International Journal of Industrial Engineering & Production Research June 2024 Vol. 35, No. 2: 1-26
DOI: 10.22068/ijiepr.35.2.1889

RESEARCH PAPER

Integrated Delphi-Interpretive Structural Modeling for Sustainable Green Software Development

Zahra Taherikhonakdar[1] & Hamed Fazlollahtabar[2*]

Received 13 September 2023; Revised 5 March 2024; Accepted 6 April 2024;

ABSTRACT
These days, industries, individuals and organizations are highly dependent on software. Software plays an important role in our daily life. They use in embedded systems, databases, computers, mobiles etc. Great demand for information and communication technology (ICT) causes environmental problem and endanger the future sustainability. In this case, sustainable development has become a hot research topic in software engineering community. Sustainability as a software quality is a general term. Therefore, there is a chance that software developers mislead about developing sustainable software. Thus, there are some questions that should be answered to help practitioners to develop sustainable software: how developers could develop green and sustainable software? What requirements should be considered to reach green and sustainable software? Which non-functional requirement has an effect on each sustainability dimension? In this paper, we selected 20 non-functional requirements out of 60 collected from the literature. It was identified the effective non-functional requirements in green and sustainable software development by using Delphi method integrated with the interpretive structural modeling (ISM). The study aimed to pave the way for software eco-labeling and help users to choose the green and sustainable one. Also, provide software developers with guideline to develop green and sustainable software by identifying effective non-functional requirements. This would lead to the sustainable future and green environment..

KEYWORDS: *Sustainable software development; Green software; Delphi method; Interpretive structural modeling.*

1. Introduction

Nowadays, software attracts researchers and practitioners' attention as our society intensively is dependent on software systems ranging from small-scale software systems like: smartphones, laptops, embedded systems or large-scale software like big databases. Therefore, software has a great effect on our daily routine life [1]. As a result, software could have a potential threat to both future sustainability and environment greenness.

In the past, green information technology (IT) focuses on hardware design to mitigate their negative effect [2]. With no surprise hardware energy efficiency has dramatically improved [3]. But researchers realized that just by improving hardware can reach only limited efficiency in the system and also notice that hardware designers can only do so much until it becomes the responsibility of software developers to develop software in an energy efficient and sustainable ways [4]. Since software plays an important role in platform's overall energy efficiency, one mis-behaving, power unfriendly software component on the system can thwart any power management benefits built into the hardware [5]. In recent years, consequently, developers and researchers' attention have gone beyond hardware level to reach the software level for having sustainable and green system [2]. Sustainable software systems are still great challenge for software engineers involved in software development process in spite of the emergence clear and systematic approaches [1].

Software sustainability is a demand in our modern world, which is highly dependent on

* **Corresponding author: Hamed Fazlollahtabar**
h.fazl@du.ac.ir

1. *Computer Science Department, University of Windsor, Ontario, Canada.*
2. *Deaprtment of Industrial Engineering, School of Engineering, Damghan University, Damghan, Iran.*

complex software systems [1, 6]. We have become conscious about reducing our energy consumption and our carbon footprint. Software sustainability is increasingly gaining attention and importance and like occurs in any other new discipline, however some attempts have been made but there are still many misunderstandings. There are several following definitions for sustainability in the literature:

1- "Development that meets the needs of present without compromising the ability of future generations to meet their own needs" [7].

2- "The capacity of the software-intensive system itself to endure will be a concern for the operating organization and the community" [8].

Also, green and sustainable software is defined as "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" [9, 10]. However, research communities only considered one dimension for sustainability, but others consider several dimensions [11, 12]. In this study following dimension are concerned [1, 13]:

- The environmental dimension is concerned with the long-term effect of software development, maintenance and use on energy consumption and the usage of other resources, which includes natural resources, pollution and waste, etc.
- The economic dimension focuses on assets, capital and benefit that comprises wealth creation, profitability and how the software life cycle process protect stakeholder's investment. This dimension directly related to a software product characteristic that is called "software greenness".
- The human (individual) dimension refers to the well-being of humans as individuals, which includes mental and physical well-being, education, freedom and livability, etc. It should be noted that from the social dimension of sustainability, and due to the consideration of human as a portion of society, we considered the human dimension as a software developer for this research.

Software sustainability is increasingly gaining attention and importance and like occurs in any other new disciplines, however, some attempts have made but there are still many misunderstandings. Sustainability as a 21st century software's quality is a comprehensive term in software development [14]. There is lack of information about how achieve sustainability in software's systems. Therefore, this study conducted to find out which software non-functional requirements affect the software sustainability and software greenness. Our objective is, take a step forward to eco-labeling software based on specified requirements. This can be useful in several ways. For instance, it can encourage developers to develop green software and sustainable software. In addition, it can help software users to choose their software for their personal use or even their organizations base on the software's eco-label. Hence, by having green and sustainable software we have green and sustainable environment, society and future.

2. Related Work

In this section, we point out to the researches that have been done in this area to show our aim is different from other researchers. Most researchers attempt to clearly define sustainability in the realm of software from different perspective like software quality and software architecture [15]. Also clarify the different aspects of the software sustainability [13]. Other researches point out that greenness and sustainability should be considered as software non-functional requirement [2, 14].

Garcia-Mireles [15] developed profiles of the sustainability concept as a quality model according to the approaches to classify quality models. In this research, the following dimensions are considered: object, purpose, perspective, application domain, and sustainability dimension and quality terms. A set of software quality papers that addressing sustainability were selected by using Systematic Mapping Study (SMS). These set of classified papers addressed software product entity, concentrate on both estimating and specifying sustainability qualities. Some schemes extent the ISO/IEC 25010 quality model to address sustainability and all papers that categorized are concentrated on the development stage of a software product.

Venters et al. [1] concentrated on the software architectures potential and architectural design choices to endure over time. Software architecture is a bedrock of the sustainable development since architecture sustainability is adaptable to changes that occur in long time and was discussed that if software architectures be cost efficiently maintained and evolve over their entire life cycle, they can be a fundamental to achieving software sustainability. Salam and Khan [16] identifies a list of 20 success factors for the development of green and sustainable software, via systematic literature review (SLR) process from a sample of 74 research papers. Seven factors out of these 20 factors categorized as critical success factor. These critical factors are: 'green software design and efficient coding', 'power-saving software strategies', 'low carbon emission throughout the software development process', 'efficient resource utilization', 'paperless communication', 'filtration of requirements through green evaluator' and 'e-waste management'. The identified factors could lead vendors towards the development of green and sustainable software.

The current software development lifecycle focuses on systematic execution and maintenance. Certain important decisions taken in software development phases like use of paper, generation of e-waste, power consumption etc. could be harmful to the environment directly and indirectly and that is a problem here. Shenoy and Eeratta [17] suggested some practices in software requirements gathering, design, implementation, testing, deployment and maintenance that help in developing a software system in an environment friendly way. If the presented ideas of this paper be practiced helps in reducing power consumption, use of paper, pollution etc.

Therefore, there is a lack of road map to guide developers and software engineers to develop green and sustainable software, guide and encourage users and organizations to demand, choose green and sustainable ones. In this paper, by analyzing software non-functional requirements we will find out if they have an effect on software greenness or sustainability. The result of this research can be the step forward to software eco-labeling and having green and sustainable future.

3. Methodology

Requirements are the bedrock for all software products [18]. Set of functional and non-functional requirements determine software function and behavior [19]. Therefore, software requirements play an important role in software sustainability and greenness. If we treated software sustainability and greenness as a non-functional requirement, then the question is how we can implement them. Although sustainability is considered as a software quality characteristic [14], sustainability and also greenness are comprehensive terms, that means, for gaining sustainable software or green software developers should consider some specific non-functional requirement which help them to reach the sustainability and greenness. In this paper, we identified these non-functional requirements to help developers. The software development team could reduce the cost of maintenance of the software in long term also reduce e-waste, carbon footprint and energy consumption by developing software in sustainable and green way. The factors are collected from the literature. In this paper, two popular decision-making method, Delphi method and Interpretive Structural Modelling (ISM), were applied to identify non-functional requirements and determine their effect on developing green and sustainable software.

3.1. Delphi method

To identify non-functional requirements, we used Delphi method, which is a structured communication technique, or method, originally developed as a systematic, interactive forecasting method, which relies on a panel of experts. Delphi has been widely used for business forecasting and has certain advantages over another structured forecasting approach, prediction markets [20]. This method includes the specialist of the research subject who should answer the questions in several round until achieve agreement. Developing a technique to achieve the most reliable consensus of a group of experts was the aim of this method [21].

3.2. Interpretive structural modelling (ISM)

ISM is a conventional methodology to identify relationships among specific items. This approach has been highly used by different researchers to represent the interrelationships among various items related to the issue [22]. ISM technique involved following steps:

Step 1. In this step a set of elements that are relevant to the issue were identified.

Step 2. In this phase, structural self-interaction matrix (SSIM) of the elements was developed. In this matrix, four following symbols are used that show the contextual relationship for each factor and the existence of a relationship between any two factors (i and j):

(a) V if element i will be influenced by element j.
(b) A if element j will be influenced by element i.
(c) X if elements i and j will influence each other.
(d) O if barriers i and j are unrelated [22].

Step 3. Develop a reachability matrix from SSMI that was developed in previous step. The rules in this step are as follows:
(a) If the (i, j) entry in the SSIM is V, then the (i, j) entry in the reachability matrix becomes one and the (j, i) entry becomes 0.
(b) If the (i, j) entry in the SSIM is A, then the (i, j) entry in the matrix becomes zero and the (j, i) entry becomes one.
(c) If the (i, j) entry in the SSIM is X, then the (i, j) entry in the matrix becomes one and the (j, i) entry also becomes 1.
(d) If the (i, j) entry in the SSIM is O, then the (i, j) entry in the matrix becomes zero and the (j, i) entry also becomes 0.
Following these rules, the initial reachability matrix is prepared [22].

Step 4. In this phase, for each factor, reachability sets and antecedent sets are obtained from the final reachability matrix. Reachability set involve the factor itself and other factor that may impact, whereas the antecedent set involve the factor itself and the other factor that may impact it. Afterwards, for all the elements, the crossing of these sets is derived and different elements' levels are determined. The element's which the reachability and the crossing sets are the same insert in the top level of the ISM hierarchy. The top-level elements mean these elements will not lead the other factors above their own level in the hierarchy. Once the element's level is identified, it is removing from the list. Then the same process is repeated until all element's level is found [22].

Step 5. In this step, conical matrix was developed by clustering factors in the same level across the rows and columns of the final reachability matrix. The drive power of a factor is derived by summing up the number of ones in the rows and its dependence power by summing up the number of ones in the columns. Then, drive power and dependence power ranks are calculated by giving highest ranks to the factors that have the maximum number of ones in the rows and columns, respectively [22].

Step 6. Finally, the ISM preliminary model is developed from reachability matrix and the level of factors is determined.

Then, drive power and dependence power of factors are analyzed by MICMAC analysis (cross-impact matrix multiplication applied to classification). Multiplication properties of matrix are MICMAC fundamental. The MICMAC done to identify the key factors that drive the systems in different categories. According to factor's drive power and dependence power, the factors, have been classified into four categories which following as:

1. Autonomous factors have both weak drive and dependence power. These factors approximately detached from the system.
2. Dependent factors have weak drive power but strong dependence power.
3. Linkage factors have strong drive power and strong dependence power. Any actions of this these factors affected others in addition feedback affects themselves.
4. Independent factors have strong drive power but weak dependence power. These factors that have very strong drive power, called the key factors.

4. Implementation

In order to conduct this research, we gathered 60 non-functional requirements from the literature, which are commonly used in developing software (Table 1). Delphi method was used to make the final list of requirements out of 60 requirements. Twenty of requirements were considered as a most important and commonly used in software development (Table 2) by Delphi method and answering following questions:

- Which requirements are considered as an important software quality, in general?
- Which requirements are essential considering today's software market, in general?
- Which requirements are useful for sustainable software or green software?

Then, by using Interpretive structural modelling (ISM) the effect of the identified non-functional requirements on software greenness and software sustainability in three dimensions of environmental, economic, and human was scrutinized. Environmental sustainability could call "software greenness" as it is related to a software product characteristic [13]. ISM was implemented for each software sustainability dimension to study the effect of identified non-functional requirements on software sustainability and greenness.

Step1:

First, the elements, which related to the research subject, were identified. In this research, the elements include 20 non-functional requirements (Table 2) and five items were considered in each software sustainability dimensions, which consist of following items [1, 13]:

- Environment dimension: V1. Energy usage, V2. Resource usage, V3.

Carbon footprints, V4. Waste/ e-waste, V5. pollution

- Economic dimension: V1. Assets, V2. Profitability, V3. Wealth creation, V4. Protect stakeholders' investment, V5. Benefits.
- Human dimension: V1. Mental well-being, V2. Social well-being, V2. Education, V4. Freedom, V5. Livability.

Step 2:

In this step, structural self-interaction matrix (SSIM) was developed based on the contextual relationship for each factor. According to this research, there would be three structural self-interaction matrix (SSIM) which represented the elements in three dimensions of sustainability (Figure 1, Figure 2, Figure 3).

Step 3:

Then, we developed reachability matrix from each SSMI in Step 2 which is showed in Figure 4, Figure 5, Figure 6.

Tab. 1. Nonfunctional requirement list collected from the literature

Accessibility	Efficiency	Modifiability	Seamlessness
Accuracy	Evolvability	Modularity	Self-sustainability
Adaptability	Extensibility	Mobility	Serviceability
Affordability	Fault-tolerance	Operability	Simplicity
Agility	Fidelity	Portability	Stability
Auditability	Flexibility	Precision	Sustainability
Autonomy	Inspectability	Predictability	Traceability
Availability	Installability	Recoverability	Ubiquity
Compatibility	Integrity	Reliability	Understandability
Configurability	Deployability	Repeatability	Upgradability
Correctness	Dependability	Reproducibility	Usability
Credibility	Failure transparency	Resilience	Provability
Customizability	Interchangeability	Responsiveness	
Disreputability	Learnability	Reusability	
Durability	Maintainability	Safety	
Effectiveness	Manageability	Scalability	

Tab. 2. Selected non-functional requirement list using delphi

V6. Accessibility	V13. Correctness	V20. Modularity
V7. Accuracy	V14. Customizability	V21. Reliability
V8. Adaptability	V15.Durability	V22. Reusability
V9. Agility	V16. Extensibility	V23. Simplicity
V10. Availability	V17. Flexibility	V24. Upgradability
V11. Compatibility	V18. Maintainability	V25. Usability
V12.Configurability	V19. Modifiability	

	accessi-bility	accur-acy	adapt-ability	agil-ity	avail-ability	compati-bility	configur-ability	correct-ness	customiz-ability	durab-ility	extensib-ility	flexib-ility	maintain-ability	modifiab-ility	modula-rity	reliabil-ity	reusab-ility	simpl-icity	upgrad-ability	usab-ility	pollu-tion	waste	carbon footprints	resource usage	energy usage
energy-usage	O	O	O	O	A	A	O	A	O	O	O	O	O	O	A	O	O	O	O	O	O	O	O	O	
resource usage	A	A	A	O	A	A	O	A	A	O	O	O	O	O	A	O	A	A	A	O	O	O	O		
carbon footprints	O	A	A	A	A	O	A	O	O	O	O	O	O	O	O	O	A	O	O	O	O	O			
waste	O	O	A	A	O	O	O	O	O	A	A	A	A	A	O	O	O	O	O	O	O				
pollution	O	O	O	O	O	A	O	O	O	O	O	O	O	O	O	O	O	O	O	O					
usability	O	A	O	O	A	O	O	A	O	V	A	A	A	A	O	V	O	A	O						
upgradability	O	O	O	O	O	O	O	O	O	O	A	A	O	O	A	O	O	O							
simplicity	O	O	O	O	O	O	O	O	O	O	O	O	V	O	O	O	O								
reusability	O	O	O	O	O	O	O	O	O	O	O	V	O	O	O	O									
reliability	O	X	A	O	X	O	O	X	O	V	O	A	O	A	O										
modularity	O	O	O	O	O	O	O	O	V	O	O	V	V	O											
modifiability	O	O	O	O	O	O	X	O	O	O	A	A	V												
maintain-ability	O	A	A	O	O	A	A	A	O	A	A	A													
flexibility	O	O	V	A	O	O	V	O	O	O	X														
extensibility	O	O	O	A	O	O	V	O	O	V															
durability	O	O	O	O	V	O	O	O	O																
customiza-bility	V	O	O	O	O	O	O	O																	
correctness	O	V	O	O	O	O	O																		
configur-ability	O	O	O	O	O	O																			
compatibility	O	O	O	O	O																				
availability	O	O	O	O																					
agility	O	O	O																						
adaptability	O	O																							
accuracy	O																								
accessibility																									

Fig. 1. Environmental sustainability- SSIM

	accessibility	accuracy	adaptability	agility	availability	compatibility	configurability	correctness	customizability	durability	extensibility	flexibility	maintainability	modifiability	modularity	reliability	reusability	simplicity	upgradability	usability	benefits	protect invesme	wealth creation	profitability	assets
assets	O	O	O	A	A	O	O	O	O	A	O	A	O	O	O	O	O	O	O	O	O	O	O	O	
profitability	A	A	A	A	A	A	A	A	O	A	A	A	A	A	O	O	A	O	O	O	O	O	O		
wealth creation	O	O	A	A	O	O	O	O	A	A	O	O	O	A	O	A	A	A	A	A	O	O			
protect invesment	A	A	A	A	A	A	O	A	A	A	A	A	A	A	O	O	A	O	O	O	O				
benefits	A	A	A	O	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A					
usability	O	A	O	O	A	O	O	A	O	V	A	A	A	A	O	V	O	A	O						
upgradability	O	O	O	O	O	O	O	O	O	O	A	A	O	O	A	O	O	O							
simplicity	O	O	O	O	O	O	O	O	O	O	O	O	V	O	O	O	O								
reusability	O	O	O	O	O	O	O	O	O	O	O	V	O	O	O	O									
reliability	O	X	A	O	X	O	O	X	O	V	O	A	O	A	O										
modularity	O	O	O	O	O	O	O	O	V	O	O	V	V	O											
modifiability	O	O	O	O	O	O	X	O	O	O	A	A	V												
maintainability	O	A	A	O	O	A	A	A	O	A	A	A													
flexibility	O	O	V	A	O	O	V	O	O	O	X														
extensibility	O	O	O	A	O	O	V	O	O	V															
durability	O	O	O	O	V	O	O	O	O																
customizability	V	O	O	O	O	O	O	O																	
correctness	O	V	O	O	O	O	O																		
configurability	O	O	O	O	O	O																			
compatibility	O	O	O	O	O																				
availability	O	O	O	O																					
agility	O	O	O																						
adaptability	O	O																							
accuracy	O																								
accessibility																									

Fig. 2. Economic sustainability- SSIM

	accessibility	accuracy	adaptability	agility	availability	compatibility	configurability	correctness	customizability	durability	extensibility	flexibility	maintainability	modifiability	modularity	reliability	reusability	simplicity	upgradability	usability	livability	freedom	education	social well-being	mental well-being
mental well-being	A	A	A	A	A	A	A	A	O	A	A	A	A	A	O	O	O	A	O	A	O	O	O	O	
social well-being	A	O	O	O	A	A	A	A	O	A	O	A	A	A	O	A	O	O	O	A	O	O	O		
education	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O			
freedom	A	O	A	O	O	O	A	O	A	A	A	A	A	A	O	O	O	O	O	O	O				
livability	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A					
usability	O	A	O	O	A	O	O	A	O	V	A	A	A	A	O	V	O	A	O						
upgradability	O	O	O	O	O	O	O	O	O	O	A	A	O	O	A	O	O	O							
simplicity	O	O	O	O	O	O	O	O	O	O	O	O	V	O	O	O	O								
reusability	O	O	O	O	O	O	O	O	O	O	O	V	O	O	O	O									
reliability	O	X	A	O	X	O	O	X	O	A	O	A	O	A	O										
modularity	O	O	O	O	O	O	O	O	V	O	O	V	V	O											
modifiability	O	O	O	O	O	O	X	O	O	O	A	A	V												
maintainability	O	A	A	O	O	A	A	A	O	A	A	A													
flexibility	O	O	V	A	O	O	V	O	O	O	X														
extensibility	O	O	O	A	O	O	V	O	O	V															
durability	O	O	O	O	V	O	O	O	O																
customizability	V	O	O	O	O	O	O	O																	
correctness	O	V	O	O	O	O	O																		
configurability	O	O	O	O	O	O																			
compatibility	O	O	O	O	O																				
availability	O	O	O	O																					
agility	O	O	O																						
adaptability	O	O																							
accuracy	O																								
accessibility																									

Fig. 3. Human sustainability- SSIM

	energy usage	resource usage	carbon footprints	waste	pollu-tion	usab-ility	upgrad-ability	simpl-icity	reusab-ility	reliabil-ity	modula-rity	modifiab-ility	maintain-ability	flexib-ility	extensib-ility	durab-ility	customiz-ability	correct-ness	configur-ability	compati-bility	avail-ability	agil-ity	adapt-ability	accur-acy	accessi-bility
energy-usage	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
resource usage	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
carbon footprints	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
waste	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
pollution	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
usability	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
upgradability	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
simplicity	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
reusability	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
reliability	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
modularity	1	1	0	0	0	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0
modifiability	0	0	0	1	0	1	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
maintain-ability	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
flexibility	0	0	0	1	0	1	1	0	0	1	0	1	1	1	1	0	0	0	1	0	0	0	1	0	0
extensibility	0	0	0	1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0	0
durability	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
customiza-bility	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
correctness	1	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
configur-ability	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
compatibility	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
availability	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
agility	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0
adaptability	0	1	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
accuracy	0	1	1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
accessibility	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fig. 4. Environmental sustainability- reachability matrix

	assets	profit-ability	wealth creation	protect invesment	benefits	usab-ility	upgrad-ability	simpl-icity	reusab-ility	reliabil-ity	modula-rity	modifiab-ility	maintain-ability	flexib-ility	extensib-ility	durab-ility	customiz-ability	correct-ness	configur-ability	compati-bility	avail-ability	agil-ity	adapt-ability	accur-acy	accessi-bility
assets	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
profitability	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
wealth creation	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
protect invesment	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
benefits	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
usability	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
upgradability	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
simplicity	0	0	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
reusability	0	1	1	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
reliability	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
modularity	0	0	0	0	1	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0
modifiability	0	1	1	1	1	1	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
maintain-ability	0	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
flexibility	1	1	0	1	1	1	1	0	0	1	0	1	1	1	1	0	0	0	1	0	0	0	1	0	0
extensibility	0	1	0	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0	0
durability	1	1	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
customiza-bility	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
correctness	0	1	0	1	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
configur-ability	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
compatibility	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
availability	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
agility	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0
adaptability	0	1	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
accuracy	0	1	0	1	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
accessibility	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fig. 5. Economic sustainability- reachability matrix

	mental well-being	social well-being	educa-tion	freed-om	livabi-lity	usab-ility	upgrad-ability	simpl-icity	reusab-ility	reliabil-ity	modula-rity	modifiab-ility	maintain-ability	flexib-ility	extensib-ility	durab-ility	customiz-ability	correct-ness	configur-ability	compati-bility	avail-ability	agil-ity	adapt-ability	accur-acy	accessi-bility
mental well-being	1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
social well-being	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
education	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
freedom	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
livability	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
usability	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
upgradability	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
simplicity	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
reusability	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
reliability	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
modularity	0	0	0	0	1	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0
modifiability	1	1	0	1	1	1	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
maintain-ability	1	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
flexibility	1	1	0	1	1	1	1	0	0	1	0	1	1	1	1	0	0	0	1	0	0	0	1	0	0
extensibility	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0	0
durability	1	1	0	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
customiza-bility	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
correctness	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
configur-ability	1	1	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
compatibility	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
availability	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
agility	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0
adaptability	1	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
accuracy	1	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
accessibility	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fig. 6. Human sustainability- reachability matrix

Step 4:

In this phase, the level of each element was determined. As Figures 7, 8, and 9 showed in all three-sustainability dimension modularity, agility and reusability are in the low-level of ISM hierarchy that mean these non-functional requirements play a significant role in developing green and sustainable software. In other word, these non-functional requirements have a most effect on other factors. Flexibility and extensibility are in the fourth level. Simplicity, modifiability, configurability, customizability, compatibility and adaptability are in the third level. Usability, reliability, maintainability, durability, correctness, availability, accuracy, upgradability upgradability and accessibility are in the second level. These non-functional requirements have highest driving power and play an important role to reach sustainability in software, respectively.

In environment dimension Energy usage, resource usage, carbon footprint, waste and pollution, in economic dimension assets, profitability, wealth creation, protect investment and benefits, in human dimension mental well-being, social well-being, education, freedom and livability are in the top level which means these factors have a less effect on other factors. They are affected by other factors that are in top levels.

Variabels	Reachability set	Antecedent set	Commen set	Level
energy usage	energy usage	energy usage, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	energy usage	1
resource usage	resource usage	resource usage, usability, upgradability, simplicity, reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy, accessibility	resource usage	1
carbon footprints	carbon footprints	carbon footprint, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	carbon footprints	1
waste	waste	waste, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	waste	1
pollution	pollution	pollution	pollution	1
usability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy,	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
upgradability	upgradability	upgradability, reusability, modularity, flexiability, extensability, agility	upgradability	2
simplicity	simplcity	simplicity	simplicity	3
reusability	reusability, flexiability, extensability	reusability	reusability	5
reliability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
modularity	modularity	modularity	modularity	5
modifiability	modifiability, configurability	reusability, modularity, modifiability, flexiability, extensability, configurability, agility	modifiability,configurability	3
maintainability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
flexibility	flexiability, extensability,	reusability, modularity, flexiability, extensability, agility	flexiability, extensability	4
extensibility	flexiability, extensability	reusability, modularity, flexiability, extensability, agility	flexiability, extensability	4
durability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
customizability	customizability	modularity, customizability	customizability	3
correctness	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
configurability	modifiability, configurability	reusability, modularity, modifiability, flexiability, extensability, configurability,	modifiability, configurability	3
compatibility	compatibility	compatibility	compatibility	3
availability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
agility	agility	agility	agility	5
adaptability	adaptability	reusability, modularity, flexiability, extensability, agility, adaptability	adaptability	3
accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
accessibility	accessability	modularity, customizability, accessibility	accessibility	2

Fig. 7. Environment dimension

Variabels	Reachability set	Antecedent set	Commen set	Level
assets	assets	assets, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	assets	1
profitability	profitability	profitability, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy, accessibility	profitability	1
wealth creation	wealth creation	wealth creation, usability, upgradability, simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	wealth creation	1
protect invesment	protect invesment	protect investment, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy, accessibility	protect invesment	1
benefits	benefits	benefits, usability, upgradability, simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy, accessibility	benefits	1
usability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy,	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
upgradability	upgradability	upgradability, reusability, modularity, flexiability, extensability, agility	upgradability	2
simplicity	simplcity	simplicity	simplicity	3
reusability	reusability, flexiability, extensability	reusability	reusability	5
reliability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
modularity	modularity	modularity	modularity	5
modifiability	modifiability, configurability	reusability, modularity, modifiability, flexiability, extensability, configurability, agility	modifiability,configurability	3
maintainability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
flexibility	flexiability, extensability,	reusability, modularity, flexiability, extensability, agility	flexiability, extensability	4
extensibility	flexiability, extensability	reusability, modularity, flexiability, extensability, agility	flexiability, extensability	4
durability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
customizability	customizability	modularity, customizability	customizability	3
correctness	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
configurability	modifiability, configurability	reusability, modularity, modifiability, flexiability, extensability, configurability, agility	modifiability, configurability	3
compatibility	compatibility	compatibility	compatibility	3
availability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
agility	agility	agility	agility	5
adaptability	adaptability	reusability, modularity, flexiability, extensability, agility, adaptability	adaptability	3
accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
accessibility	accessability	modularity, customizability, accessibility	accessibility	2

Fig. 8. Economic dimension

Variabels	Reachability set	Antecedent set	Commen set	Level
mental well-being	mental well-being	mental well-being, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy, accessibility	mental well-being	1
social well-being	social well-being	social well-being, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy, accessibility	social well-being	1
education	education	education	education	1
freedom	freedom	freedom, usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy, accessibility	freedom	1
livability	livability	livability, usability, upgradability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, customizability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy, accessibility	livability	1
usability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy,	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
upgradability	upgradability	upgradability, reusability, modularity, flexiability, extensability, agility	upgradability	2
simplicity	simplcity	simplicity	simplcity	3
reusability	reusability	reusability	reusability	5
reliability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
modularity	modularity	modularity	modularity	5
modifiability	modifiability	reusability, modularity, modifiability, flexiability, extensability, configurability, agility	modifiability	3
maintainability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
flexibility	flexiability, extensability	reusability, modularity, flexiability, extensability, agility	flexiability, extensability,	4
extensibility	flexiability, extensability,	reusability, modularity, flexiability, extensability, agility	flexiability, extensability,	4
durability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
customizability	customizability	modularity, customizability	customizability	3
correctness	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
configurability	modifiability, configurability	reusability, modularity, modifiability, flexiability, extensability, configurability, agility	modifiability, configurability	3
compatibility	compatibility	compatibility	compatibility	3
availability	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
agility	agility	agility	agility	5
adaptability	daptability	reusability, modularity, flexiability, extensability, agility, adaptability	daptability	3
accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	usability,simplicity,reusability, reliability, modularity, modifiability, maintainability, flexiability, extensability, durability, correctness, configurability, compatibility, availability, agility, adaptability, accuracy	usability, reliability, maintainability, durability, correctness, availability, accuracy	2
accessibility	accessability	modularity, customizability, accessibility	accessability	2

Fig. 9. Human dimension

Step 5:

Next, the conical matrix was developed. As showed in Figures 10, 11, and12 the drive power and dependence power in each dimension are following as:

- Environment dimension: modularity has the highest drive power by 20 and resource usage has the highest dependence power by 21. Therefore, modularity as a drive power has an effect on most factors and resource usage as a dependence power was affected by most factors.
- Economic dimension: modularity has the

highest drive power by 21 and benefits has the highest dependence power by 21. Therefore, modularity as a drive power has an effect on most factors and benefits as a dependence power was affected by most factors.

Human dimension: modularity has the highest drive power by 20 and livability has the highest dependence power by 21. Therefore, modularity as a drive power has an effect on most factors and livability as a dependence power was affected by most factors.

	energy usage	resource usage	carbon footprints	waste	pollu-tion	usab-ility	upgrad-ability	simpl-icity	reusab-ility	reliabil-ity	modula-rity	modifiab-ility	maintain-ability	flexib-ility	extensib-ility	durab-ility	customiz-ability	correct-ness	configur-ability	compati-bility	avail-ability	agil-ity	adapt-ability	accur-acy	accessi-bility	Driving power
energy-usage	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
resource usage	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
carbon footprints	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
waste	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
pollution	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
usability	1*	1*	1*	1*	0	1	0	0	0	1	0	0	1*	0	0	1	0	1*	0	0	1*	0	0	1*	0	11
upgradability	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
simplicity	1*	1	1*	1*	0	1	0	1	0	1*	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1*	0	12
reusability	1*	1	1	1*	0	1*	1*	0	1	1*	0	1*	1*	1	1*	1*	0	1*	1*	0	1*	0	1*	1*	0	18
reliability	1*	1*	1*	1*	0	1*	0	0	0	1	0	0	1*	0	0	1*	0	1	0	0	1	0	0	1	0	11
modularity	1	1	1*	1*	0	1*	1	0	0	1*	1	1*	1	1	1*	1*	1	1*	1*	0	1*	0	1*	1*	1*	20
modifiability	1*	1*	1*	1	0	1	0	0	0	1	0	1	1	0	0	1*	0	1*	1	0	1*	0	0	1*	0	13
maintain-ability	1*	1*	1*	1	0	1	0	0	0	1*	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1*	1*	11
flexibility	1*	1*	1*	1	0	1	1	0	0	1	0	1	1	1	1	1*	0	1*	1	0	1*	0	1	1*	0	17
extensibility	1*	1*	1*	1	0	1	1	0	0	1*	0	1	1	1	1	1	0	1*	1	0	1*	0	1*	1*	0	17
durability	1*	1*	1*	1	0	1*	0	0	0	1	0	0	1	0	0	1	0	1*	0	0	1	0	0	1*	0	11
customiza-bility	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	3
correctness	1	1	1*	1*	0	1	0	0	0	1	0	0	1	0	0	1*	0	1	0	0	1*	0	0	1	0	11
configur-ability	1*	1*	1	1*	0	1*	0	0	0	1*	0	1	1	0	0	1*	0	1*	1	0	1*	0	0	1*	0	13
compatibility	1	1	1*	1*	1	1*	0	0	0	1*	0	0	1	0	0	1*	0	1*	0	1	1*	0	0	1*	0	13
availability	1	1	1	1*	0	1	0	0	0	1	0	0	1*	0	0	1*	0	1*	0	0	1	0	0	1*	0	11
agility	1*	1*	1	1	0	1*	1*	0	0	1*	0	1*	1*	1	1	1*	0	1*	1*	0	1*	1	1*	1*	0	18
adaptability	1*	1	1	1	0	1*	0	0	0	1	0	0	1	0	0	1*	0	1*	0	0	1*	0	1	1*	0	12
accuracy	1*	1	1	1*	0	1	0	0	0	1	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1	0	11
accessibility	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2
Dependece power	18	21	18	18	1	17	6	1	1	17	1	7	17	5	5	17	2	17	7	1	17	1	6	17	3	

Fig. 10. The drive power and dependence power of environment dimension

	assets	profit-ability	wealth creation	protect invesment	benefits	usab-ility	upgrad-ability	simpl-icity	reusab-ility	reliabil-ity	modula-rity	modifiab-ility	maintain-ability	flexib-ility	extensib-ility	durab-ility	customiz-ability	correct-ness	configur-ability	compati-bility	avail-ability	agil-ity	adapt-ability	accur-acy	accessi-bility	Driving power
assets	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
profitability	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
wealth creation	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
protect invesment	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
benefits	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
usability	1*	1*	1	1*	1	1	0	0	0	1	0	0	1*	0	0	1	0	1*	0	0	1*	0	0	1*	0	12
upgradability	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
simplicity	1*	1*	1	1*	1	1	0	1	0	1*	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1*	0	13
reusability	1*	1	1	1	1	1*	1*	0	1	1*	0	1*	1*	1	1*	1*	0	1*	1*	0	1*	0	1*	1*	0	19
reliability	1*	1*	1	1*	1	1*	0	0	0	1	0	0	1*	0	0	1*	0	1	0	0	1	0	0	1	0	12
modularity	1*	1*	1*	1*	1	1*	1	0	0	1*	1	1*	1	1	1*	1*	1	1*	1*	0	1*	0	1*	1*	1*	21
modifiability	1*	1	1	1	1	1	0	0	0	1	0	1	1	0	0	1*	0	1*	1	0	1*	0	0	1*	0	14
maintain-ability	1*	1	1*	1	1	1	0	0	0	1*	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1*	0	12
flexibility	1	1	1*	1	1	1	1	0	0	1	0	1	1	1	1	1*	0	1*	1	0	1*	0	1	1*	0	18
extensibility	1*	1	1*	1	1	1	1	0	0	1*	0	1	1	1	1	1	0	1*	1	0	1*	0	1*	1*	0	18
durability	1	1	1	1	1	1*	0	0	0	1	0	0	1	0	0	1	0	1*	0	0	1	0	0	1*	0	12
customiza-bility	0	1*	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	6
correctness	1*	1	1*	1	1	1	0	0	0	1	0	0	1	0	0	1*	0	1	0	0	1*	0	0	1	0	12
configur-ability	1*	1	1*	1*	1	1*	0	0	0	1*	0	1	1	0	0	1*	0	1*	1	0	1*	0	0	1*	0	14
compatibility	1*	1	1*	1	1	1*	0	0	0	1*	0	0	1	0	0	1*	0	1*	0	1	1*	0	0	1*	0	13
availability	1	1	1*	1	1	1	0	0	0	1	0	0	1*	0	0	1*	0	1*	0	0	1	0	0	1*	0	12
agility	1	1	1	1	1*	1*	1*	0	0	1*	0	1*	1*	1	1	1*	0	1*	1*	0	1*	1	1*	1*	0	19
adaptability	1*	1	1	1	1	1*	0	0	0	1	0	0	1	0	0	1*	0	1*	0	0	1*	0	1	1*	0	13
accuracy	1*	1	1*	1	1	1	0	0	0	1	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1	0	12
accessibility	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4
Dependece power	18	20	20	20	21	17	6	1	1	17	1	7	17	5	5	17	2	17	7	1	17	1	6	17	3	

Fig. 11. The drive power and dependence power of economic dimension

	mental well-being	social well-being	educa-tion	freed-om	livabi-lity	usab-ility	upgrad-ability	simpl-icity	reusab-ility	reliabil-ity	modula-rity	modifiab-ility	maintain-ability	flexib-ility	extensib-ility	durab-ility	customiz-ability	correct-ness	configur-ability	compati-bility	avail-ability	agil-ity	adapt-ability	accur-acy	accessi-bility	Driving power
mental well-being	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
social well-being	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
education	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
freedom	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
livability	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
usability	1	1	0	1*	1	1	0	0	0	1	0	0	1*	0	0	1	0	1*	0	0	1*	0	0	1*	0	11
upgradability	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
simplicity	1	1*	0	1*	1	1	0	1	0	1*	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1*	0	12
reusability	1*	1*	0	1*	1	1*	1*	0	1	1*	0	1*	1*	1	1*	1*	0	1*	1*	0	1*	0	1*	1*	0	18
reliability	1*	1	0	1*	1	1*	0	0	0	1	0	0	1*	0	0	1*	0	1	0	0	1	0	0	1	0	11
modularity	1*	1*	0	1*	1	1*	1	0	0	1*	1	1*	1	1	1*	1*	1	1*	1*	0	1*	0	1*	1*	1*	20
modifiability	1	1	0	1	1	1	0	0	0	1	0	1	1	0	0	1*	0	1*	1	0	1*	0	0	1*	0	13
maintain-ability	1	1	0	1	1	1	0	0	0	1*	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1*	0	11
flexibility	1	1	0	1	1	1	1	0	0	1	0	1	1	1	1	1*	0	1*	1	0	1*	0	1	1*	0	17
extensibility	1	1*	0	1	1	1	1	0	0	1*	0	1	1	1	1	1	0	1*	1	0	1*	0	1*	1*	0	17
durability	1	1	0	1	1	1*	0	0	0	1	0	0	1	0	0	1	0	1*	0	0	1	0	0	1*	0	11
customiza-bility	1*	1*	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	6
correctness	1	1	0	1*	1	1	0	0	0	1	0	0	1	0	0	1*	0	1	0	0	1*	0	0	1	0	11
configur-ability	1	1	0	1	1	1*	0	0	0	1*	0	1	1	0	0	1*	0	1*	1	0	1*	0	0	1*	0	13
compatibility	1	1	0	1*	1	1*	0	0	0	1*	0	0	1	0	0	1*	0	1*	0	1	1*	0	0	1*	0	12
availability	1	1	0	1*	1	1	0	0	0	1	0	0	1*	0	0	1*	0	1*	0	0	1	0	0	1*	0	11
agility	1	1*	0	1*	1	1*	1*	0	0	1*	0	1*	1*	1	1	1*	0	1*	1*	0	1*	1	1*	1*	0	18
adaptability	1	1*	0	1	1	1*	0	0	0	1	0	0	1	0	0	1*	0	1*	0	0	1*	0	1	1*	0	12
accuracy	1	1*	0	1*	1	1	0	0	0	1	0	0	1	0	0	1*	0	1*	0	0	1*	0	0	1	0	11
accessibility	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5
Dependece power	20	20	1	20	21	17	6	1	1	17	1	7	17	5	5	17	2	17	7	1	17	1	6	17	3	

Fig. 12. The drive power and dependence power of human dimension

Step 6:
Finally, ISM preliminary model is developed. the factors divided in to five levels (Figures 13, 14, and 15). The top-level factors have no effect on other factors in other words this factor have high dependence power, on the other hand the low-level factors have high drive power and have an effect on other factor, in addition they play an important role than other factors.
In environment dimension, modularity, agility and reusability placed in the lowest level with the highest driving power and energy usage, resource usage, carbon footprint, waste and pollution placed in the top level that indicates these factors have the highest dependence power.
In economic dimension, modularity, agility and reusability placed in the lowest level with the highest driving assets, profitability, wealth creation, protect investment and benefits placed in the top level that indicates these factors have the highest dependence power.
In human dimension, modularity, agility and reusability placed in the lowest level with the highest driving assets, mental well-being, social well-being, education, freedom, livability and placed in the top level that indicates these factors have the highest dependence power.

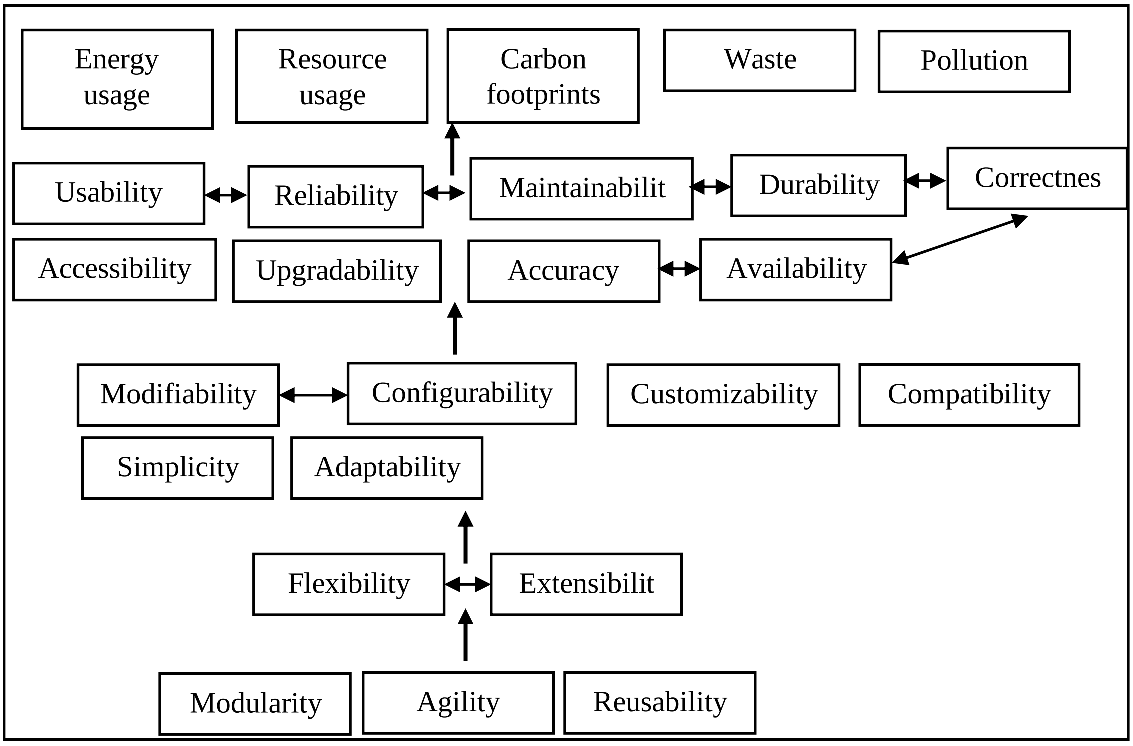

Fig. 13. The ISM preliminary model of environment dimension

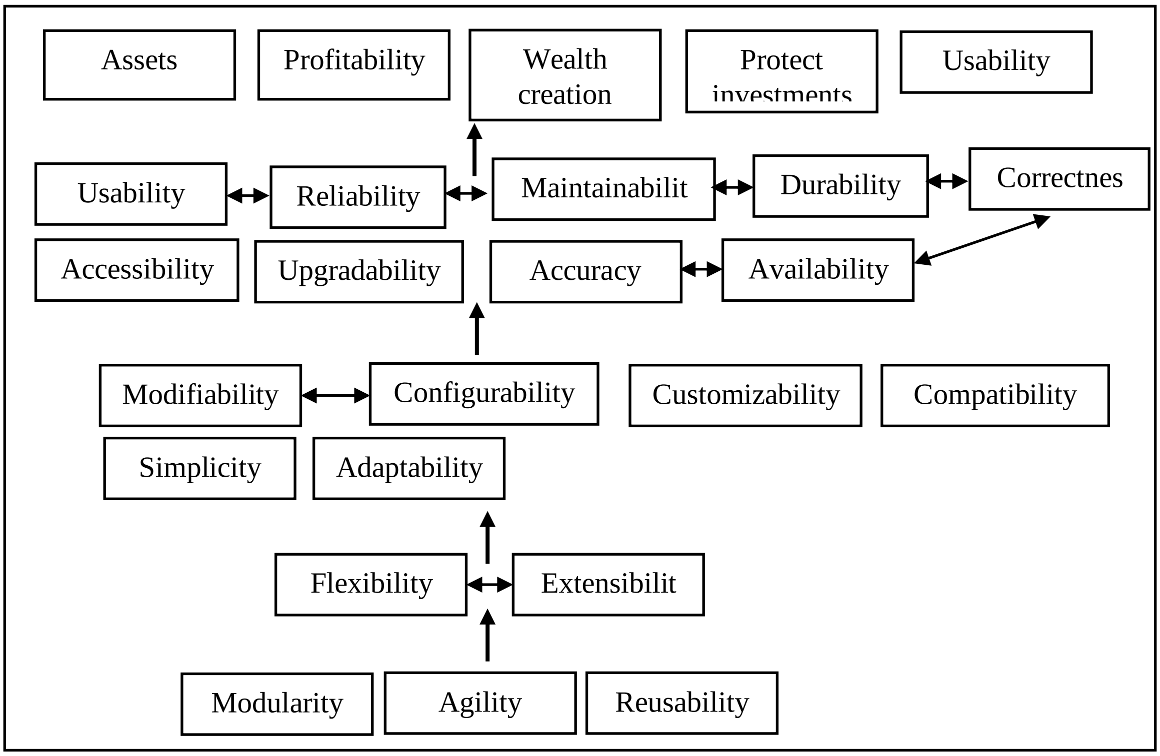

Fig. 14. The ISM preliminary model of economic dimension

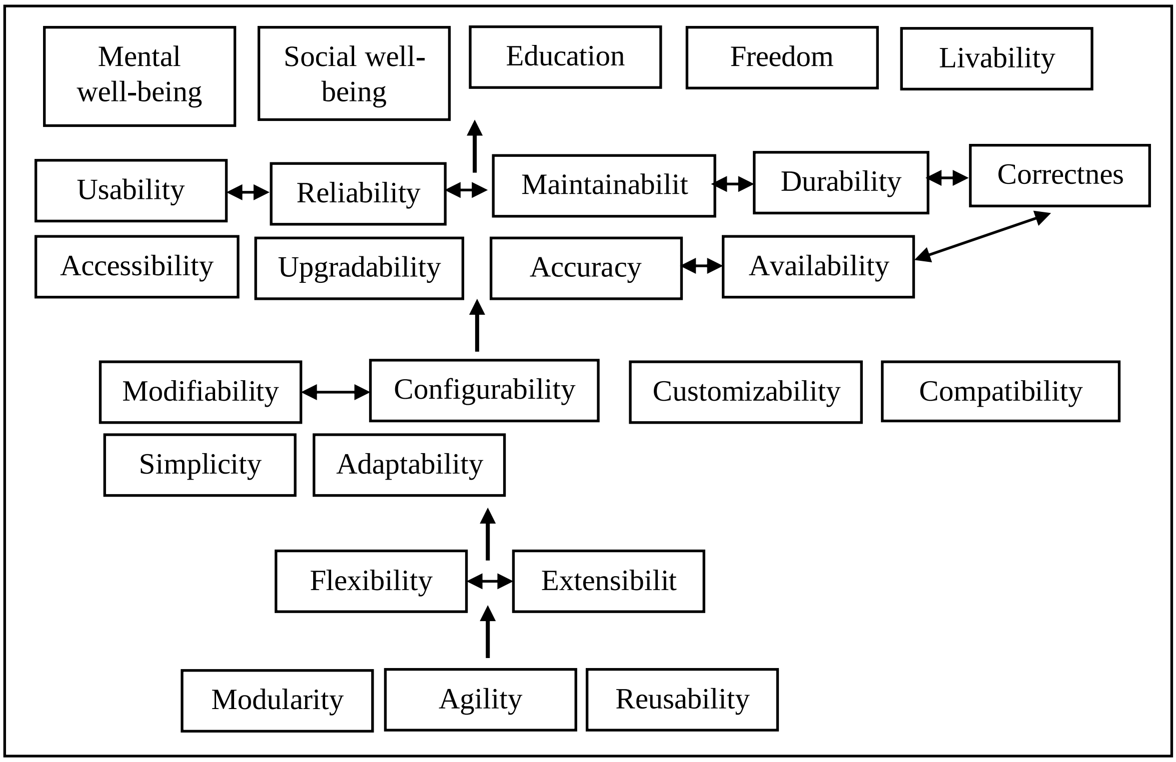

Fig. 15. The ISM preliminary model of human dimension

Then, drive power and dependence power of factors are analyzed by MICMAC analysis.
Figures 16, 17 and 18 show the MICMAC analysis in three dimension of software sustainability. In environment dimension:

1) The autonomous factors that relatively disconnected from the system are: pollution, upgradability, simplicity, modifiability, customizability, configurability, compatibity, adaptability and accessibility.

2) The dependent factor that have strong dependence power and their drive power is less than the other factors are energy usage, resource usage, carbon footprints, waste, usability,

reliability, maintainability, durability, correctness, availability and accuracy.

3) The independent factors are reusability, modularity, flexibility, extensibility and agility. These are the most influential factors in environmental dimension and considered as a key factor.

In economic dimension:

1) The autonomous factors that relatively disconnected from the system are upgradability, simplicity, customizability, compatibity, adaptability and accessibility.

2) The dependent factor that have strong dependence power and other factors have an effect on them are: assets, profitability, wealth creation, protect investment, benefits, usability, reliability, maintainability, durability, correctness, availability and accuracy.

3) The independent factors are reusability, modularity, modifiability, flexibility, extensibility, configurability and agility. These are the most influential factors in economic dimension and considered as a key factor.

In human dimension:

1) The autonomous factors that relatively disconnected from the system are: education, upgradability, simplicity, modifiability, customizability, configurability, compatibity, adaptability and accessibility.

2) The dependent factor that have strong dependence power and other factors have an effect on them are mental well-being, social well-being, freedom, livability, usability, reliability, maintainability, durability, correctness, availability and accuracy.

3) The independent factors are: reusability, modularity, flexibility, extensibility and agility. These are the most influential factors in environmental dimension and considered as a key factor.

In all three dimensions, modularity has strong drive power so it has effect in environment, economic and human dimension and should be considered in all three dimensions of sustainability in order to develop green and sustainable software.

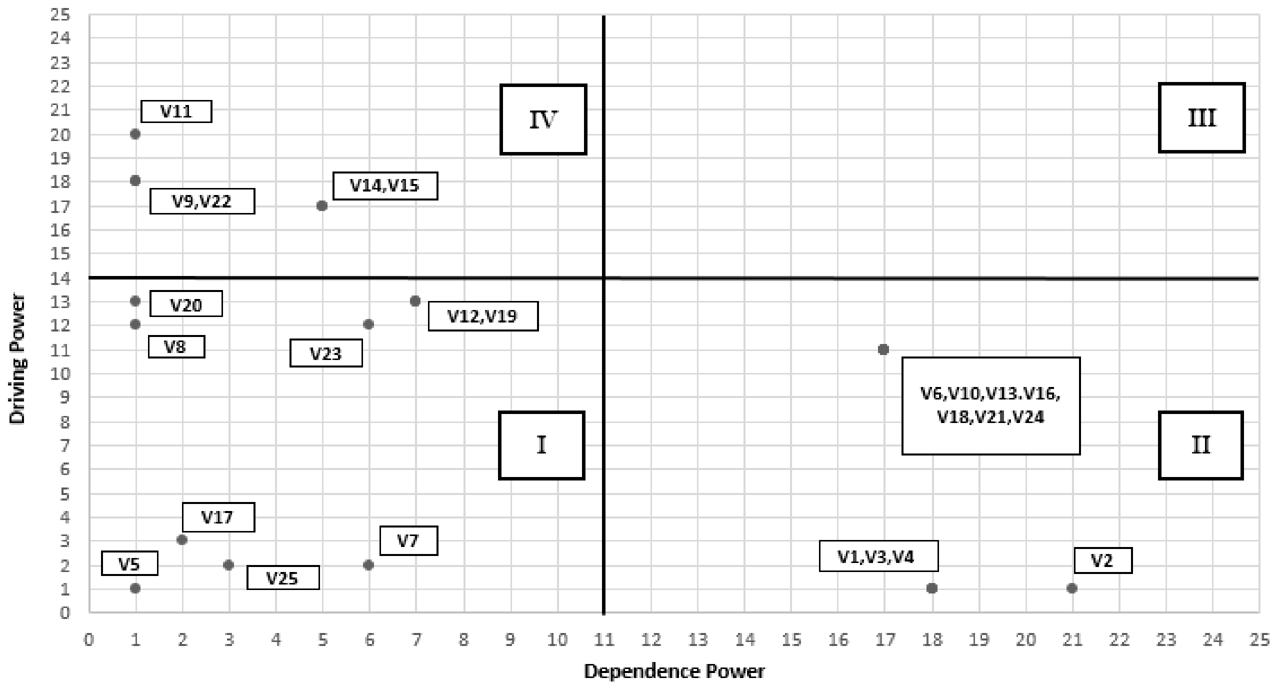

Fig. 16. Environment dimension MICMAC

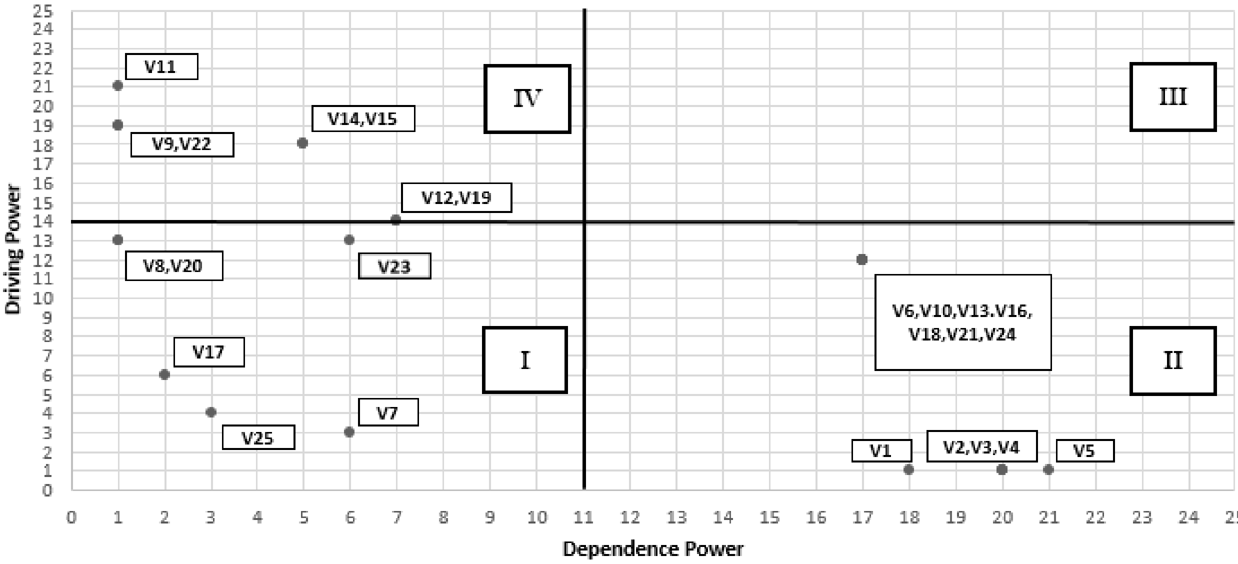

Fig. 17. Economic dimension MICMAC

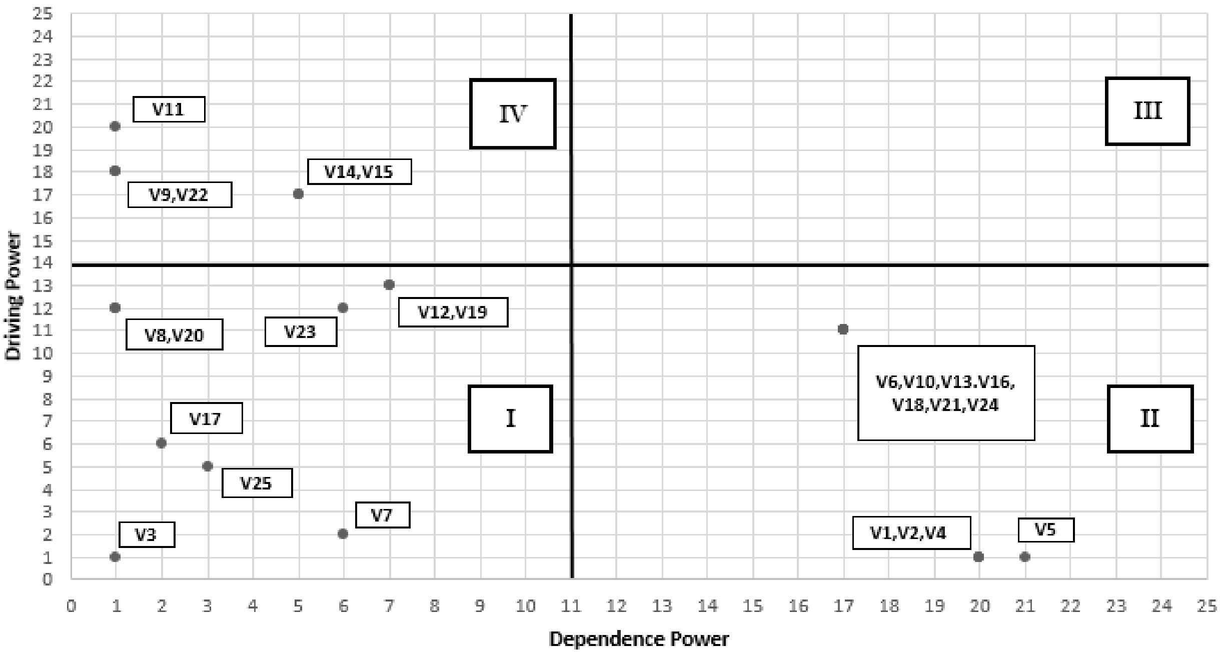

Fig. 18. Human dimension MICMAC

5. Discussion

In this paper, three dimension of sustainability was determined. It was specified the common non-functional by using Delphi. Then, it was identified the non-functional requirement which has effect on developing green and sustainable software in environment, economic and human dimension by using ISM. ISM has ability to analyses the link between factors that was identified for a problem and also can identified the effect of each factor that have on each other. Other researchers try to describe software sustainability as a new software quality in 21th century and develop tools to measure software energy consumption to determine which one consume less energy than the other so it could be called green software. But, the attempt of this paper was to answer this important following question how software developer can develop sustainable and green software? What software qualities make it sustainable and green? Answering these questions is a great step to have sustainable future and green environment.

The study shows that in all three-software sustainability dimension modularity, agility and reusability following them flexibility and extensibility were specified as a software non-functional requirement that more effective than other non-functional requirements on software sustainability. It implies that software developers can save time, money, staff effort, energy and prevent producing carbon footprints, e-waste and pollution during software development, software performance and software maintenance which can lead to green software development. Also, they can develop a software which can endure for a long-time despite of users' requirements changes and hardware change which can lead to sustainable software development. Specialist in software engineering can rate the software based on the qualities that determined in this paper and eco-labeled in order to specified whether the software is sustainable and green or not. By eco-labeling software those users who don't have special knowledge in the realm of software can choose sustainable and green software with ease. In addition, it could be an encouragement for developers to develop software in green and sustainable way to gain eco-label for their software.

In the future, more software non-functional requirement effect on developing green and sustainable software should be investigated in order to have comprehensive software qualities and guideline for software developers to develop green and sustainable one and also for eco-labeling software based on these non-functional requirements.

6. Conclusions

With ever growing energy consumption and global warming due to increase in use of software in everyday of individual's lives and industries, researchers in software development and also software engineering have become aware of the significant role of software in our future sustainability and having green environment. So, greenness and sustainability gain importance as a software non-functional requirement in 21th century. In spite of the importance of this software non-functional, there is misunderstanding of how the sustainable and green software can be developed. However, sustainability identified as software quality but other software qualities have an important effect on developing green and sustainable software.

The objective of this paper is to help software developers to develop sustainable software and green software so that we will have sustainable future and also green environment. To do that we identified the software non-functional requirements that have an effect on software sustainability and software greenness. First, we selected 20 software non-functional requirements out of 60 most important software non-functional requirements that were chosen by Delphi method. Then, we considered five items for each software sustainability dimensions, so by 25 factors we analysis the effect of these factors in software sustainability by ISM. As a result, the most effective factors in software sustainability and greenness were identified as following: reusability, modularity, flexibility, extensibility and agility. These factors should be considered in software development in order to develop green software and sustainable software in environment, economic and human dimension. Modularity was identified as key factor in all three dimensions.

This result could be step forward to eco-labeling software. It could be allocated score to software base on testing software in order to determine which these 20 software non-functional requirements it has, according to their priority. Eco-labeled software would be a motivation for software developers to care about the sustainability and greenness of their software, also would help users and organizations to choose the sustainable and green one.

As for future research directions, one can study the impact of sustainability factors on different software or even different operating systems; another direction can be analyzing the greenness of the software on the human interactions with the process of software development.

References

[1] Venters CC, Capilla R, Betz S, Penzenstadler B, Crick T, Crouch S, et al. Software sustainability: Research and practice from a software architecture viewpoint. Journal of Systems and Software. Vol. 138, (2018), pp. 174-88.

[2] Beghoura MA, Boubetra A, Boukerram A. Green software requirements and measurement: random decision forests-based software energy consumption profiling. Requirements Engineering. Vol. 22, No. 1, (2017), pp. 27-40.

[3] Capra E, Francalanci C, Slaughter SA. Measuring Application Software Energy Efficiency. IT Professional. Vol. 14, No. 2, (2012), pp. 54-61.

[4] Hindle A, editor Green Software Engineering: The Curse of Methodology. 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER); (2016), pp. 14-18.

[5] Sabharwal M, Agrawal A, Metri G. Enabling Green IT through Energy-Aware Software. IT Professional. Vol. 15, No. 1, (2013), pp. 19-27.

[6] Tiako PP, editor Conceptual software infrastructure for sustainable development. 2004 IEEE International Engineering Management Conference (IEEE Cat No04CH37574); (2004), pp. 18-21.

[7] Fenton NE. Software metrics: a rigorous approach. (1991).

[8] Allen A, Aragon C, Becker C, Carver J, Chis A, Combemale B, et al. Engineering Academic Software (Dagstuhl Perspectives Workshop 16252). Dagstuhl Manifestos. 6: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik; (2017).

[9] Lutz R, Weiss D, Krishnan S, Yang J, editors. Software product line engineering for long-lived, sustainable systems. International Conference on Software Product Lines; (2010), Springer.

[10] Dick M, Drangmeister J, Kern E, Naumann S, editors. Green software engineering with agile methods. 2013 2nd International Workshop on Green and Sustainable Software (GREENS); (2013), pp. 20-20.

[11] Venters CC, Seyff N, Becker C, Betz S, Chitchyan R, Duboc L, et al., editors. Characterising sustainability requirements: A new species red herring or just an odd fish? 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS); (2017), IEEE.

[12] Condori-Fernandez N, Procaccianti G, Ali N, editors. Metrics for Green and Sustainable Software: MeGSuS 2014. 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement; (2014), pp. 6-8.

[13] Calero C, Piattini M. Puzzling out Software Sustainability. Sustainable Computing: Informatics and Systems. Vol. 16, (2017), pp. 117-24.

[14] Penzenstadler B, Raturi A, Richardson D, Tomlinson B. Safety, Security, Now Sustainability: The Nonfunctional Requirement for the 21st Century. IEEE Software. Vol. 31, No. 3, (2014), pp. 40-7.

[15] Garcia-Mireles GA, editor Exploring sustainability from the software quality model perspective. 2018 13th Iberian Conference on Information Systems and Technologies (CISTI); (2018), pp. 13-16.

[16] Salam M, Khan SU, editors. Developing green and sustainable software: Success factors for vendors. 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS); (2016), pp. 26-28.

[17] Shenoy SS, Eeratta R, editors. Green software development model: An approach towards sustainable software development. 2011 Annual IEEE India Conference; (2011), pp. 16-18.

[18] Sommerville I, Sawyer P. Requirements engineering: a good practice guide: John Wiley & Sons, Inc.; (1997).

[19] Kotonya G, Sommerville I. Requirements engineering: processes and techniques: Wiley Publishing; (1998).

[20] Linstone HA, Turoff M. The delphi method: Addison-Wesley Reading, MA; (1975).

[21] Okoli C, Pawlowski SD. The Delphi method as a research tool: an example, design considerations and applications. Information & Management. Vol. 42, No. 1, (2004), pp. 15-29.

[22] Attri R, Dev N, Sharma V. Interpretive structural modelling (ISM) approach: an overview. Research Journal of Management Sciences. Vol. 2319, (2013), p. 1171.

Follow this article at the following site:

Zahra Taherikhonakdar & Hamed Fazlollahtabar. Integrated Delphi-Interpretive Structural Modeling for Sustainable Green Software Development. IJIEPR 2024; 35 (2) :1-26
URL: http://ijiepr.iust.ac.ir/article-1-1889-en.html