Exploring Factors Affecting the Adoption of Green Process Management Model in the Software Industry: Progress Towards Sustainability and Circularity

Changiz Valmohammadi and Farkhondeh Mortaz Hejri

1 Introduction

In response to global warming and rising energy costs, governmental agencies and private enterprises have expanded their environmental protection investments (Alam et al., 2022; Marquis et al., 2011). As a result, a great global movement has emerged to use information and communication technology (ICT) in an environmentally responsible manner. The title "Green Information and Communication Technology" was chosen to signify the usage of ICT to reduce energy consumption

C. Valmohammadi (✉)
Department of Industrial Management, Faculty of Management, South Tehran Branch, Islamic Azad University, Tehran, Iran
e-mail: valmohammadi@yahoo.com

F. M. Hejri
Department of IT Management, Faculty of Management, South Tehran Branch, Islamic Azad University, Tehran, Iran

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

M. O. Erdiaw-Kwasie and G. M. M. Alam (eds.), *Circular Economy Strategies and the UN Sustainable Development Goals*, Sustainable Development Goals Series,
https://doi.org/10.1007/978-981-99-3083-8_16

and waste in the environment. In addition, it indicates the manufacturing of green hardware devices and components, cutting-edge techniques for building energy-efficient software, and a collection of tools to help attain the primary goal of "saving the world" (Garcia et al., 2016). Every aspect of life, including new technology, has become environmentally conscious. Mankind recognises the importance of reducing energy consumption and carbon footprint. On a global scale, innovations are being developed to address these issues. Significant research and development programmes and substantial budgeting for projects to develop eco-friendly technologies demonstrate the global focus on this issue (Calero et al., 2019). Users are increasingly utilising software applications due to the growing number of mobile devices, including smartphones, tablets, and laptops, as well as internet users, high accessibility provided by cloud computing, and the Internet of Things (IoT). Accordingly, the impact of software development on greenhouse gas emissions has increased. In the meantime, engineering equipment, procedures, and IT services harm the environment due to their high energy consumption (Anthony et al., 2018).

As demands for corporate software increased, it has become a significant concern in recent years. Green software development is a relatively new research area in green computing, which requires more attention from academicians (Calero et al., 2022; Yahaya et al., 2019). Technologies, such as software and applications used by the community, and their usage are developing more than ever, which affect modern human lifestyle. Besides, technological signs of progress increased energy consumption rates (Mishra & Mishra, 2021). Manotas et al. (2016) argue that there are two main problems, including the lack of need for more knowledge and tools for developing energy-efficient software. From a technical point of view, examples of successful tools and experiences are available. However, not all of these tools have yet been presented in a consistent and integrated framework and cannot provide an integrated view for software developers and designers. Furthermore, the study of the relationship between software features and natural resources demand stemming from the structure and use of ICT systems has not received much attention so far (Alam et al., 2021; Hilty & Aebischer, 2015). Previous studies have focused more on the greenness of software products, and less attention has been paid to the green features in software processes (Ibrahim, 2021). Therefore, this study aims to explore how identified factors impact the green software development processes to facilitate the progress towards enhancing the suitability and circular economy of the software industry using the interpretative structural modelling (ISM) approach in the context of Iran.

2 Context

Sustainability has become the foundation of IT; it commits that IT will support our future and that the industry will improve. Using IT to create more eco-friendly systems has been considered to reduce energy consumption (greening of IT). As Calero et al. (2019), and Calero and Piattini (2017) mention on the hard side, such as green data centres, green hardware, etc., necessary attention has been made, so now it is necessary to pay careful attention to the soft (software-related) aspect of the issue. Green software engineering is an important aspect of the software engineering process in this era. Previously, software engineers focused on hardware or software development, regardless of sustainability. Duboc et al. (2019) believe that engineers focus more on the technical aspects of a software system, and any concerns beyond the rapid release of software are usually less priority. Sustainability is important for all software systems because using each new system depends on becoming part of the technical infrastructure. Its frequent usage may cause new burdens on social ecosystems (Raisian et al., 2016; Saputri & Lee, 2021). As hardware becomes more powerful, the impact of software behaviour on energy consumption increases significantly (Calero et al., 2022).

The development of integrated software systems has made daily human life highly dependent on technology and increased energy consumption (Saputri & Lee, 2021). Sustainability is an important issue for future jobs and especially for software engineers, because of its impact on society (Mishra & Mishra, 2021). Green business process management includes a sustainable approach to managing business processes, which can help to create and undertake environmentally sustainable operations (Maciel, 2017). In different industries, the importance of business processes with a green approach has been considered, and by defining the corresponding business strategies, the first steps have been taken to improve the environmental impact on processes. However, there are still no clear solutions for achieving the green stage and sustainability in the software development life cycle (Raisian et al., 2016).

In most cases, the green process of IT has led to improved resource efficiency. Reducing costs in the company's structure and strengthening its competitive position compared to competitors and organisations provide new services in today's society and unique benefits for current products, leading to a better image of the company compared to its competition. This will increase the worthiness of the organisation and thus increase efficiency (Alam et al., 2022; Garcia et al., 2016).

According to new technologies, green software can fit the software development process in complex software settings where software systems are green (Jannat, 2016). Therefore, developing concepts, methods, and tools that lessen environmental software impact, software products, and resulting processes is essential. However, using energy-efficient software development tools and methodologies has rarely been explored. Furthermore, because of Not bof the abundance of energy-efficient low-priced hardware components, designers have not paid enough attention to the energy-saving potentials of software (Mahmoud & Ahmad, 2013).

3 Literature Review

3.1 Business Process

Process management has gained increasing importance in various economies. The importance of well-designed processes in optimising limited resources is well established. In addition, processes can be redesigned to use available resources better. However, to transition to a sustainable economy, we must learn how to redesign many aspects of modern life, using business process management as a key tool. Therefore, we require a set of appropriate processes to apply business process management to sustainability issues (Vom Brocke et al., 2012). Furthermore, business practices are a subset of business processes in most organisations, being in place to help create value in the form of profits, credits, and other incentives. Therefore, organisations' appearance, performance, and utility are naturally described by their design and implementation of such processes (Seidel et al., 2012).

Thus far, business process management has not seriously considered environmental sustainability as a change driver or a goal. Thus, the present understanding of business process management must be developed to incorporate the notion of "green business process management" to encompass sustainability as a goal and a catalyst for change in business process management. Furthermore, to better define, implement, and use green business process management in organisational activities consistent with sustainability goals, we need to know how to describe, implement, and apply green business process management in organisational actions. Accordingly, green business process management is all about understanding, documenting, modelling, analysing, simulating, executing, and constantly changing the process with a focus on the environmental effects of these processes (Seidel et al., 2012).

3.2 Software Development Process

The software development process, commonly referred to as the software development lifecycle (SDLC) methodology or model, is a framework for organising, planning, and controlling the development of information systems. Over time, various frameworks have emerged, each with strengths and weaknesses (Spenser, 2010). Software development is the process of writing and maintaining codes. In addition, it encompasses all activities ranging from the first concept of the intended software to its final version, which sometimes occurs in a planned and systematic manner. Hence, software development can encompass any actions resulting in software products, including research, new developments, prototyping, modifications, reuses, reengineering, or maintenance.

This process aims to satisfy user needs while ensuring proper system performance. Thus, it must include procedures for validation (outputs meeting requirements) and verifiability (output performance being accurate) purposes. Manuals, programme documentation, and test items are part of the materials produced throughout the development process, together with computer programmes. For example, directions for software use are detailed in user manuals. In addition, test cases and programme documentation help developers become familiar with the future maintenance and deployment of software (Pressman & Maxim, 2016). Suppose there is no precise software process for an individual or organisation, in that case, it will be defined in terms of the ongoing project, organisational culture, and competencies of individuals involved in the project. Software development is often confused with programming. However, software development is concerned with the commercial production of software, optimal use of existing components, and strategies for organising, expediting, and improving the quality of software projects (Birrell, 1985). The environmental impacts of software are categorised into three categories. The first level includes the ICT supply effects which directly impact the use of energy or natural resources, such as hardware needs, implementation and performance requirements, software product packaging, network bandwidths, etc. Impacts resulting from using software services are referred to as second-level impacts or impacts of information technology use. Third-level impacts, also known as systematic ICT impacts, result from multiple conflicting systems interacting to lead to reactive consequences. For example, if some natural resources are utilised to manufacture a specific type of software, but the same number of resources can be used to produce more diversified software, demands for these natural resources will increase (Naumann et al., 2011).

3.3 Green Software

Sustainable software engineering (SSE), or green software engineering (GSE), has gained prominence in the software engineering community over the past few years. This is because many academicians know the direct and indirect effects of software programmes on the system and environmental energy consumption. As a result, software engineering methodologies are being developed, with green software developments being promoted by software engineers and developers. The major goal of these endeavours is to be green and sustainable at every level of the software development process; furthermore, their other goal is to consider it as a software quality feature or a non-functional aspect of software programmes.

Many research projects on green computing use the term "sustainability" interchangeably with "greenability". Software engineering sustainability, for instance, was considered to be a part of software sustainability in a book written by Calero and Piattini (2017). According to these authors, software sustainability can be applied to various items, including software systems, software products, web applications, data centres, etc. The term "sustainable software" has been defined in two ways. First, software-assisted greening has been around for a while now. Software greening generally refers to environmental activities that any software addresses, including energy management software. In addition, green software refers to how software is developed to produce a more sustainable product (called GSE). Engineering principles are applied to environmental issues using GSE methods. Accordingly, the software is developed, operated, and maintained in a green manner, resulting in green software.

Each step of the SDLC is supported and performed by specified activities. However, according to a study by Shenoy and Eeratta (2011), SDLC implementation may currently cause issues with green environment upgrades. The problem is that several major decisions made at these stages, such as paper use, e-waste production, electricity use, increased carbon effects of transportation and travel, air conditioning, and the like, may directly or indirectly harm the environment. Thus, human, economic, and energy resources are required for operations in the software life cycle. These factors will define the aspects of software stability (Calero & Piattini, 2017). In addition, software development and maintenance endeavours affect a software development community sociological and psychological features and its people. This category addresses labour rights, mental health, social protection, social fairness, and viability, among other topics. Conversely, economic stability is determined by how the software life cycle process safeguards stakeholder investment, ensures profitability, lowers risks, and preserves assets. Finally, environmental sustainability is defined as the impact of a software product's development, maintenance, and use on energy consumption and resource use (Calero & Piattini, 2017).

According to Jetley et al. (2013), the software industry needs to incorporate top-notch software engineering methodologies into its software development strategy to improve quality and costs. However, this requires a comprehensive set of software industry processes and tools and software engineering methods, tools, and techniques that are limited thus far. This paper discusses some of the issues the software industry faces when applying software engineering concepts and procedures to application development. Furthermore, this research stresses the need for more research and initiatives to promote the software industry in adapting software engineering, methodologies and concepts.

Dick et al. (2013) used agile methodologies in their GSE research. Accordingly, they provided a methodology integrating green computing aspects of software engineering approaches into agile green methods. In addition, they introduced a model that incorporated green computing features into software engineering methods and agile methods for developing green and sustainable software. Ardito and Morisio (2014) surveyed information system guidelines and data to see how they would reduce energy consumption. The authors, for instance, provided energy efficiency guidelines for infrastructures, applications, operating systems, hardware, and networks. Finally, Betz and Caporale (2014) focused on the design of sustainable software systems. Accordingly, they considered sustainability management as one of the most pressing issues today, which explained why private and public organisations were so interested in "sustainable" practices and solutions. In addition, their research highlighted the scarcity of existing sustainable development methods and solutions. Besides, the authors introduced a conceptual model for integrating sustainability features into business development processes to accomplish this goal. Furthermore, this study suggested that stability needs to be measured during the SDLC to integrate stability features into software engineering.

Lago et al. (2015) discussed the opportunities and challenges of sustainable software development. Software, according to these authors, is critical to societal success. Thus, environmental sustainability has become an integral component of software system operation and development. According to Lago et al. (2015), key challenges in producing green and sustainable software include software power shortages, green software, supervision of software energy consumption in practice, sustainable architectural design, green software, and cloud-based ready-made software. According to Pinto and Castor (2017), green software is concerned with lowering energy consumption through software analysis and optimisation. Accordingly, they added that "even tiny inefficiencies in applications compound throughout the system, with a major impact on battery life". Yahaya et al. (2019) conducted an empirical study in Malaysia on the software process regarding stability factors contributing to the development of green software. Green IT is becoming more prevalent in GSE, according to evidence. Besides, green software design and implementation are demanding and challenging from a managerial standpoint, including complicated equations between several actors. Although numerous tools and methods are available, they have not yet been well integrated into a single framework to provide software developers and designers with a unified perspective. Against this background, the present study introduced a new model for green software development from a process perspective.

In Iran, research has been done on green information technology, and the importance of this issue and effective factors in the country have been studied. Green management has also been considered in various industries. Taghva et al. (2019) provided a model for developing organisational sustainability through green information technology. The results of their research indicate that the dimensions of green information technology are as follows: green readiness, information and communication technology, as an empowerment (low carbon), green actions and activities, green information technology cycle management, green organisations and data centres, green information technology monitoring that can be used for the sustainability of organisations. Khadivar et al. (2018) presented a dynamic system model to study the component relations and determine the appropriate strategy to reduce energy consumption and eventually increase the level of green information technology maturity in the organisation. This study aims to analyse a complex system focused on reducing energy consumption and thus increasing the maturity of green information technology and then modelling its dynamic behaviour using Vensim software. Also, Shahbandarzadeh and Kabgani (2016) state that the level of acceptance and movement towards using green information technology among industries and industry managers is gradual. And budget constraints put managers under pressure to invest smartly in green information technology.

4 Circular Economy

The concept of circular economy (CE) as a potential guide towards a sustainable economic system has been noticed by industry and academia. The circular economy, like ICT systems, affects all aspects of modernity. Cutting-edge technologies such as big data, cloud computing, cyber-physical systems, the internet of things, virtual and augmented reality, and blockchain have an important role in accepting the concepts and strategies of the circular economy in the direction of reduction, reuse, recycling, and restoration (Demestichas & Daskalakis, 2020). The results show that solutions related to data collection and analysis, particularly to the Internet of things (IoT), blockchain, digital platforms, artificial intelligence algorithms, and software tools, are among the most popular solutions proposed by researchers. Results also suggest that greater emphasis is placed on the "reduce" component of the CE, although ICT solutions for the other "R" components, as well as holistic ICT-based solutions, do exist as well.

Specific important challenges impeding the adoption of ICT solutions for the CE are also identified and reviewed, with consumer and business attitudes, economic costs, possible environmental impacts, lack of education around the CE, and the lack of familiarisation with modern technologies being found among the most prominent ones (Demestichas & Daskalakis, 2020). The software has become an indispensable part of industrial production and thus influences the life cycle of manufacturing systems, as many of these systems have to be replaced or evolved due to changing requirements. Software adaptation through continuous evolution extends the service time of these systems and thus saves valuable resources (Kutscher et al., 2020). An alternative is reengineering the existing software, which enables the recycling of systems but causes major challenges. First, the existing software must be analysed efficiently, and only knowledge implicitly present in the source code must be obtained. Second, the modified software needs to be functionally verified and validated. Finally, it must be configured and commissioned on the production system. The software has an end-of-life. This ageing process results from environmental transformations (Grottke et al., 2008). In industry, this change is induced by new technologies, business models, and changing requirements, i.e. Industry 4.0 (Anderl, 2015). Although, since the 1990s, Product Service Systems (PSS) have been heralded as one of the most effective instruments for moving society towards a resource-efficient, circular economy (Tukker, 2015), PSS has also become embedded in a wider range of science fields such as manufacturing, ICT, business management, and design.

One definition of PSS is 'a mix of tangible products and intangible services designed and combined to fulfile the customer needs jointly (Tukker & Tischner, 2006). From the mid-1990s, PSS became a popular subject for researchers engaged with sustainability and business alike. Experts in the realm of sustainability argue that if one focused on the end user's needs or the service a user wants rather than the product, it would become much easier to design need-fulfilment systems with radically lower impacts. In a product-oriented business model, firms are incentivised to maximise the number of products sold. This is their principal method of boosting turnover, increasing market share, and generating profits. However, in service-oriented business models, in theory, the incentive differs. Firms then make money, by offering services and consumables, which are key in creating the cost factors. Hence, firms will be incentivised to prolong the service life of products, to ensure that they are used efficiently and to reuse parts as far as possible after the end of the product's life cycle. These elements could lead to a depreciation of material flows in the economy while maximising service output or user satisfaction.

In line with realising the global transformation towards digitising businesses and smart initiatives, automation, and robotisation and given the intensive competitive business environment, manufacturing organisations are shifting from traditional product-based business models towards developing and implementing service-oriented business models (Martin et al., 2018).

5 Methodology

5.1 Statistical Population and Sample

The statistical population of this study consisted of academic and industry specialists. Playing a key role, as identified by others, theoretical understating of the subject, diversity, and agreement to participate were the criteria for selecting participants in this study. Accordingly, significant persons were identified in the disciplines of information systems, business processes, networking and hardware, and business strategy for this study. These individuals had an important role in organisational software engineering and information technology. Besides, academic specialists were among the selected participants as they were familiar with the research model concepts and dimensions. The industry experts included senior and middle managers, consultants, and university professors. Additionally, people of various levels of knowledge, experience, and organisational affiliations were employed in this study. The interviewees were contacted before the interviews to ensure that everything was in place. Table 1 shows the profile of the respondents.

5.2 Data Collection Methods and Tools

Desk research was used to gather information about the literature and research background. Besides, semi-structured interviews were conducted to gather experimental data. The interviewees were directly contacted using this tool, which allowed for a more in-depth assessment of their perceptions, attitudes, interests, and aspirations. Interviewing is used to delve into complex issues, follow up on answers, determine root causes, and ensure that subjects comprehend questions. In in-depth semi-structured interviews, interviewees are free to describe how they perceive, act, and behave as much as possible (Kallio et al., 2016). The interview

Table 1 Profile of the respondents

Characteristics	*Frequency*	*%*
Education		
Bachelor	2	20
Master	5	50
Ph.D.	3	30
Major		
Information Technology	3	30
computer	4	40
other	3	30
Work Experience		
5–10	3	30
10–15	3	30
15–20	2	20
Up to 20	2	20

questions were extracted from the first stage of the theoretical framework. Table 2 presents the 11 factors extracted from the extant literature.

5.2.1 The Interpretive Structural Modelling Approach

The theory of ISM is based on discrete mathematics, graph theory, social sciences, group decision-making, and computer assistance. The procedures of ISM were developed through individual or group mental models that were used to calculate binary matrices, also called relation matrices. Besides, ISM was proposed by Warfield (1974) to present the relations of the criteria. ISM is a well-established method for recognising relationships among specific elements defining a problem. It originates as an interactive group learning process, yet individuals can also use it. In this process, a set of directly or indirectly linked elements are structured into a systematic model. Moreover, ISM is utilised to understand the relationships among the barriers and to develop insights into a collective understanding of these relationships (Huang et al., 2005).

This method identifies effects and explains the direction among the system's attributes. In addition, it establishes relationships among specific attributes to define a problem through their dependence and driving power (Mathiyazhagan et al., 2013). Due to its capability, ISM is a popular tool among academicians for analysing interrelationship attributes.

Table 2 The extracted factors

Factor (selected code)	*ID*	*Reference*
Governance	C1	Anthony et al. (2018), Maciel (2017), Gallotta et al. (2016), Nowak et al. (2011), Post et al. (2011)
Emerging trends	C2	Baiyere et al. (2020), Srivastava et al. (2020), Jnr et al. (2018), Kawdawatta and Marcelline (2018), Michanan et al. (2017), López-Pintado et al. (2017), Zhu et al. (2015), Djemame et al. (2014)
Infrastructure	C3	Mohammed et al. (2019), Anthony et al. (2018), Maciel (2017), Mohankumar and Kumar (2016), Gandomi and Amin (2014), Ghamkhari and Mohsenian-Rad (2013), Bianzino et al. (2010)
Strategy	C4	Rashid and Khan (2018), Anthony et al. (2018), Maciel (2017), Mohankumar and Kumar (2016), Ardito and Morisio (2014)
Green readiness	C5	Molla and Cooper (2014), Molla et al. (2011), Campbell et al. (2019), Nhamo (2013), Molla and Cooper (2010)
Tool	C6	Chowdhury and Hindle (2016), Noureddine and Rajan (2015), Li et al. (2015), Pathak et al. (2012)
Method	C8	Afum et al. (2021), Rashid and Khan (2018), Dick et al. (2013)
Stakeholders	C9	Saputri and Lee (2021), Roscoe et al. (2019), Anthony et al. (2018), Karita et al. (2019), Kern et al. (2019), Jnr et al. (2018), Gallotta et al. (2016), Manotas et al. (2016)
Policies and laws	C10	Shahzad et al. (2020), Ibrahim (2021), Anthony et al. (2018), Maciel (2017), Laskurain et al. (2017), Rashid and Khan (2018), Singh and Gond (2017), Ardito and Morisio (2014), Kern et al. (2019), Dick et al. (2013)
Assessment and monitoring	C11	Anthony et al. (2018), Maciel (2017), Gallotta et al. (2016), Mohankumar and Kumar (2016), Ardito and Morisio (2014), Kern et al. (2019), Betz and Caporale (2014), Nowak et al. (2011)

In reality, the ISM method consists of a series of repeated interrogations, usually using questionnaires, of a group of individuals whose opinions or judgements are of interest. After the initial interrogation of each individual, each subsequent interrogation will be accompanied by information regarding the preceding round of responses, usually presented anonymously. Thus, the individual is encouraged to reconsider and, if appropriate, to change their previous response in light of the responses of other group members (Seuring & Mueller, 2008).

The objective of this research was achieved by integrating MICMAC and ISM to produce validated frameworks for strengthening enablers and reducing barriers in different groups. The integrated methodological approach is used because MICMAC is flexible in evaluating the level of interactions between variables on a suitable scale. Based on the interrelationships, ISM, being an established technique in the literature, helps to build hierarchical structures (Dubey et al., 2016). Many researchers employed the ISM method in their studies; for instance, Valmohammadi and Dashti (2016) studied barriers to e-commerce implementation in the context of Iran. However, no study was found to have applied this approach to designing the green process management model in software development. Therefore, this approach was utilised to achieve the objective of the study.

5.2.2 Interpretive Structural Modelling Steps

Interpretive structural modelling was performed using six steps as follows:

1. Formation of the structural self-interaction matrix (SSIM): In this step, enablers were identified and added to the SSIM. Besides, the enablers' dimensions and their comparison results were included in this matrix. In addition, the types of the relations among the enablers in this matrix were determined by relations V, X, O, and A, as follows:
   - V: Barrier i helps to achieve barrier j, but j does not lead to i.
   - A: Barrier i does not help to achieve barrier j, but i leads to j.
   - X: Barriers i and j help to achieve each other.
   - O: Barriers i and j are unrelated.
2. Formation of the initial reachability matrix (RM): This matrix transformed SSIM matrix relation symbols to values 0 and 1 (Table 3 shows the rules).

Table 3 Converting conceptual relations to numbers (Thakkar et al., 2007)

Conceptual symbol	*i to j*	*j to i*
V	1	0
A	0	1
X	1	1
O	0	0

3. Formation of the final reachability matrix: This matrix was created by combining several relations among variables. Secondary relations among the indicators' aspects were managed in this matrix. For example, if dimension I led to dimension J, and dimension J led to dimension K, the dimension I would lead to dimension K, according to the secondary relation. In addition, if this were not the case in the reachability matrix, the matrix would be changed, with missing relations replaced. As a result, certain 0 elements would turn into 1, as represented by the symbol 1*. The final matrix was produced by recognising secondary relations and altering the received matrix.
4. Determining the level and priority of variables: The variables were levelled once reachability and antecedent sets for each element and the common set were determined. A reachability set was created for each element in which the final reachability matrix rows appeared as 1. Besides, a necessary set was one in which all columns numbered 1. In addition, a united collection was obtained by subscribing to these two collections. The first degree of precedence was given to elements whose common and reachability sets were the same. In addition, all elements' levels were determined by eliminating these elements and continuing the process with the remaining ones.
5. Producing an interpretive structural modelling (ISM) based on the final reachability matrix and levels specified.
6. Analysis of the driving power and dependence (the MICMAC method): The amount of driving was determined by the sum of the rows of values in the final reachability matrix for each element, while the amount of dependence was determined by the column sum (Ghasemi & Valmohammadi, 2021). Accordingly, four groups of elements were identified based on these two factors under the categories of autonomous, dependent, connected, and independent. The first group was characterised by low driving power, dependence, and separation from other factors. The second group had little driving power but much dependence. The third group had much power, yet it depended on other elements. Any actions applied to these factors would change other factors. The fourth group had strong driving power but low dependence. In fact, they were known as essential factors falling into one of the two groups of independent or related.

6 Results

The self-interaction matrix was established for the first time in this study. The symbols described in step one were employed in the self-interaction matrix. As Table 4 shows, expert judgement was used to produce this matrix. Next, the initial reachability matrix was created using values 0 and 1, followed by transverse relationships and the final reachability matrix (Table 5). In the original matrix, all entries in this table were 1* and 0. In the fourth step, the reachability set and the antecedent were extracted from the final reachability matrix, with the criteria graded. See Table 6.

6.1 ISM Formation

The ISM was drawn after each indicator's level was determined, with the final reachability matrix considered. Figure 1 depicts the final model. Accordingly, this model has four levels, the first and fourth being the most influential and the most permeable, respectively.

6.2 Driving and Dependence Power Analysis (MICMAC)

Using MICMAC analysis, the criteria were grouped based on the driving and dependent powers of each of the enablers (Table 3, Fig. 2). Accordingly, the criteria namely governance (C1), strategy (C4), stakeholders

Table 4 Structural self-interaction matrix (SSIM)

	C1	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*	*C8*	*C9*	*C10*	*C11*
C1		O	O	O	V	O	O	O	O	O	O
C2			O	O	A	O	V	O	O	O	O
C3				O	A	O	V	O	O	O	O
C4					V	O	O	O	O	O	O
C5						V	V	V	A	A	A
C6							V	O	O	O	O
C7								A	O	O	O
C8									O	O	O
C9										O	O
C10											O
C11											

Table 5 Final reachability matrix

	C1	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*	*C8*	*C9*	*C10*	*C11*	*Driving power*
C1	1	1*	1*	0	1	1*	1*	1*	0	0	0	7
C2	0	1	0	0	0	0	1	0	0	0	0	2
C3	0	0	1	0	0	0	1	0	0	0	0	2
C4	0	1*	1*	1	1	1*	1*	1*	0	0	0	7
C5	0	1	1	0	1	1	1	1	0	0	0	6
C6	0	0	0	0	0	1	1	0	0	0	0	2
C7	0	0	0	0	0	0	1	0	0	0	0	1
C8	0	0	0	0	0	0	1	0	0	0	0	2
C9	0	1*	1*	0	1	1*	1*	1*	1	0	0	7
C10	0	1*	1*	0	1	1*	1*	1*	0	1	0	7
C11	0	1*	1*	0	1	1*	1*	1*	0	0	1	7
Degree of dependence	1	7	7	1	6	7	11	7	1	1	1	

(C9), policy (C10), and assessment and monitoring (C11) were considered independent. These criteria had low dependence and high driving force, indicating that they had a high impact and low permeability. On the other hand, the dependent criteria were the emerging trends (C2), infrastructures (C3), tools (C6), green software development process (C7), and method (C8). Accordingly, these criteria had a high degree of dependence, low driving power, high permeability, and minimal impact on the system. The other criteria included interface factors with a high degree of dependence and driving power; in other words, they had a strong influence and high permeability, meaning that any minor changes in these criteria would cause significant changes.

7 Discussion

According to our findings, green governance is one of the important driving forces in managing green processes in software development. This finding is in line with the result of Hardin-Ramanan et al. (2018), who suggested a model for the governance of green information technology that reflects the responsiveness, mechanisms, and drivers of green IT in large Mauritian organisations. In the same vein, Jr et al. (2017) argue that the IT industry needs to adopt a green approach to its governance process; thus, a green IT governance framework can help reduce

Table 6 Determining levels of indicators

	Reachability set	*Antecedent set*	*Intersection set*	*Level*
C1	C1-C2-C3-C5-C6-C7-C8	C1	C1	4
C2	C2-C7	C1-C2-C4-C5-C9-C10-C11	C2	2
C3	C3-C7	C1-C3-C4-C5-C9-C10-C11	C3	2
C4	C2-C3-C4-C5-C6-C7-C8	C4	C4	4
C5	C2-C3-C5-C6-C7-C8	C1-C4-C5-C9-C10-C11	C5	3
C6	C6-C7	C1-C4-C5-C6-C9-C10-C11	C6	2
C7	C7	C1-C2-C3-C4-C5-C6-C7-C8-C9-C10-C11	C7	1
C8	C7-C8	C1-C4-C5-C8-C9-C10-C11	C8	2
C9	C2-C3-C5-C6-C7-C8-C9	C9	C9	4
C10	C2-C3-C5-C6-C7-C8-C10	C10	C10	4
C11	C2-C3-C5-C6-C7-C8-C11	C11	C11	4

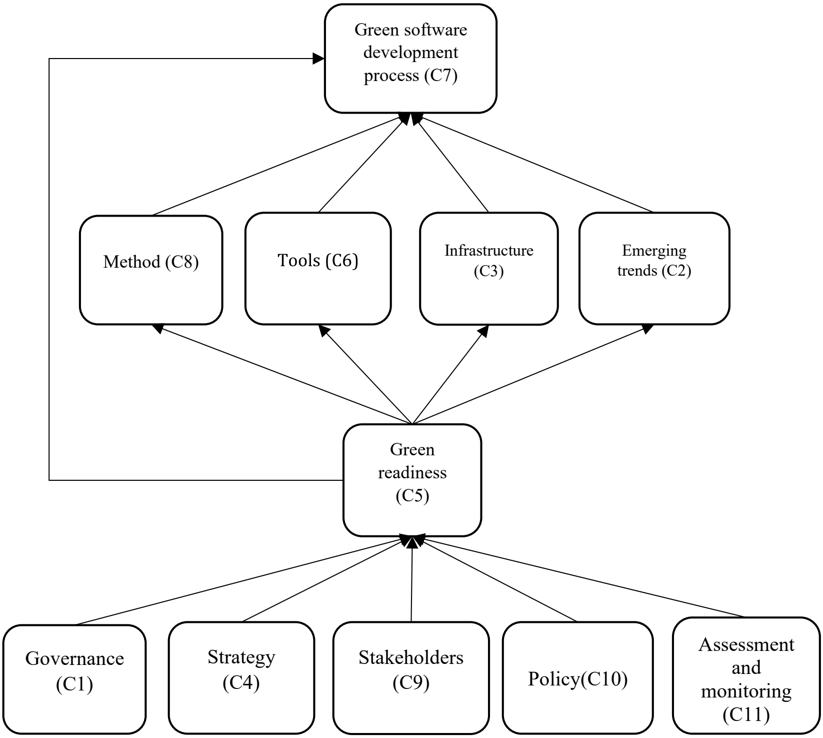

Fig. 1 The levels of structural model

environmental risks in an environmentally friendly manner. Therefore, they proposed a green information technology governance framework to reduce environmental risks.

Organisations need a green strategy fitting green governance, which must be aligned with the strategies of the business and the organisation. Accordingly, adopting a clear green strategy is vital to obtaining long-term and short-term goals. Organisations must develop specific strategies for carbon emissions and energy and waste management. Carbon emission management focuses on managing and reducing carbon emissions by organisations. This will include the use of information technology systems specifically designed to reduce carbon emissions and programmes for measuring software systems' carbon emissions and reducing them

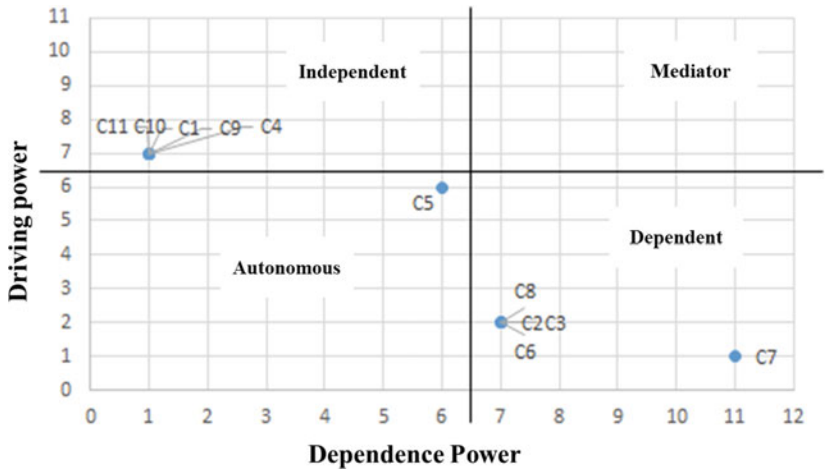

Fig. 2 Driving power and dependence

in the software development process. Energy and waste management is another essential issue considered in organisational strategies. Organisations need comprehensive, coordinated, and well-managed policies and rules. The framework of the green IT policy should be fit to ensure that green IT becomes a business work plan which ultimately leads to the least risk-faced projects. Strategies must be developed to be consistently implemented in various fields, including risk management, energy costs, resource consumption, and environmental effects of carbon emissions. In Dezdar's study (2017), the green policy factor was considered the acceptance factor for green information technology.

Assessment and monitoring of the use of resources and the environmental effects of carbon emissions on a continuous basis and at specified intervals is another important factor which needs specific attention by the software industry. For instance, at the beginning of a software development project, measuring environmental risks is one of the important factors which must be considered. Another factor that organisations should pay attention to is measuring energy costs in such a way that energy costs are considered and monitored separately. In this regard, the research of Noureddine et al. (2012) can be used to implement a time-energy consumption monitoring framework to help developers identify points with high energy consumption. In another study, Duarte et al. (2019) introduced a framework for evaluating software energy costs based on model analysis. Software is modelled as label transfer systems in this model, and the energy costs are interpreted using existing tools. Graph-based algorithms are then applied to scroll the models to obtain information about the software consumption associated with software behaviour, such as the least/most expensive software implementation, specific implementation cost, and average software implementation cost. In their research, they concluded that developers could create more energy-efficient software and consider potential exchanges for time, space, and energy costs when producing new versions of their systems.

Stakeholders are among the most important factors influencing the process of green software development. In software development, stakeholders include senior executives, project managers, engineers, software developers, and software users (Hirasawa, 2020). Having a green attitude towards creating a desire for change and a commitment to change is one of the prerequisites for the process of green software development. The green attitude indicates an organisation and employees' attitude towards environmental sustainability, which is the primary factor for implementing long-term green business process management. Promoting environmental awareness is an easy way to become an environmental observer, which results in improving environmental behaviours. Many resources are available for increasing awareness of environmental aspects. In addition, administrators, developers, and users of the training software can be instructed through group learning, informative and inspiring seminars, environmental books, brochures, and social networks as Karita et al. (2019) surveyed on the software industry's awareness of green and sustainable software engineering. New research results confirm the original survey's evidence, showing that software engineering sustainability is a novel issue for software specialists. However, experts show interest in this issue, and there is a general understanding that sustainability should be considered a qualitative feature. Among the observations made, they developed an initial theory indicating that software specialists are aware of this issue, even subconsciously in the "green in software" state. This study provided evidence about how the industry understands sustainability practices in software development. Although the community of software engineers has increased its interest in green and sustainable software engineering, the software industry in this area has not adequately considered it. Accordingly, green and sustainable practices have not been fully understood. Green readiness is on the second level of the model that affects the tools, methods, and infrastructures used and emerging trends. These four criteria are at the third level. It is worth mentioning that software and hardware make sense together. The use of correct infrastructures is effective in the green software development process. Cloud computing, virtualisation, and green data centre could be used to achieve this goal. In computer science, virtualisation technology has led to great development. The virtual implementation of hardware devices with the same functionality has brought many benefits. Software is virtual, and it is not physical. Therefore, virtualisation is often in software, which runs on a specific piece of hardware.

Various approaches have been introduced to software development. Software development using agile and Lean methods is one of the effective methods in green software development. Various studies examined the agile approach to green software development. In this regard, Rashid and Khan (2018), in their research, addressed agile methods employed by global software development manufacturers to develop green and sustainable software.

Various tools can be used to facilitate the process of green software development. For example, one can refer to automation tools in various processes of the software life cycle (testing, developing, and monitoring [energy measurement]). Fatima et al. (2020) systematically studied private tool support for developing the green android operating system. Macro et al. (2019) introduced a new multi-objective optimisation tool for green infrastructure planning. So, it is inferred that to enhance the practices of circularity, which can lead to the fortification of sustainability practices of companies engaged in software development, paying necessary attention and following the criteria and the interaction identified in this study is vital. As discussed above, all of the identified criteria play an important role in managing the economical consumption of tangible and intangible resources, such as natural resources, energy, the total time spent on software development projects, etc.

8 Conclusion

Green development is one of the most important innovations of the new millennium that will bring about great changes in human society. Sustainability and sustainable development are important issues in all industries, including IT and software systems development. Since the development of integrated software systems has made daily human life highly dependent on technology. Sustainability has been considered for all software systems because each new system becomes dependent as it becomes part of the technical infrastructure. Its continued use may place new burdens on ecological systems. Processes can be redesigned to reduce resource and energy use, and sustainability as an important dimension by considering carbon emissions, renewable energy consumption, waste generation, and other environmental performance criteria can help facilitate a circular economy and establish a sustainable organisation, particularly in the realm of the software industry.

In this model, we presented a set of factors using qualitative content analysis, extracted from texts and specialised interviews and classified into four levels based on the ISM method. By drawing the network of interactions, the factors, as well as their roles and functions, were determined. The final model consists of 4 levels. The first and fourth levels were determined as the most influential and permeable, respectively. Accordingly, the criteria, namely, governance, strategy, stakeholders, policy, and assessment and monitoring, were determined independently. These criteria have low dependence and high driving force, indicating that they have a high impact and low permeability. On the other hand, the emerging trends, infrastructures, tools, green software development process, and method were determined as dependent criteria. These criteria have a high degree of dependence, low driving power, a high level of permeability, and a minimal impact on the system. The other criteria included interface factors with a high degree of dependence and driving power; in other words, they had a strong influence and high permeability, meaning that any minor changes in these criteria would cause significant system changes. In summary, the obtained results might help software industry managers and experts apply the necessary measures to implement a sustainable software development project successfully.

References

Afum, E., Gao, Y., Agyabeng-Mensah, Y., & Sun, Z. (2021). Nexus between lean operations, eco-product innovativeness, social, green and business performances: Empirical evidence from Ghanaian manufacturing SMEs. *Journal of Manufacturing Technology Management, 32*(8), 1557–1577.

Alam, G. M., Erdiaw-Kwasie, M. O., Nagy, G. J., & Leal Filho, W. (Eds.). (2021). *Climate vulnerability and resilience in the global south: Human adaptations for sustainable futures*. Springer Nature.

Alam, K., Ali, M. A., Erdiaw-Kwasie, M., Shahiduzzaman, M., Velayutham, E., Murray, P. A., & Wiesner, R. (2022). Impact of ICTs on innovation and performance of firms: Do start-ups, regional proximity and skills matter? *Sustainability, 14*(10), 5801.

Anderl, R. (2015). Industrie 4.0—Technological approaches, use cases, and implementation. *at-Automatisierungstechnik, 63*(10), 753–765.

Anthony, B. J., Majid, M. A., & Romli, A. (2018). An empirical study on predictors of green sustainable software practices in Malaysian electronic industries. *Journal of Information and Communication Technology, 17*(2), 347–391.

Ardito, L., & Morisio, M. (2014). Green IT—Available data and guidelines for reducing energy consumption in IT systems. *Sustainable Computing: Informatics and Systems, 4*, 24–32.

Baiyere, A., Salmela, H., & Tapanainen, T. (2020). Digital transformation and the new logics of business process management. *European Journal of Information Systems, 29*(3), 238–259.

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

Bianzino, A. P., Chaudet, C., Rossi, D., & Rougier, J. L. (2010). A survey of green networking research. *IEEE Communications Surveys & Tutorials, 14*(1), 3–20.

Birrell, N. D. (1985). *A practical handbook for software development*. Cambridge University Press.

Calero, C., & Piattini, M. (2017). Puzzling out software sustainability. *Sustainable Computing: Informatics and Systems, 16*, 117–124.

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

Calero, C., Moraga, M. Á., & García, F. (2022). Software, sustainability, and UN sustainable development goals. *IT Professional, 24*(1), 41–48.

Campbell, L. K., Svendsen, E., Sonti, N. F., Hines, S. J., & Maddox, D. (2019). Green readiness, response, and recovery: A collaborative synthesis. General Technical Reports NRS-P-185. US Department of Agriculture, Forest Service. 358 p., 185, 1–385.

Chowdhury, S. A., & Hindle, A. (2016, May). Characterising energy-aware software projects: Are they different? In *Proceedings of the 13th International Conference on Mining Software Repositories* (pp. 508–511).

Demestichas, K., & Daskalakis, E. (2020). Information and communication technology solutions for the circular economy. *Sustainability, 12*(18), 7272.

Dezdar, S. (2017). Green information technology adoption: Influencing factors and extension of theory of planned behavior. *Social Responsibility Journal, 13*(2), 292–306.

Dick, M., Drangmeister, J., Kern, E., & Naumann, S. (2013). Green software engineering with agile methods. In *2nd International Workshop on Green and Sustainable Software (GREENS)* (pp. 78–85).

Djemame, K., Armstrong, D., Kavanagh, R., Ferrer, A. J., Perez, D. G., Antona, D., & Blasi, L. (2014). Energy efficiency embedded service lifecycle: Towards an energy efficient cloud computing architecture. In *CEUR Workshop Proceedings* (Vol. 1203, pp. 1–6). CEUR Workshop Proceedings.

Duarte, L. M., da Silva Alves, D., Toresan, B. R., Maia, P. H., & Silva, D. (2019, September). A model-based framework for the analysis of software energy consumption. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (pp. 67–72).

Dubey, R., Gunasekaran, A., Papadopoulos, T., Childe, S. J., Shibin, K. T., & Fosso Wamba, S. (2016). Sustainable supply chain management: Framework and further research directions. *Journal of Cleaner Production, 142*(2), 1119–1130.

Duboc, L., Betz, S., Penzenstadler, B., Kocak, S. A., Chitchyan, R., Leifler, O., Porras, J., Seyff, N., & Venters, C. (2019). Do we really know what we are building? Raising awareness of potential sustainability effects of software systems in requirements engineering. In *IEEE International Requirements Engineering Conference*. IEEEComputer Society.

Fatima, I., Anwar, H., Pfahl, D., & Qamar, U. (2020). Tool support for green Android development: A systematic mapping study. In *5th International Conference on Software Technologies—ICSOFT* (pp. 409–417).

Gallotta, B., Garza-Reyes, J. A., Anosike, A., Lim, M. K., & Roberts, I. (2016). A conceptual framework for the implementation of sustainability business processes. In *Proceedings of the 27th Production and Operations Management Society (POMS)*.

Gandomi, Z., & Amin, S. (2014). Towards green computing application for measuring the sustainability of data centers: An analytical survey. In *The International Conference on Electrical, Electronics, Computer Engineering and their Applications, EECEA2014* (pp. 86–96).

Garcia, S., Cintra, Y., Rita de Cássia, S. R., & Lima, F. G. (2016). Corporate sustainability management: A proposed multi-criteria model to support balanced decision-making. *Journal of Cleaner Production, 136*, 181–196.

Ghamkhari, M., & Mohsenian-Rad, H. (2013). Energy and performance management of green data centers: A profit maximisation approach. *IEEE Transactions on Smart Grid, 4*(2), 1017–1025.

Ghasemi, B., & Valmohammadi, C. (2021). Identification and prioritisation of critical success factors of knowledge management implementation using a novel hybrid group decision-making model. *VINE Journal of Information and Knowledge Management Systems*.

Grottke, M., Matias, R., & Trivedi, K. S. (2008, November). The fundamentals of software aging. In *2008 IEEE International Conference on Software Reliability Engineering Workshops (ISSRE Workshop)* (pp. 1–6). IEEE.
Hardin-Ramanan, S., Chang, V., & Issa, T. (2018). A Green information technology governance model for large Mauritian companies. *Journal of Cleaner Production, 198*, 488–497.
Hilty, L. M., & Aebischer, B. (2015). ICT for sustainability: An emerging research field. In *ICT innovations for sustainability* (pp. 3–36). Springer.
Hirasawa, N. (2020). Usability in software development process—Proposal of society/stakeholder centered design (SCD).
Huang, J.-J., Tzeng, G.-H., & Ong, C.-S. (2005). Multidimensional data in multidimensional scaling using the analytic network process. *Pattern Recognition Letters, 26*(6), 755–767.
Ibrahim, S. A. (2021). Green software process assessment: The theoretical framework. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*(3), 2011–2016.
Jannat, U. K. (2016). Green software engineering adaption in requirement elicitation process. *5*, 94–98.
Jetley, R., Nair A., Chandrasekaran, P., & Dubey A. Applying software engineering practices for development of industrial automation applications. In *2013 11th IEEE International Conference on Industrial Informatics (INDIN)* (pp. 558–563).
Jnr, B. A., Majid, M. A., & Romli, A. (2018). An empirical study on predictors of green sustainable software practices in Malaysian electronic industries. *Journal of Information and Communication Technology, 17*(2), 347–391.
Jr, B. A., Majid, M. A., & Romli, A. (2017). A green information technology governance framework for eco-environmental risk mitigation. *Progress in Industrial Ecology, an International Journal, 11*(1), 30–48.
Kallio, H., Pietilä, A. M., Johnson, M., & Kangasniemi, M. (2016). Systematic methodological review: Developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing, 72*(12), 2954–2965.
Karita, L., Mourão, B. C., & Machado, I. (2019, September). Software industry awareness on green and sustainable software engineering: A state-of-the-practice survey. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (pp. 501–510).
Kawdawatta, T., & Marcelline, Z. (2018, October). Reducing emissions through digital transformation. In *2018 National Information Technology Conference (NITC)* (pp. 1–3). IEEE.
Kern, E., Guldner, A., & Naumann, S. (2019). Including software aspects in green IT: How to create awareness for green software issues. In *Green IT engineering: Social, business and industrial applications* (pp. 3–20). Springer.

Khadivar, A., & Abdian, S. (2018). Modeling the maturity of green information technology with the approach of systems dynamics. *Organizational Resource Management Research, 8*(2), 73–91.

Kutscher, V., Ruland, S., Müller, P., Wasser, N., Lochau, M., Anderl, R., & Hähnle, R. (2020). Towards a circular economy of industrial software. *Procedia CIRP, 90*, 37–42.

Lago, P., Koçak, S. A., Crnkovic, I., & Penzenstadler, B. (2015). Framing sustainability as a property of software quality. *Communications of the ACM, 58*(10), 70–78.

Laskurain, I., Ibarloza, A., Larrea, A., & Allur, E. (2017). Contribution to energy management of the main standards for environmental management systems: The case of ISO 14001 and EMAS. *Energies, 10*(11), 1758.

Li, F., Qanbari, S., Vögler, M., & Dustdar, S. (2015). Constructing green software services: From service models to cloud-based architecture. In *Green in software engineering* (pp. 83–104). Springer.

López-Pintado, O., García-Bañuelos, L., Dumas, M., & Weber, I. (2017, September). Caterpillar: A blockchain-based business process management system. In *BPM* (Demos).

Maciel, J. C. (2017). The core capabilities of green business process management—A literature review. In *Proceedings of the Wirtschaftsinformatik (WI)* (pp. 1526–1537).

Macro, K., Matott, L. S., Rabideau, A., Ghodsi, S. H., & Zhu, Z. (2019). OSTRICH-SWMM: A new multi-objective optimisation tool for green infrastructure planning with SWMM. *Environmental Modelling & Software, 113*, 42–47.

Mahmoud, S. S., & Ahmad, I. (2013). A green model for sustainable software engineering. *International Journal of Software Engineering and Its Applications, 7*(4), 55–74.

Manotas, I., Bird, C., Zhang, R., Shepherd, D., Jaspan, C., Sadowski, C., & Clause, J. (2016, May). An empirical study of practitioners' perspectives on green software engineering. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)* (pp. 237–248). IEEE.

Marquis, C., Zhang, J., & Zhou, Y. (2011). Regulatory uncertainty and corporate responses to environmental protection in China. *California Management Review, 54*(1), 39–63.

Martin, S. L., Javalgi, R. R. G., & Ciravegna, L. (2018). Service advantage built on service capabilities: An empirical inquiry of international new ventures. *Journal of Business Research, 88*, 371–381.

Mathiyazhagan, K., Govindan, K., NoorulHaq, A., & Geng, Y. (2013). An ISM approach for the barrier analysis in implementing green supply chain management. *Journal of Cleaner Production, 47*, 283–297.

Michanan, J., Dewri, R., & Rutherford, M. J. (2017). GreenC5: An adaptive, energy-aware collection for green software development. *Sustainable Computing: Informatics and Systems, 13*, 42–60.

Mishra, A., & Mishra, D. (2021). Sustainable software engineering: Curriculum development. In C. Calero, M. A. Moraga, & M. Piattini (Eds.), *Software sustainability. Springer*. ISBN978-3-030-69970-3.

Mohammed, M. A., Mohammed, I. A., Hasan, R. A., Ţăpuş, N., Ali, A. H., & Hammood, O. A. (2019, October). Green energy sources: Issues and challenges. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)* (pp. 1–8). IEEE.

Mohankumar, M., & Kumar, M. A. (2016). Green database design model in software development life cycle phase. *Indian Journal of Science and Technology, 9*(30).

Molla, A., & Cooper, V. (2010). Green IT readiness: A framework and preliminary proof of concept. *Australasian Journal of Information Systems, 16*(2).

Molla, A., & Cooper, V. (2014). Greening data centers: The motivation, expectancy and ability drivers.

Molla, A., Cooper, V., & Pittayachawan, S. (2011). The green IT readiness (G-readiness) of organisations: An exploratory analysis of a construct and instrument. *Communications of the Association for Information Systems, 29*(1), 4.

Naumann, S., Dick, M., Kern, E., & Johann, T. (2011). The GREENSOFT Model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems, 1*(4), 294–304.

Nhamo, G. (2013). Green economy readiness in South Africa: A focus on the national sphere of government. *International Journal of African Renaissance Studies-Multi-, Inter- and Transdisciplinarity, 8*(1), 115–142.

Noureddine, A., & Rajan, A. (2015, May). Optimising energy consumption of design patterns. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering* (Vol. 2, pp. 623–626). IEEE.

Noureddine, A., Bourdon, A., Rouvoy, R., & Seinturier, L. (2012, September). Runtime monitoring of software energy hotspots. In *2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering* (pp. 160–169). IEEE.

Nowak, A., Leymann, F., Schumm, D., & Wetzstein, B. (2011, August). An architecture and methodology for a four-phased approach to green business process reengineering. In *International Conference on Information and Communication on Technology* (pp. 150–164). Springer.

Pathak, A., Hu, Y. C., & Zhang, M. (2012, April). Where is the energy spent inside my app? Fine grained energy accounting on smartphones with Eprof.

In *Proceedings of the 7th ACM European Conference on Computer Systems* (pp. 29–42).

Pinto, G., & Castor, F. (2017). Energy efficiency: A new concern for application software developers. *Communications of the ACM, 60*(12), 68–75.

Post, C., Rahman, N., & Rubow, E. (2011). Green governance: Boards of directors' composition and environmental corporate social responsibility. *Business & Society, 50*(1), 189–223.

Pressman, R. S., & Maxim, B. R. (2016). Software engineering: A professional approach.

Raisian, K., Yahaya, J., & Deraman, A. (2016). Current challenges and conceptual model of green and sustainable software engineering. *Journal of Theoretical & Applied Information Technology, 94*(2).

Rashid, N., & Khan, S. U. (2018). Agile practices for global software development vendors in the development of green and sustainable software. *Journal of Software: Evolution and Process, 30*(10), e1964.

Roscoe, S., Subramanian, N., Jabbour, C. J., & Chong, T. (2019). Green human resource management and the enablers of green organisational culture: Enhancing a firm's environmental performance for sustainable development. *Business Strategy and the Environment, 28*(5), 737–749.

Saputri, T. R. D., & Lee, S. W. (2021). Integrated framework for incorporating sustainability design in software engineering life-cycle: An empirical study. *Information and Software Technology, 129*, 106407.

Seidel, S., Recker, J., & vom Brocke, J. (2012). Green business process management. In *Green business process management* (pp. 3–13). Springer.

Seuring, S., & Mueller, M. (2008). From a literature review to a conceptual framework for sustainable supply chain management. *Journal of Cleaner Production, 16*(15), 1699–1710.

Shahbandarzadeh, H., & Kabgani M. H. (2016). Quantitative analysis of the opportunities and challenges of utilising green information technology in the organisation. *Intelligent Business Management Studies, 4*(14), 97–116.

Shahzad, M., Qu, Y., Zafar, A. U., Rehman, S. U., & Islam, T. (2020). Exploring the influence of knowledge management process on corporate sustainable performance through green innovation. *Journal of Knowledge Management*.

Shenoy, S., & Eeratta, R. (2011, December). Green software development model: An approach towards sustainable software development. In *India Conference (INDICON), 2011 Annual IEEE*.

Singh, S., & Gond, S. (2017). Green computing and its impact. In *Nature-inspired computing: Concepts, methodologies, tools, and applications* (pp. 1628–1642). IGI Global.

Spenser, M. (2010). Understanding the software development process. https://www.projectsmart.co.uk.

Srivastava, A., Gupta, M. S., & Kaur, G. (2020). Energy efficient transmission trends towards future green cognitive radio networks (5G): Progress, taxonomy and open challenges. *Journal of Network and Computer Applications*, 102760.

Taghva, M. R., Zohrabi, M., & Dehdashti Shahrokh, Z. (2019). A Structural model for green information technology and organizational sustainability. *BI Management Studies, 7*(27), 5–28.

Thakkar, J., Deshmukh, S., Gupta, A., & Shankar, R. (2007). Development of a balanced scorecard: An integrated approach of Interpretive Structural Modeling (ISM) and Analytic Network Process (ANP). *International Journal of Productivity and Performance Management, 56*(1), 25–59.

Tukker, A. (2015). Product services for a resource-efficient and circular economy—A review. *Journal of Cleaner Production, 97*, 76–91.

Tukker, A., & Tischner, U. (2006). Product-services as a research field: Past, present and future. Reflections from a decade of research. *Journal of cleaner production, 14*(17), 1552–1556.

Valmohammadi, C., & Dashti, S. (2016). Using interpretive structural modeling and fuzzy analytical process to identify and prioritise the interactive barriers of e-commerce implementation. *Information & Management, 53*(2), 157–168.

Vom Brocke, J., Seidel, S., & Recker, J. (Eds.). (2012). *Green business process management: Towards the sustainable enterprise*. Springer Science & Business Media.

Warfield, J. N. (1974). Toward interpretation of complex structural models. *IEEE Transactions on Systems, Man, and Cybernetics, 5*, 405–417.

Yahaya, J., Raisian, K., Ibrahim, S. R. A., & Deraman, A. (2019). *Green software process based on sustainability dimensions: The empirical investigation.*

Zhu, C., Leung, V. C., Shu, L., & Ngai, E. C. H. (2015). Green internet of things for smart world. *IEEE Access, 3*, 2151–2162.