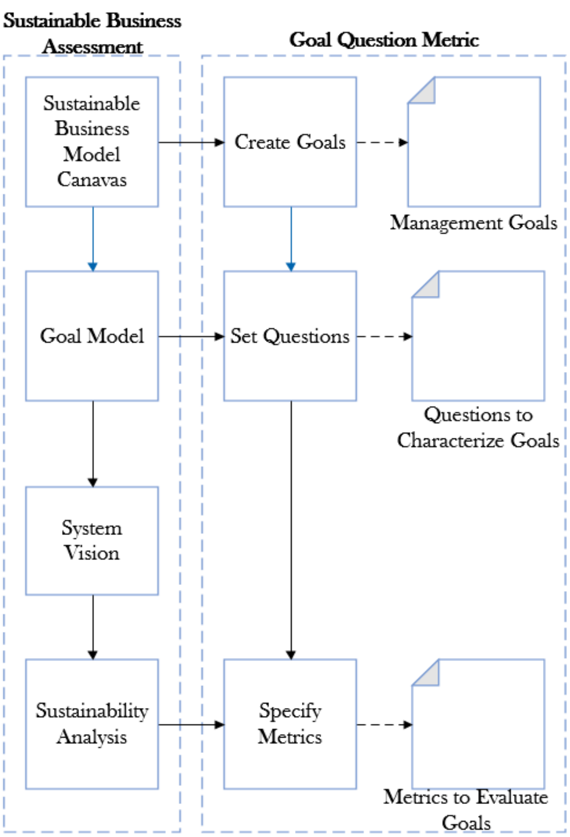

Figure 1 Sustainable Business Goal Metric Process Flow (S-BGQM)

The Sustainable Business Assessment component involves the following (See Figure 1):

- Sustainable Business Model Canvas: The Business Canvas incorporates sustainability considerations during business model design. It allows users to describe, design, challenge, invent, and pivot their business model with sustainability consideration [25] [26].
- Goal Model: It shows comprehensive and holistic goals of the organization or company in relation to the software under development from the economic, social and environmental perspective represented in business goal, usage goal and system goal [27].
- 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 [28].
- Sustainability Analysis: Sustainability analysis describe the system from sustainability perspective by considering sustainability purpose of the system, impact the system has on environment as well as sustainability goal and constraint of the system [29].

The Goal Question Metric (GQM) component covers the following (See Figure 1):

- Tracing and measurement of system goals based on the result from the sustainable business assessment.
- Allows software engineers/ managers and company to define questions that can be used to evaluate their software system goals
- Choose appropriate metrics that can be used to measure their software system base the questions with consideration of sustainability.

These metrics are categorized according the sustainability dimensions as discussed by Raturi et al. [30] and Penzenstadler and Femmer [31] (Economic, Environmental, Social, Individual and Technical Sustainability).

Table II portrays metrics and their categorization according to the five dimensions of sustainability. The metrics samples presented in the GQM component give managers simple yardsticks to calibrate how well their company is doing in terms of resource consumption while extracting more value from their processes. The metrics support decision-making by providing a mechanism for benchmarking performance, tracking improvement over time, evaluating products and processes, and developing strategies for improvement.

TABLE II. METRIC CATEGORIZATION

Category	Metric	Description
Technical	BMI=Number of problems close/number of problems arrival *100	Backlog Management index (BMI) is a workload statement for software maintenance. It is related to both the rate of defect arrivals and the rate at which fixes for reported problems become available.
	Rework Metric	The total number of functions modified per commit related to adding a new feature/function. The "extensibility" of a system is generally the ability of the system to tolerate additional features or functionality with little or no required rework.
Economy	BMI=Number of problems close/number of problems arrival *100	Same as the above BMI
	Defect Density= Total defects/Size	The value of the total defects which are known to the size of the software product

Category	Metric	Description
		calculated.
	Net Cost	The Budgeted Capital - Total Capital Spent
Environment	BMI=Number of problems close/number of problems arrival *100	Same as the above BMI
	Defect Density= Total defects/Size	Same as the above Defect Density
	Energy efficiency	Useful work done/Used Energy
Social	Gateway metric (1=Task success and 0= Task failure)	The amount of successful task completed
	Defect Density= Total defects/Size	Same as the above Defect Density
	Net working hours	Budgeted hours - Total working hours
Individual	Gateway metric (1=Task success and 0= Task	Same as the above Gateway metric

Category	Metric	Description
	failure)	
	Defect Density= Total defects/Size	Same as the above Defect Density

V. S-BGQM Preliminary Study Based on Information Researched Online

The preliminary study described here provides an example of how S-BGQM, as a way of quantifying sustainability works during requirements. A sample project where the project team proposed development of car sharing system called ShareVoyage for students in City of Lappeenranta is presented. It is an online web platform for group shopping and also to share unused foods.

The following seven steps process demonstrate how S-BGQM works while illustrating the different artifacts such as the Sustainable Business Canvas, Goal Model, System vision sustainability analysis of the system and metric worksheet.

1. Create Sustainable Busines Canvas. Figure 3 is an example of a canvas created in this study.

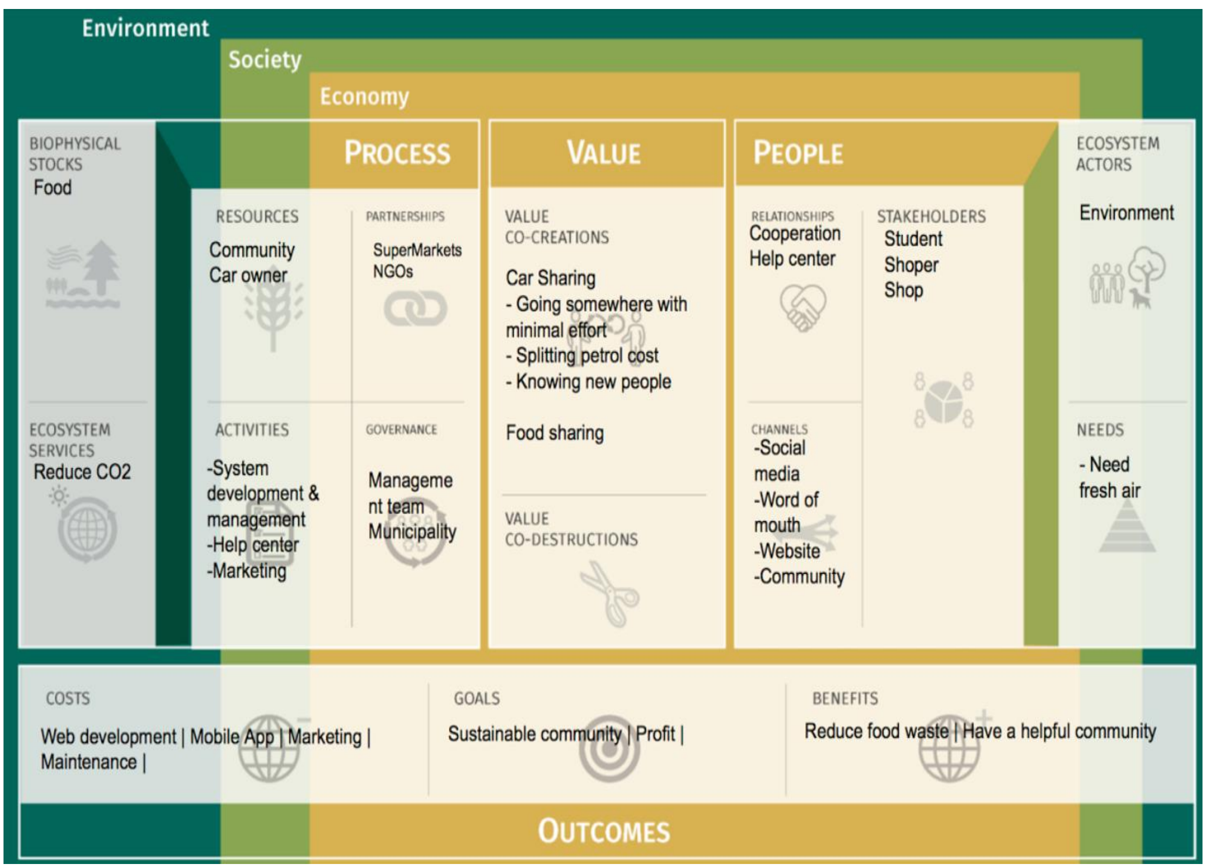

Figure 2. Sustainability Business Model Canvas (Sustainable Business Assessment)

2. Measurable management goals are created based on the information derived from the sustainable business canvas (see Figure 2). These are the goals derived based on the contents from the Canvas :
   - Reduce C02
   - Encourage car sharing
   - Reduce food waste by encouraging food sharing
   - Promote sustainable community

3. All the goals from step 2 are divided into three in the Goal Model phase show the business, usage and system goal of the software system. This division of goals serves as a means of proper classification for easier measurement after system development. Goal model is the basis for early conflict identification and resolution in the system development. Figure 3 shows the details of Goal Model.

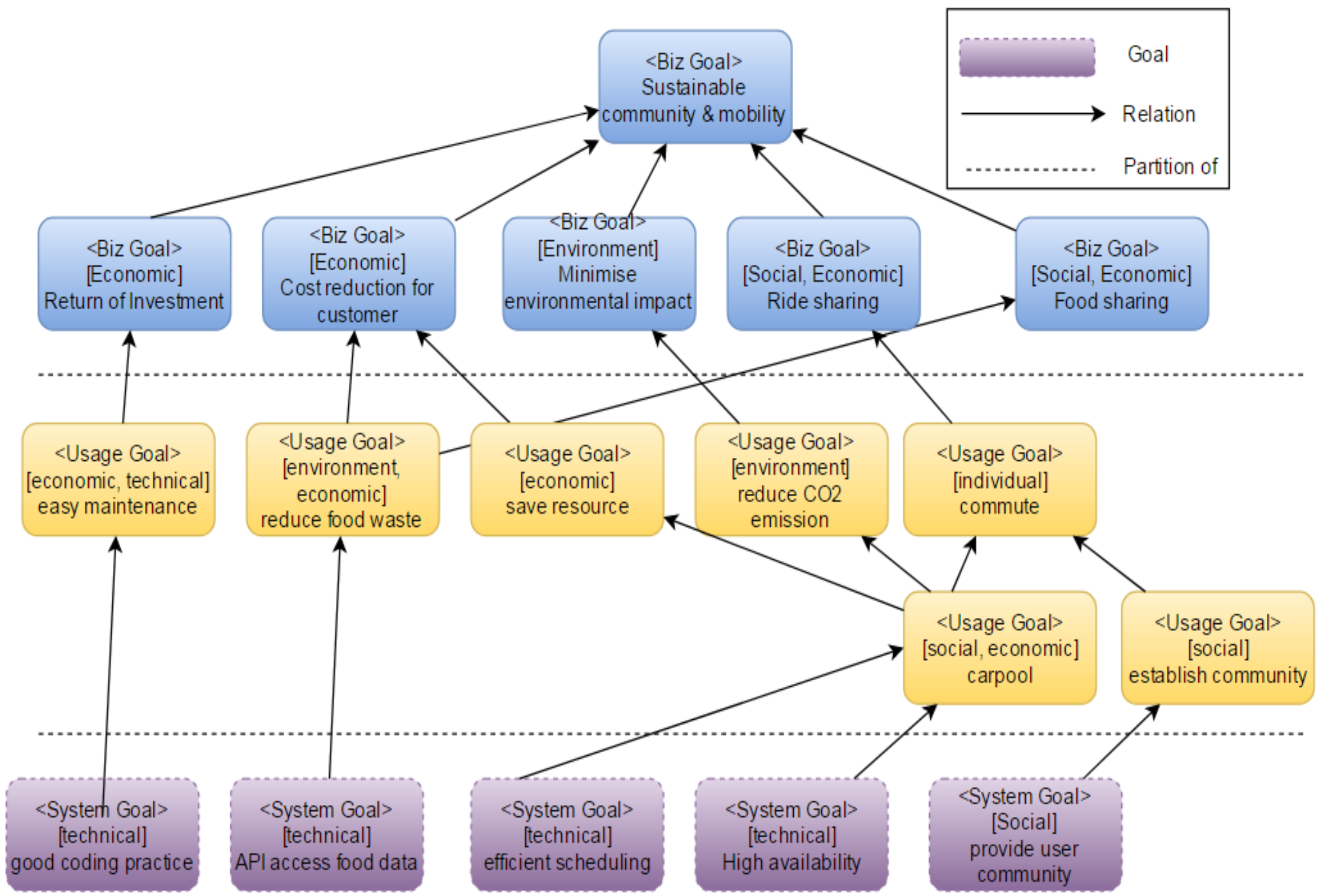

Figure 3. Goal Model (Sustainable Business Assessment)

4. The Biz Goal from Figure 3 represents the business goals that have direct impact on the system. The usage goals are those functional objectives of the system based on how it should behave. The system goals relates to the systems features. The color semantics in Figure 3 is only used to different each section. Based on the Goal Model (see Figure 3), a set of questions is created to characterize each goal. Table II details the questions associated with each goal.

TABLE III. SET QUESTIONS (GQM)

Goals	**Questions**
Reduce C02	Does the application reduce the amount of carbon emission in Lappeenranta?
Encourage car sharing	Is there an increase in car sharing among students?
Reduce food waste	What is the percentage of food waste after the application launch?
Promote sustainable community	Are students more aware of Sustainability?

5. System vision created to show the common understanding of all the stakeholders including users, management staffs, and developers. It is usually a pictorial overview of the system. It portrays how the system functions during operation.

6. Sustainability analysis shows the software system first, second and third order impact as shown in

Figure 4 with consideration for economic, sustainability dimensions. This analysis is based on the inputs from step 1 (sustainable business canvass) on contents of the environment, society, environment, social, individual and technical economy, process, value and people. It provides a holistic view of how different dimension of sustainability impact each other and their relation.

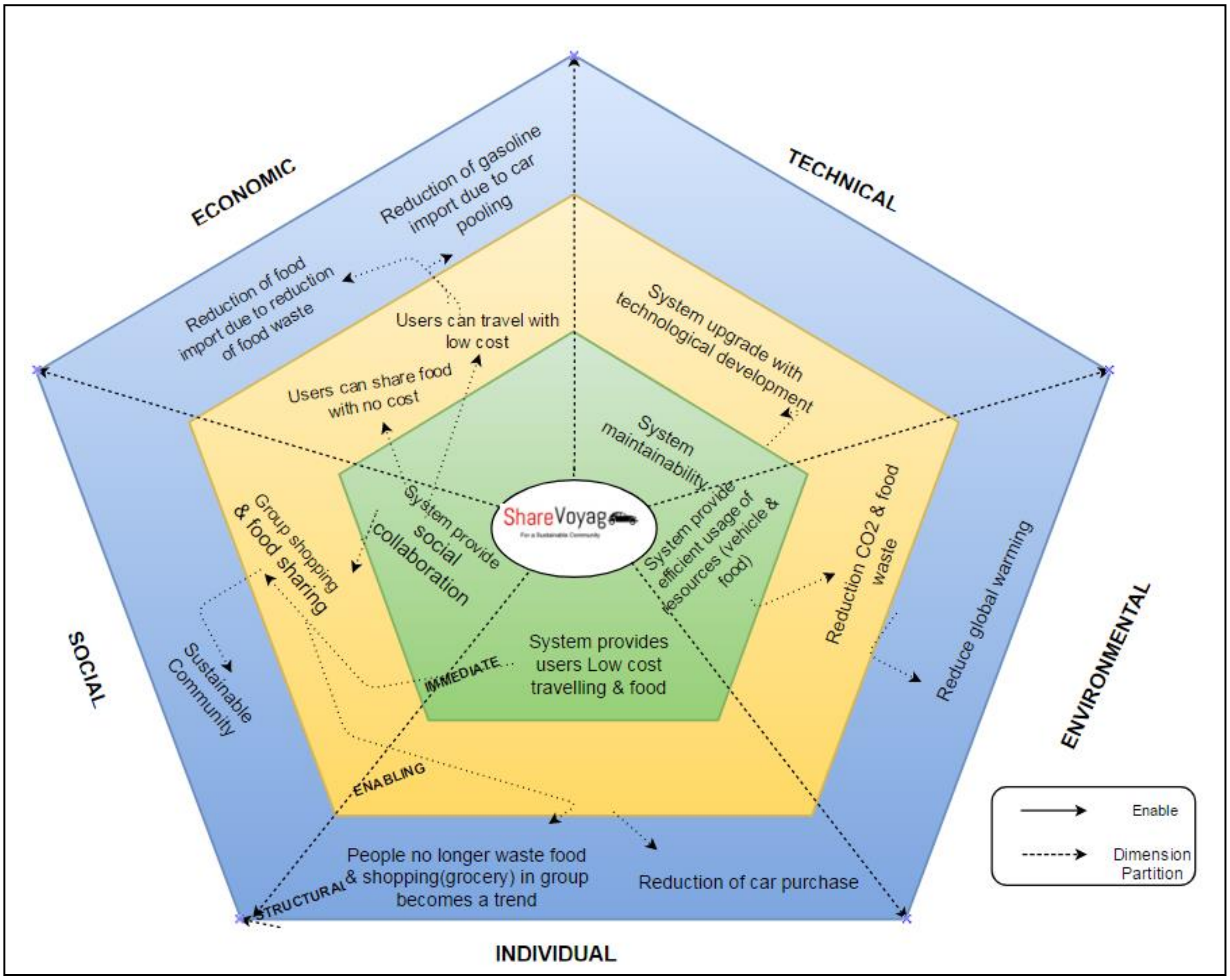

Figure 4. Sustainability Analysis (Sustainable Business Assessment)

7. Based on the system vision and sustainability analysis (see Figure 4) the software development team will be able to generate a metric worksheet (Table III) to evaluate the software system. To clarify, benchmark values are calculated based on the total software project modules and lines of codes.

TABLE IV. METRIC WORKSHEET (GQM)

Category	Question	Metric	Benchmark Value
Technical	What is the Backlog Management Index (BMI)?	BMI=Number of problems close/number of problems arrival *100	0 or 100
	What is the amount of rework?	Rework Metric (Total Number of function modified)	0

Category	Question	Metric	Benchmark Value
Economy	What is the BMI?	BMI=Number of problems close/number of problems arrival *100	0
	What is the software defect density?	Defect Density= Total defects/Size	≤10.46
	Does the actual project cost outweigh budgeted cost?	Net Cost	Positive Number
Environment	What is the BMI?	BMI=Number of problems close/number of problems arrival *100	0 or 100
	What is the defect density?	Defect Density= Total defects/Size	≤10.46
	How much energy does the software consume?	Energy efficiency = Useful work done/Used Energy	
	What is the percentage of car sharing?	Total amount of rides /100	
	What is the percentage of food shared?	Total amount of food share /100	
Social	Can users successfully complete task?	Gateway metric (1=Task success and 0= Task failure)	7
	What is the software defect density?	Defect Density= Total defects/Size	≤10.46
	Are the project teams happy?	Net working hours = Budgeted hours - Total working hours	Positive number
	Are the people more aware of sustainability?	Percentage of food shared	Positive number
Individual	Can users successfully complete task?	Gateway metric (1=Task success and 0= Task failure)	7
	What is the software defect density?	Defect Density= Total defects/Size	≤10.46

The result from Table IV provides a quantifiable result of the system measurement from the five sustainability dimensions. It allows for all-inclusive overview of the system with traces back the questions that are used to characterize each goals during the initial requirement stage.

The procedures and steps in S-BGQM encourage major stakeholders to consider sustainability during the software system development. It can be applied to software development life cycle using the enhancement model for sustainable software engineering proposed by Dick et al. [32]. This model covers sustainability review and preview, sustainability journal, process assessment and sustainability retrospect.

S-BGQM does not cover all aspects of sustainability. There is still need to improve the methodology used in deriving requirements goals from the business assessment component. The lack of intermediate stages to transform sustainability metrics has hinder the ability of S-BGQM to provide a better metric categorization.

VI. Conclusion

As highlighted in this paper, researchers have concentrated their efforts on the definitions and meanings of sustainability. Sometimes, definitions are somehow similar and often they are contradictory or conflicting. There is not yet a general consensus or a common ground on what sustainability and software sustainability means and how it can be quantified objectively. There is an urgent need for the entire software engineering community including practitioners and standardization bodies to have a standardized definition of sustainability, similar to other software quality factors. This will help to ground it in software measurement theories and practices.

We identified a set of sustainability requirements from the most cited definitions. This motivated our research on quantifying sustainability using those requirements. Quantification of sustainability means using variables that are measures of sustainability. We noticed that the biggest issue is that building a model or framework for sustainability quantification or/and defining its measurement scale and measures is already a difficult endeavor. The interpretation of such measures and their validation is a real challenge that requires a long-term research investigations and industry experiments.

Without a standard for software sustainability requirements, it becomes difficult to identify sustainability boundaries. A standard will lead to a unifying consensus that can foster sustainability quantification in software system.

S-BGQM is a modest contribution. We do not claim in this paper that S-BGQM is by itself a completed validated approach or framework. It is a kind of foundation that would be understood as “showing the map or road about what is need to be done to quantify and measure sustainability”. It’s not by itself the right and the unique road but it’s just a possible one.

Our ambition was also to open the doors, or ground the efforts in a research agenda on how to measure sustainability. However we found that these concerns are necessary to overcome the obstacles on this long road for building a model for sustainability. The model should be based on a consensus and it can or should be part of ISO standards. There is a need for software engineering community to create cross-disciplinary research platform, for example building a kind of forum for discussing the definitions, perceptions and understanding of sustainability quantification. That forum can take the form of a new workshop or it can be part of an existing workshop of RE like RE4SuSy or ICSE like the GREENS or it can be a joint book that bring people together to discuss it. This paper also calls for a forum that brings together all the different workshops like GREENS, RE4SuSy, GIBSE, and GinSENG to create a wider consensus.

Based on all these investigation, our intention is to bring this to the workshop discussion community with the hope that it can raise interest among researchers for further research on sustainability requirements, quantification and measurement. The following are some of the issues awaiting for further investigations:

- How to methodically specify sustainability requirements, meaning to quantify it?
- How can the sustainability requirements be measured? What are the measurement scales or measures for those requirements?
- How to categorize the current sustainability metrics and how they related to the five sustainability dimension?

Answers to these questions are a major milestone towards a model of sustainability as a quality attribute. One next stage in our research is a survey to explore sustainability perceptions and practices in industry.

Our future work includes carrying out large-scale industrial case studies to identify the practices of sustainability in software design and also to test the approach proposed in this paper. The goal is also to understand the ways to integrate and measure software sustainability. Another work is to study the process of issuing sustainability and green certification to companies. What are the activities that can be used to improve sustainability practices in the industry? One of such certification is the Albert Sustainable Production Certification [33] and Green Business certification. [34][35].

ACKNOWLEDGMENT

This work is fully supported and funded by DIGI-USER - Smart Services for Digitalization platform in Lappeenranta University of Technology (LUT).

REFERENCES

[1] B. Penzenstadler, "What does Sustainability mean in and for Software Engineering ?," *1st Int. Conf. ICT Sustain.*, 2013.

[2] C. Becker *et al.*, "Sustainability Design and Software: The Karlskrona Manifesto," *Proc. - Int. Conf. Softw. Eng.*, vol. 2, pp. 467–476, 2015.

[3] B. Christoph *et al.*, "Requirements: The key to sustainability," *IEEE Softw.*, vol. 33, no. 1, pp. 56–65, 2016.

[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] 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.

[6] ISO, "ISO/IEC 25010," *https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en*, 2001.

[7] M. R. Idio, "Measuring Sustainability Impact of Software," vol. 16, no. 1, pp. 5–7, 2014.

[8] C. C. Venters *et al.*, "Software sustainability: The modern tower of babel," *3rd Int. Work. Requir. Eng. Sustain. Syst. Work. Proc.*, vol. 1216, pp. 7–12, 2014.

[9] H. Koziolek, "Sustainability Evaluation of Software Architectures : A Systematic Review," *Architecture*, pp. 3–12, 2011.

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

[11] R. Seacord *et al.*, "Measuring Software Sustainability," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[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] 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.

[14] 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.

[15] M. Mahaux and C. Canon, "Integrating the Complexity of Sustainability in Requirements Engineering," *First Int. Work. Requir. Eng. Sustain. Syst.*, 2012.

[16] K. Roher and D. Richardson, "Sustainability requirement patterns," *2013 3rd Int. Work. Requir. Patterns, RePa 2013 - Proc.*, pp. 8–11, 2013.

[17] G. Lami, F. Fabbrini, and L. Buglione, "An ISO / IEC 33000-compliant Measurement Framework for Software Process Sustainability Assessment," pp. 50–59, 2014.

[18] C. Calero, M. F. Bertoa, and M. Angeles Moraga, "A systematic literature review for software sustainability measures," *Green Sustain. Softw. ({GREENS)}, 2013 2nd Int. Work.*, pp. 46–53, 2013.

[19] 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.

[20] 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.

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

[22] 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.

[23] B. Penzenstadler, "Infusing green: Requirements engineering for green in and through software systems," *3rd Intl. Work. Requir. Eng. Sustain. Syst. 2014*, vol. 1216, no. 1, pp. 44–53, 2014.

[24] V. Basili, G. Caldiera, and H. D. Rombach, "Goal Question Metric Paradigm," *Encyclopedia of Software Engineering*. pp. 528–534, 1994.

[25] "3 Minute Introduction to Strongly Sustainable Business Model Canvas, https://prezi.com/k5x2civcaw7y/3-minute-introduction-to-strongly-sustainable-business-model-canvas/ , Accessed on 8-06-2017," no. November, p. 2014, 2014.

[26] "Strongly Sustainable Business Model Canvas Http://www.ssbmg.com/ Accessed on 8-06-2017."

[27] B. Penzenstadler, "Goal Model Requirements Engineering for Sustainability," *Lect. slides guest course Softw. Eng. Sustain. Lappeenranta Univ. Technol. 2016. Slides available online http//birgit.penzenstadler.de/teach/LUT.html.*

[28] B. Penzenstadler, "System Vision Requirements Engineering for Sustainability," *Lect. slides guest course Softw. Eng. Sustain. Lappeenranta Univ. Technol. 2016. Slides available online http//birgit.penzenstadler.de/teach/LUT.html.*

[29] B. Penzenstadler, "Requirements Engineering for Sustainability -

Sustainability Analysis," *Lect. slides guest course Softw. Eng. Sustain. Lappeenranta Univ. Technol. 2016. Slides available online http//birgit.penzenstadler.de/teach/LUT.html.*

[30] 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.

[31] 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.

[32] M. Dick and S. Naumann, "Enhancing Software Engineering Processes towards Sustainable Software Product Design," vol. 2010, pp. 706–715, 2010.

[33] Bbc, "albert certification . Available at: http://www.bbc.co.uk/responsibility/environment/albert-plus."

[34] C. G. B. Network, "Green Business Certification, Available at: http://www.greenbusinessca.org/."

[35] G. B. Bureau, "Green Business Available at : https://greenbusinessbureau.com/how-gbb-certification-works/," no. June, pp. 1–6, 2013.