

The Development of Green Software Process Model

A Qualitative Design and Pilot Study

Siti Rohana Ahmad Ibrahim¹

Jamaiah Yahaya², Hasimi Salehudin³

Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

Aziz Deraman⁴

Faculty of Ocean Engineering Technology and Informatics
Universiti Malaysia Terengganu
Kuala Terengganu, Terengganu, Malaysia

Abstract—Software process and development are the fundamental activities in software engineering. Increasing software usage either develops in-house or outsourcing requires improving the software process accordingly to minimise the adverse effects on the environment. The resources and power consumption controlled by hardware affected the software to move the process that causes high emission of power and energy. Thus, most of the existing work on software development is aimed at the efficiency of hardware operation through CPU, memory, and processor. Although sustainability is still initial in software engineering, the green software process can be achieved through sustainable development that concerns the preservation of the environment. However, there is still a lack of study and effort in the software process that emphasises sustainability perspectives and software waste elimination. Therefore, this study proposes the green factors for the software process that consider the sustainability elements and waste reduction during development. The green factors are the benchmark to measure a sustainable and green software process. Besides, this paper also presents the qualitative interview design and pilot study. The pilot study analysis has demonstrated the reliability of the interview protocol. Therefore, the actual interview and data analysis are currently in progress.

Keywords—Software process; green factor; waste reduction; qualitative instrument; green software process model

I. INTRODUCTION

Software was becoming crucial and became part of our life today. The development of software systems needs to reflect on sustainability and environmental impact [1]. Software is vital to do work and tasks more efficiently. Software causes hardware activity, and it is responsible for the increment of energy consumption in that way. It has become the central aspect of daily life, and most people cannot imagine that future development without software.

Electricity consumption in the ICT sector is expected to increase by 60% from 2007 to 2020. Since ICT contributes to 2% of carbon dioxide emissions from the electronics, household, ICT services, and equipment sector, an estimated 8% is from electricity consumption [1]. Systematic methods for improving the organisation's environmental sustainability provided several guidelines such as the Eco-Management and Audit (EMAS) Scheme, the ISO 14001 standard, and the Environmental Management System (EMS). Meanwhile, large organisations need to publish their organisation's corporate social reports for employee compliance [2].

Generally, sustainability means the beginning to nurture the future generation aligned with the present and compliance with the own needs [3]. Sustainability in the software system is a concept of recycling to protect the environment as excessive waste production can harm the environment's well-being [4]. In addition, sustainability is a resource to support life and the environment that requires the courage of individuals or communities to be maintained for a long time [5]. In engineering, it needs to demonstrate efforts to improve the environment, social and economic. Therefore, sustainable software development is for the system requirements required [6] to ensure the sustainability of nature.

Today, "going green" is an essential part of business practices worldwide related to the environment [7]. Several factors that determine the greenness of the software process have been considered that take into account the sustainability aspect such as resources and products [8][9], people [7][10], organisation [11][12], technical [13][14], and environmental [15][16]. To create every phase of the software process sustainable, it must go through the green strategy, guideline, and green metric that ultimately reflect the greenness of the entire stage [4].

Over the past three years, green software engineering has grown and evolved due to concerns over the software, industry, and consumers' development. However, it focuses on the fundamental aspects of energy efficiency and power consumption, affecting the environment [17]. Many studies discuss green software development that focuses on hardware perspective on energy efficiency and power consumption [18]. Green software development as a whole, the software process methodology should be considering investigating their requirement to create sustainable and green manner.

Green software engineering goals are to minimise the negative effect on the environment [1][19] and produce sustainable software products. Previous studies [20-23] emphasise green hardware in energy efficiency, power consumption, waste reduction, and disposal. However, it seems to lack studies on green software even though the impacts are direct to hardware [17], but the software also indirectly impacts. The initiatives of green software are to save resources such as energy, power, and natural resources to preserve the stability of the ecosystem [24]. Nowadays, the software is no longer viewed in terms of productivity but needs to focus on scalability, usability, and quality [25] with recent advances in-demand new technology.

This research is funded by the Malaysia Ministry of Higher Education, Fundamental Research Grant Scheme (FRGS/1/2019/ICT01/UKM/02/1).

Therefore, this research is essential to ensure the software development process complies with the green requirement as needed by the community and industry in general. Currently, the green development issues are not being focused on comprehensively, especially from a sustainability perspective.

The article organisation is as follows. In Section 2, we discuss the background study and related works of the green software process. While in Section 3, the research method and activities are presented. It discusses the approach of carrying out this research. In Section 4, we will discuss the qualitative design and pilot analysis, and lastly, we will conclude this paper with a conclusion and future work.

II. RELATED WORK

The software process is a work in progress requiring changes to the user's needs to use the software effectively. In addition, the target of the software process is to improve the job done and used by various companies facing the inevitable difficulties and uncertainties of the process software. Some aspects of software development require the user's attention; the product developed should reach the customer well based on the specified cost and time [26].

Applying green practices as part of the software development process attempts to achieve a green software process. It means to practice for the software is considering the environmental aspects. In addition, the software process involves the development, operation, and maintenance that is carried out in a green approach and producing green software processes. Besides, the green software process can create less waste throughout the development [17, 22].

The transition to the green software process requires careful study as it involves several critical activities in the software development process. The implementation of green initiatives can be done in phases, observing how the software works without cost during the software development process. Research on the transition and development of green software processes is to improve the practices of the process. There are ways to develop green software processes that are to implement green software process practices. Implementing this method supports reducing waste during software development, such as unproductive workers, power savings, and resources.

A. Issues and Challenges in Green Software Process

Information and Communication Technology (ICT), particularly in software development, remains significant and increasingly important even with the emergent of new areas such as big data, cloud computing, Cyber-Physical Systems (CPS), and IoT [25]. Nevertheless, software development still has issues and challenges maintaining relevance and sustainability of the process to ensure the quality of the developed product.

1) *Software development process*: As described by [27], software development involves a wide range of skills and disciplines, including identifying user needs, values, and features that support the final product. Software development projects have many obstacles and barriers to complete and deliver the project. Several factors come from the organisation

environment, team, and user perspective that are not well-prepared according to requirement, budget, and time [28-29]. Most software development projects failed due to higher budgets, overtime, and low customer satisfaction. According to [25], the causes of failures are time, finances, staffing, and management.

Various software projects are often ambiguous, where the goals and objectives are unclear due to the lack of experience from the users. Software project development requires a lot of communication and the time taken to complete it successfully. As developers, they are intelligent users who do not use specialists to translate their needs into software engineering. In addition, developers should engage users in software development to help them understand the needs and delivery of project functions well. Users involvement [26], from the beginning of software development to the end, is vital as this is one of the criteria for the success of a project.

Generally, the software process has five main phases: requirement specification, design, implementation, testing, and maintenance. Some activities in this phase may cause problems and challenges, especially in fulfilling current demands such as reducing paper use, e-waste generation, and controlling carbon footprint and energy efficiency.

Shenoy and Eeratta [22] suggested creating sustainable and green software processes without adversely affecting global economic, environmental, social, and individual well-being. At the same time, there are five primary and fundamental activities associated with the development process and will be discussed in the following subsections:

a) *Requirement Specification Phase*: The requirement specification is an important activity in software development. It is conducted with the specified team by getting information from the users. The software requirements specification (SRS) is delivered as the outcome of requirement gathering and analysis activity [30]. A good software requirement specification aims to satisfy the needs of its customers. Based on [31], some possible green analysis criteria include viability, requirements, and tests.

Many problems exist in requirement analysis, such as lack of user input, incomplete and frequently changing requirements, and specifications. The literature suggested coming across the software regarding its lifespan and durability [22]. In addition, the collection of requirement specifications is required through electronic methods to save resources such as paper and preserve the environment. Furthermore, as [30, 32] suggested, software should adopt new hardware for energy efficiency and coping with power down during its operation.

b) *Design Phase*: The functional and technical requirements gathered during the requirement stage create the design document and artefact. During the design phase, the SRS is referred to as the primary reference for product architecture.

This documentation is supportive of the software development process. Therefore, the design team needs to cooperate with the software developer on this document

before proceeding to the next phase. All stakeholders must review the Document Design Specification (DDS), and the best approach is selected based on budget, time constraints, risk, robustness, and design method [30]. Reference [31] suggested that performance, reusability, and documentation are among the green analysis criteria in the green process. Design reuse and virtual design are also other factors defined in the green process by [33]. Furthermore, previous researchers [22, 33] argue that the initial design should be moderate to avoid frequent design changes. The design should not be too large and lead towards practices that protect the resources.

c) Implementation Phase: Implementation and coding are the subsequent activities in the software development process. The source code is generated based on a detailed and well-organised design to avoid problems during the development. This phase involves source code written in specific programming languages based on the developer's preference and the project approach.

The implementation phase is an emphasis on programming. There are several things to avoid, which are the use of duplicate code, custom hardware APIs, and resource-intensive APIs. It encouraged the user to utilise the high resources on tap systems that need to process programming complexity. In addition, paired programming, code reuse, and automated code generation support minimising and saving energy consumption [22].

The guidelines in the design and implementation phase can support the sustainability of the environment if the programmer needs to be proficient in writing algorithms by summarising design code and data structure according to the hardware architecture, programming languages, and applications. Programmers must produce practical algorithms by minimising the number of program lines to reduce their execution later. As mentioned by [33-34], extending the program code requires energy efficiency and additional work to execute the code.

d) Testing Phase: The testing phase relates to the product errors detected and re-analysed until they meet the quality requirements of SRS. There are different types of tests performed by integration system testing. Reference [30] stated that testing aims to ensure that the developed software is free of bugs, faults, or defects and meets all requirements specified during the requirement stage. The specifications are needed to be understood by conducted initial testing at the final stage of requirements. Then, different testers [35] can avoid bias and validate requirements correctly and consistently. The user acceptance test is the last activity in the system testing. The end-users [31] usually are carried out this task to ensure the system meets their needs.

According to [32], the software development team faces time constraints to find errors through testing. Therefore, previous studies [22, 30] recommend using automated testing, reusing test cases to determine performance scalability and resource testing. In the testing phase, the program involved improvement of energy efficiency will use energy measuring instruments. The green analysis criteria are functionality and

measurement. The functionality refers to all tests explained in the requirements, while the measurement evaluates the suitability of product energy consumption [35] measured and inspected during testing.

e) Maintenance Phase: The maintenance phase is associated with software operation and maintenance, which implement the product in the real environment. Software maintenance takes place if issues need to be repaired or improved [36] [37] to ensure the sustainability and quality of the product are maintained and guaranteed.

Any errors and complaints should take through electronic documentation to save time and paper during the maintenance phase. Furthermore, software archaeology, reverse engineering, and software migration are not recommended because these will contribute to the high costs and effort in execution. It is encouraged to manage knowledge among the team member effectively [22] in the development and maintenance phase to ensure a clear understanding of the system. It may reduce cost and efforts in maintenance.

Maintenance costs are high due to understanding the system and the changes that need to be analysed. Some maintenance and development contracts are separate during system development because they are kept by the company and not by the system developer. It makes it difficult for maintenance tasks and increases energy consumption if there are any changes in the system as it needs time to understand the system [34]. It is suggested by [72] that the programmer write the code that is easy and more under regular programming because if any code change is required at any stage, the correction could be done quickly. The maintenance phase could be green by including the matrices like fault tolerance, failure management, and disposal stage.

Furthermore, to reduce costs and improve energy efficiency, managers must provide training or courses to the staff to understand the old and new programming languages to speed up the maintenance process. Environmental sustainability can be supported at the implementation stage, in which program development is written for good and clear understanding by programmers. If any changes are required, the program can be easily understood, and the internal maintenance work is completed quickly, improves energy efficiency [33,71], quality and long life of the product [38].

2) Software waste: According to [39], waste management is one of the critical aspects of preserving the environment. Waste is deemed by [40] as irrelevant and has no value to production. Solid waste produces waste in the manufacturing [40] and construction [41] industries. In general, software waste is used by resources without delivering any advantage. It may contain characteristics, objects, conditions, processes, and actions in project elements. In the software domain, wastes act as friction and occur throughout the development process [42] until they produce the end of the product. The wastes need to be reduced and thus will increase efficiency in development even though the awareness and identification of waste are critical during production.

Software waste often has a problem related [43] to scope, unclear requirements, specification and design, unnecessary features and technical, team conflict, and unorganised code programming during the development process. Many researchers have been investigated software waste during the development process [27, 44-46]. These wastes are identified, such as building the wrong feature or product, mismanaging the backlog, rework, unnecessarily complex solutions, extraneous cognitive load, psychological distress, waiting/multitasking, knowledge loss, and ineffective communication in software development.

Sedano et al. [47] suggested the way to improve software engineering productivity by proposing a theory of sustainable software development that incorporates sustainability toward the principles, policies and practices into the knowledge of extreme programming. These included encouraging knowledge sharing, developing a positive attitude toward the team disruption, and considering code quality. Besides that, the policies include team code ownership, a standard schedule, and avoiding technical debt. The developers can apply good practices like test-driven development, continuous refactoring, pair programming, overlapping pair rotation, and knowledge pollination [47].

3) *Green software process*: Previously, the general issues in the software development process are related to time constraints, high costs, bugs and errors. But nowadays, [48] discovers that more known problems are related to people, resource and development tasks coordination and tracking.

As mentioned earlier, the software development process in a suitable environment manner and practices will lead and enhance the efficiency of the development. Sound environmental practices attempt to improve the activities that need to be implemented through software process methodology. The method of green software refers to the use of resources to meet the needs of the software with regards to the economic, social, and environmental aspects and effects. Software sustainability is considered as the ability of the software to have a longer life. In addition, the production of little waste during software development and operation [49] is one of the meanings of green software.

Sustainability and green are closely related in software engineering to ensure the long-lasting of the software in their environment. The sustainable design of the software is essential to fulfilling the user's needs and current demands.

Sustainable software refines user needs in terms of economy, society, and environment [50] in the whole life cycle phase. It concerns ensuring the long-lasting software to design, develop and deploy. From a software perspective, green is aimed to reduce the carbon footprint and negative impacts of software based on economy, society, and environment [31] during the development process. The software engages with the life cycle in green software engineering and sustainable criteria. Furthermore, Agarwal, Nath, and Chowdhury [32] proposed integrating green software development life cycle and sustainability criteria.

Sustainability supports these three main dimensions, namely economic, social and environmental. There are two different views when it comes to software. The first is sustainable software that prioritises the principles, practices, and processes that contribute to the durability of technical sustainability software. Next, [51] argues that software systems need to support other dimensions of sustainability and solve problems external to software systems. The second is software engineering for sustainability (SE4S). Both of these views are considered important as the centre of software sustainability development [53], and they need to support the five primary dimensions of economic, social, environmental, individual, and technical [52].

Generally, the economic dimension refers to capital assets or values involving capital investment, income, wealth, and profit. In contrast, the social element encompasses society, which consists of organisations and groups of people engaged in democracy, justice, employment, and social equality. The environmental dimension relates to ecosystems, climate, pollution, waste, and natural resources that can long-term impact humans and biological systems. The individual dimensions include mental and physical well-being, self-esteem, and the freedom to view human well-being as an individual. The latter is the technical dimension which involves system maintenance, infrastructure, the evolution of changing environmental conditions, and the long-term concept of system information.

B. Green Factors

The green factor is the benchmark used to assess the greenness of the software development process based on a certain standard. It should be used to ensure greenness in the software process and be applicable for future generations.

A literature study has identified green resources, people, organisational, technical, and environmental factors. From the software perspective, resource means the natural resources that must be protected from human needs and wastes. For instance, cloud computing is an alternative way to reduce energy and space in physical conditions. Meanwhile, people refer to improving the quality of human life [54] that reflects emotional health, including behaviour, thought, and action.

Organisational refers to the current need of an organisation includes leadership, employee, and performance [55]. Besides, the technical factor [56] is to cope with changes, quickly adapt to future changes, and the longevity of software systems. The environmental aspect relates to avoiding harm to the environment [57], and thus software should have minimal effects on the background during the development and maintenance process. Table I indicates the green factors and elements in the software process based on previous studies. It shows that the most popular aspects are energy efficiency, resource optimisation, environmental effect, performance, organisational support, employee support, and capacity optimisation associated with the green software process (refer to Table I).

TABLE I. GREEN FACTORS AND ELEMENTS BASED ON SOFTWARE PROCESS BY PREVIOUS STUDIES

Green factors	Elements	A1 [21]	A2 [58]	A3 [11]	A4 [59]	A5 [60]	A6 [8]	A7 [56]	A8 [61]	A9 [62]	A10 [63]	A11 [64]	A12 [65]	A13 [66]	A14 [67]	A15 [68]	Total
Resource	Cost						X			X							2
	Energy efficiency	X	X									X	X	X	X		6
	Performance												X	X		X	3
	Resource usage	X														X	2
People	Personalization						X										1
	Leadership development						X										1
	Human resource			X							X						2
	Human competences										X						1
	User's perception of environmental					X											1
Organisation	Organisational Support				X				X	X							3
	Quality of Human Resource								X								1
	Employee Support			X	X					X							3
	Tool Support			X	X												2
Technical	Perdurability					X											1
	Support of shared services						X										1
	Software quality requirement						X										1
	Portability						X										1
	Modularity						X										1
	Maintainability							X									1
	Evolution							X									1
Environmental	Waste reduction					X											1
	Environmental effect		X			X						X		X			4
	Capacity optimisation					X							X			X	3
	Emission			X										X			2
	Resource optimisation		X			X				X		X				X	5

III. METHOD

During the literature study, a list of green software process factors is identified. The factors are broken down into several elements and measurements. They are used to assess the greenness compliance and level of specific software. Fig. 1 presents the flow of this research, which shows the step-by-step activities carry out in this study.

The literature study has revealed five primary green factors: resource, people, organisation, technical and environmental. The qualitative interview is designed and constructed for field expert's or academician's validation of the factors. A pilot study and content validation are carried out

before the actual survey is conducted to ensure the reliability of the interview protocol. After completing the pilot study, the actual empirical study through the interview approach will be carried out. The informants are the developers, practitioners, or project managers of in-house software development companies and agencies in the public sector, which is the scope of this study.

Then, the next activity will be the data analysis. This activity will be carried out after the interview series have been completed. The data analysis will be conducted using Atlas.ti 8 software to determine the code and thematic of the interview finding. Another important task during the analysis is triangulation. Triangulation refers to measuring the reliability

of the study and the extent to which it is accurate for evaluating the concept or idea. One way to conduct the triangulation is through obtaining confirmation and approval from the interviewees on the findings.

As shown in Fig. 1, this research is conducted in four main phases. The following sub-sections explain each of the phases in detail.

A. Phase 1: Theoretical Study

The first phase is the theoretical study investigating the integration between sustainability elements and waste management for software processes. The references consist of the current journals, books, and proceedings that are being studied and investigated. This phase aims to identify the types of waste and the green factors that influence the software process. It also examines issues and problems concerns to green software process from literature. The result of this phase is the theoretical framework for the Green Software Process Assessment based on Sustainability and Waste Elements. The author in [17] shows the theoretical framework.

B. Phase 2: Empirical Study

In this phase, interviews will be conducted involving software practitioners to get real input from the industry. The informants of this study will be identified and selected from software practitioners, project leaders, information technology (IT) officers, software engineers and related IT positions. They must involve in the software development process with experience and knowledge in the field of software development. The selection of informants is based on the purposive sampling technique and aims to the organisations involved in in-house software development.

This paper presents the activity in this empirical study phase. In this phase, the interview questions are designed and constructed. The validity of the content is determined through expert validation. Furthermore, we conducted a pilot study to

ensure the reliability of the interview protocol. The pilot analysis is carried out and presented in this paper as the preliminary findings.

The next step in this phase will be the actual interview sessions with the identified informants. This phase will reveal and verify the influential factors in the green and sustainability software process and the current green practices in the actual software industry. In addition, software wastes formed during the software process's development and sustainability measurements will also be identified at the end of this phase.

C. Phase 3: Development of Green Software Process (Green SoftPro) Model

The third phase of this study is developing the proposed green software process model based on sustainability and waste elements. The proposed model also integrates the concepts of software waste based on inputs from theory and empirical findings. The model will identify and embed the relationship between green factors and sustainability measures that influence the greenness of software processes. In addition, the wastes specified at each phase of the software process and in line with the sustainability requirements will be considered during the model construction. The outcome of this phase is a Green Software Process (Green SoftPro) model.

D. Phase 4: Model Validation

The last phase of this research is to validate the Green SoftPro model, and the validation involves two steps. The first step is the verification by experts. The experts from the software developers and practitioners will be invited to review and validate the Green SoftPro model, which consists of green factors, elements and measurements. Further, the validation will be conducted through a case study approach. At least two software companies or government agencies will be invited to participate in the case study. The Green SoftPro model will be applied and will be validated in the actual software-industrial environment.

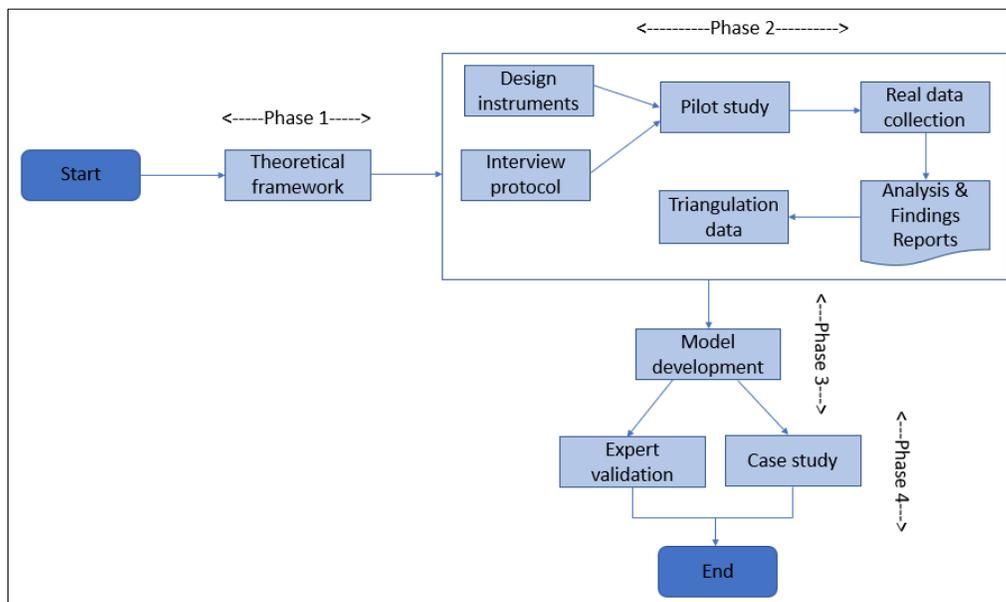


Fig. 1. Flow of the Research Activities.

IV. THE QUALITATIVE STUDY: DESIGN AND PILOT ANALYSIS

This study uses the interview method to identify the current practices, issues, and challenges for green and sustainable software processes. This paper focuses on the interview protocol design and pilot study.

A. Design and Development of Interview Protocol

The interview protocol is designed based on the literature review and the proposed theoretical framework (refer to [17]). The protocol consists of four main parts, Part 1: Demographics Background, Part 2: The Practice of Green Software Process, Part 3: Waste in Software Process, and Part 4: Green Factor and Measurement for Software Process.

Table II shows the items of part 1, part 2 and part 3, which are designed in the interview protocol. And Table III shows the items associated with part 4 of the protocol that consists of the green factor and measurement. The factors are resources, people, organisational, technical and environmental.

B. Pilot Study

In qualitative research, the purpose of a pilot study is to validate and reliable the interview instruments. The pilot study was conducted in three selected organisations. The interview session was performed in 2020, and each session took about one hour during a face-to-face interview. Each of the interview sessions was recorded with permission and transcribed were analysed using transcription software. All comments and recommendations were recorded and taken into consideration.

The interview sessions in this pilot study were carried out with three software or IT experts. The selection criteria of experts are based on expertise in software engineering and, more specifically, in the in-house software development process. The duration of working experience was also considered in selection criteria where they have working experience of at least five years in the industry. The experts' working experience is based on the years suggested by [69-70]. Table IV shows the expert's background who was involved in this interview's session.

TABLE II. INTERVIEW PROTOCOL: PART 1, 2 AND 3

Section	Items
Part 1: Informant's Background	<ul style="list-style-type: none"> Name of organisation Job position Duration of experience Job field and expertise
Part 2: The Practice of Green Software Process	<ul style="list-style-type: none"> Software development methodology used (Waterfall, Agile, Prototyping etc.) The implementation of Standard of Procedures (SOP) or process standards Green Practice Guidelines Best practices for software process activities Sustainability practices for software process activities The benefit of green practice
Part 3: Waste in Software Process	<ul style="list-style-type: none"> The understanding of software waste Software waste in software process activities. The solution of software waste.

TABLE III. INTERVIEW PROTOCOL: PART 4

Part 4: Green Factor and Measurement	
i) Resource	<ul style="list-style-type: none"> The usage of energy and paper The 3R practices (Reduce, Reuse, Recycle) The use of cloud computing
ii) People	<ul style="list-style-type: none"> The improvement of human resources Human health Roles in software developer Job satisfaction Sustainability practices in software developers
iii) Organisational	<ul style="list-style-type: none"> Effect/impact sustainability and green practices in organisation Sustainability and green practice awareness among employees in the organisation Commitment and involvement of various levels of stakeholders. The future investment with stakeholders.
iv) Technical	<ul style="list-style-type: none"> The situation of software failure The ability to change current needs and conditional change The adjustment to new requirements and modifications of software functions The expectation of software operation The adaptation of the different operating systems, hardware, and support system
v) Environmental	<ul style="list-style-type: none"> The usage of video conferencing or teleconferencing platforms The supportive of environmental The effect of the environment during development and maintenance The fully automated processes in software development

TABLE IV. INFORMANT'S BACKGROUND

Experts	Job Description	Working Experience (Years)	Sector
A	Software developer	13	Public university
B	Software developer	12	Public university
C	Software developer	11	Government agency

V. RESULT AND DISCUSSION

Based on the pilot analysis, the experts recommended that some items in the protocol be improved because they believed that they were not related to the green software process. Some of the items required correction for clarification. Other than that, the expert agreed with the defined content in the protocol.

Three interview scripts were translated and analysed. The findings consist of the three main issues: current issues in the software development process, factors and elements of the green software process, and elements of software waste.

A. Current Practice in Software Process Development

Current practice in software development is an essential part of ensuring the process runs smoothly and correctly. The findings of this pilot study discover that software practitioners apply sustainability practices indirectly in the software development process; for instance, reducing paper usage, writing more effective programming code, and using online data storage space.

B. Factors and Elements of Green Software Process

The pilot findings show that the green software measurements can be identified as effectiveness, cost reduction, job creation, competitiveness, usability, participation, ecosystem, resource, protection, human health, change, adaptation, endurance, satisfaction, happiness, and skills. The pilot informants propose and agree that these elements are appropriate and applicable for measuring the green software process.

Table V signifies the finding of new green factors and their elements through the pilot study. Based on this pilot study has revealed a further aspect, namely technology. The technology factor consists of two separate elements, namely IoT (Internet of Things) and IR 4.0 (Industrial Revolution), that relate to smartphones and automation technology in future software development.

C. Elements of Software Waste

The green process can be achieved through the elimination of waste. Thus, this pilot study has identified software wastes and is consistent with the literature [27, 44]; for instance, over-requirement, design, ineffective communication, rework, lack of knowledge, and expertise. These elements of wastes may affect the software development process in terms of budget, time constraint, and duration of completion.

During the pilot study, it discovers that most informants are not familiar with the meaning of software waste. After being explained further, they admit that particular wastes during the software development process include rework, loss of knowledge, and miscommunication. However, all these problems can be solved well and effectively in the end-stage.

TABLE V. NEW FACTOR AND ELEMENT FOR GREEN SOFTWARE

Factor	Element
Resource	<ol style="list-style-type: none">Cost reduction<ul style="list-style-type: none">Energy usagePaperless3R practices (reduce, reuse, recycle)Online storage
People	<ol style="list-style-type: none">Human capitalHuman healthRolesSatisfaction
Organisational	<ol style="list-style-type: none">AwarenessParticipationTop managementPolicy & legislationCulture
Technical	<ol style="list-style-type: none">AdaptationCapabilityCompatibilityEndurancePerdurability
Environmental	<ol style="list-style-type: none">Environmental preservationEnergy savingResource-saving
Technology	<ol style="list-style-type: none">SmartphoneAutomation

Furthermore, the pilot study proves the reliability of the protocol and the practicality of the interview instrument. The content of the interview protocol was validated and verified. The pilot study aims to test the questions' content, identify ambiguous questions, and test responses from small selected respondents. At the same time, findings from the pilot study are used to enhance the instrument via improving the questions to be more coherent or readable, accurate, presentable, justifiable, and relevant. Moreover, the time to answer the questionnaire was also identified in preparation for the actual interview session.

VI. CONCLUSION AND FUTURE WORK

This paper has presented the literature review on the green software process, the qualitative interview design, and the pilot findings. The literature study has revealed five primary green factors: resource, people, organisation, technical and environmental. The interview protocol was designed and constructed as discussed in this paper. Later, the pilot study was conducted to test the reliability and practicality of the interview protocol. The pilot analysis discovered the initial elements associated with the green software process. The current green practices in the software development process were identified through this pilot study.

Furthermore, it also discovers the influential green factors of the software process. In addition, this study reveals the relevant software wastes which the pilot informants have identified. For future work, the actual interview sessions are to be conducted until the information obtained is saturated. The data analysis will be performed to finalise the factors, elements, and measurements that will become the Green SoftPro model development input.

ACKNOWLEDGEMENT

The authors would like to express appreciation to the Faculty of Information Science and Technology, UKM, to provide research resources to encourage research activities among faculty members. The Malaysia Ministry of Higher Education funds this research under the Fundamental Research Grant Scheme (FRGS/1/2019/ICT01/UKM/02/1).

REFERENCES

- [1] Kern, M. Dick, S. Naumann, A. Guldner and T. Johann, "Green software and green software engineering – definitions, measurements, and quality aspects," in Proc. of First Int. Conf. Inf. Commun. Technol. Sustain. (ICT), 2013.
- [2] F. Khan and J. Porras, "A framework and a web application for self-assessment of sustainable green ICT practices in SMEs," no. September, pp. 1–21, 2018.
- [3] M. Dick and S. Naumann, "Enhancing software engineering processes towards sustainable software product design," Integr. Environ. Inf. Eur., 2010.
- [4] S. Naumann, E. Kern, M. Dick, and T. Johann, "Sustainable software engineering: Process and quality models, life cycle, and social aspects," Adv. Intell. Syst. Comput., 2015.
- [5] M. Dick and S. Naumann, "Enhancing software engineering processes towards sustainable software product design," in EnviroInfo 2010, Integration of Environmental Information in Europe, Proceedings of the 24th International Conference on Informatics for Environmental Protection, 2010.
- [6] C. Calero and M. Piattini, "Puzzling out software sustainability," Sustainable Computing: Informatics and Systems, vol. 16, no. Dec, pp. 117-124, 2017.

- [7] C. Calero and M. Piattini, *Green in software engineering*, Springer Cham, Switzerland 2015.
- [8] A. D. Alharthi, M. Spichkova, and M. Hamilton, "Sustainability requirements for eLearning systems: a systematic literature review and analysis," *Requir. Eng.*, 2018.
- [9] O. Gordieiev, V. Kharchenko, and M. Fusani, "Evolution of software quality models : usability, security and greenness issues." in *Proceedings of the 19-th International Conference on Computers (part of CSCC 15)*, pp. 16-20, 2015.
- [10] C. Marimuthu and K. Chandrasekaran, "Software engineering aspects of green and sustainable software: A systematic mapping study," in *Proceedings of the 10th Innovations in Software Engineering Conference*, pp. 34–44, 2017.
- [11] M. Al Hinai and R. Chitchyan, "Social sustainability indicators for software: Initial review," in *CEUR Workshop Proceedings*, 2014.
- [12] B. Penzenstadler, H. Femmer, and D. Richardson, "Who is the advocate? Stakeholders for sustainability," in *Proceedings of 2013 2nd International Workshop on Green and Sustainable Software, GREENS 2013*, 2013.
- [13] B. Penzenstadler, "Towards a definition of sustainability in and for software engineering," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC 2013)*, 2013.
- [14] E. Kern, M. Dick, S. Naumann, and T. Hiller, "Impacts of software and its engineering on the carbon footprint of ICT," *Environ. Impact Assess. Rev.*, vol. 52, pp. 53–61, Apr. 2015.
- [15] S. Naumann, M. Dick, E. Kern, and T. Johann, "The GREENSOFT Model: A reference model for green and sustainable software and its engineering," *Sustain. Comput. Informatics Syst.*, vol. 1, no. 4, pp. 294–304, Dec. 2011.
- [16] Kern E., Dick M., Johann T., Naumann S. "Green software and green IT: An end-user perspective," in: Golinska P., Fertsch M., Marx-Gómez J. (eds) *Information Technologies in Environmental Engineering*. Environmental Science and Engineering, vol 3. Springer, Berlin, Heidelberg, 2011.
- [17] S. R. Ahmad Ibrahim, J. Yahaya, H. Sallehudin and A. Deraman, "Green software process assessment: The theoretical framework," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 3, pp. 2011-2016, 2021
- [18] L. Ardito, G. Procaccianti, V. U. Amsterdam, M. Torchiano, A. Vetro, and A. Vetrò, "Understanding green software development: A conceptual framework," *IT Prof.*, vol. 17, no. 1, pp. 44–50, 2015.
- [19] M. Dick, S. Naumann, and N. Kuhn, "A model and selected instances of green and sustainable software," *IFIP Adv. Inf. Commun. Technol.*, vol. 328, pp. 248–259, 2010.
- [20] L. Ardito, G. Procaccianti, M. Torchiano, and A. Vetrò, "Understanding green software development: A conceptual framework," *IT Prof.*, vol. 17, no. 1, pp. 44–50, 2015.
- [21] S. A. Kocak, "Green Software Development and Design for Environmental Sustainability," *11th Int. Dr. Symp. an Empir. Softw. Eng.*, 2013.
- [22] S. S. Shenoy and R. Eeratta, "Green software development model: An approach towards sustainable software development," in *Proceedings of 2011 Annual IEEE India Conference: Engineering Sustainable Solutions, INDICON-2011*, 2011.
- [23] N. Amsel, Z. Ibrahim, A. Malik, and B. Tomlinson, "Toward sustainable software engineering," in *Proceeding of the 33rd international conference on Software engineering - ICSE '11*, pp. 976–979, 2011.
- [24] C. Calero, M. Angeles Moraga, M. F. Bertoa, and L. Duboc, "Quality in use and software greenability," *CEUR Workshop Proc.*, vol. 1216, pp. 28–36, 2014.
- [25] G. Casale et al., "Current and future challenges of software engineering for services and applications," *Procedia Computer Science*, vol. 97, pp. 34-42, 2016.
- [26] R. Kaur and J. Sengupta, "Software process models and analysis on failure of software development projects," vol. 2, no. 2, pp. 1–4, 2013.
- [27] T. Sedano, P. Ralph, and C. Péraire, "Software development waste," *2017 IEEE/ACM 39th Int. Conf. Softw. Eng.*, pp. 130–140, 2017.
- [28] H. Taherdoost and A. Keshavarzsaleh, "Critical factors that lead to projects' success/failure in global marketplace," *Procedia Technology*, vol. 22, pp.1066-1075, 2016.
- [29] Ardendu Mandal and S C Pal, "Identifying the reasons for software project failure and some of their proposed remedial through BRIDGE process models," *Int. J. Comput. Sci. Eng.*, 2015.
- [30] J. Yahaya, K. Raisian, S. R. Ahmad Ibrahim, A. Deraman, "Green software process based on sustainability dimensions: the empirical investigation," in *Proceedings of the 1st International Conference on Informatics, Engineering, Science and Technology (INCITEST 2019)*, 2019.
- [31] H. Acar, *Software development methodology in a Green IT To cite this version : Software development methodology in a Green IT environment*, Thesis, Université de Lyon, 2017.
- [32] S. Agarwal, A. Nath, and D. Chowdhury, "Sustainable approaches and good practices in green software engineering," *Int. J. Res. Rev. Comput. Sci.*, 2012.
- [33] S. S. Mahmoud and I. Ahmad, "A green model for sustainable software engineering," *Int. J. Softw. Eng. its Appl.*, vol. 7, no. 4, pp. 55–74, 2013.
- [34] E. Capra, C. Francalanci, and S. A. Slaughter, "Is software 'green'? Application development environments and energy efficiency in open source applications," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 60–71, 2012.
- [35] H. Acar, G. I. Alptekin, J. P. Gelas, and P. Ghodous, "Towards a green and sustainable software," *Transdisciplinary Lifecycle Analysis of Systems*, 2015.
- [36] M. Kramer, "Best practices in systems development lifecycle: An analyses based on the waterfall model," *Review of Business & Finance Studies*, vol 9, no. 1, pp. 77-84, 2018.
- [37] K. S. Ku Ibrahim, J. H. Yahaya, Z. Mansor and A. Deraman, "Towards the quality factor of software maintenance process: A review," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 3-4, pp. 115-118, 2017.
- [38] Z. H. Abdullah, J. H. Yahaya, Z. Mansor and A. Deraman, "Software ageing prevention from software maintenance perspective – A review," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 3-4, pp. 93-96, 2017.
- [39] C. C. Venters et al., "Software Sustainability: Research and practice from a software architecture viewpoint," *J. Syst. Softw.*, vol. 138, pp. 174–188, 2018.
- [40] A. N. Abdul Wahab, M. Mukhtar and R. Sulaiman, "Lean production system definition from the perspective of Malaysian industry," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 6 no. 1, pp. 1 - 11, 2017.
- [41] S. A. Dajadian and D. C. Koch, "Waste management models and their applications on construction sites," *International Journal of Construction Engineering and Management*, vol. 3, no. 3, pp. 91-98, 2014.
- [42] O. Al-Baik and J. Miller, "Waste identification and elimination in information technology organisations," *Empir. Softw. Eng.*, vol. 19, no. 6, pp. 2019–2061, 2014.
- [43] H. Alahyari, T. Gorschek, and R. Berntsson Svensson, "An exploratory study of waste in software development organisations using agile or lean approaches: A multiple case study at 14 organisations," *Information and Software Technology*, vol. 105, pp. 78-94, 2019.
- [44] O. Shmueli and B. Ronen, "Excessive software development: Practices and penalties," *International Journal of Project Management*, vol. 35, no. 1, pp. 13-27, 2017.
- [45] F. P. Santhiapillai and R. M. C. Ratnayake, "Identifying and defining knowledge-work waste in product development : A case study on Lean Maturity Assessment," in *Proceeding of 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 834-838, 2018.
- [46] S. Mujtaba, R. Feldt, K. Petersen, and E. Ab, "Waste and lead time reduction in a software product customisation process with value stream maps," in *Proceedings of the Australian Software Engineering Conference, ASWEC*, 2010.

- [47] T. Sedano, P. Ralph, and C. Péraire, "Removing software development waste to improve productivity," in *Rethinking Productivity in Software Engineering*, Berkeley, CA: Apress, pp. 221–240, 2019.
- [48] M.L. Sánchez-Gordón, and R.V. O'Connor, "Understanding the gap between software process practices and actual practice in very small companies," *Software Quality Journal*, vol. 24, no. 3, pp. 549-570, 2016.
- [49] S. R. A. Ibrahim, J. Yahaya, and H. Sallehudin, "Quality and sustainability dimensions toward green software product: A review," *Front. Artif. Intell. Appl.*, vol. 318, pp. 522–530, 2019.
- [50] S. Betz and T. Caporale, "Sustainable software system engineering," in *Proceedings of 2014 IEEE Fourth International Conference on Big Data and Cloud Computing*, 2014.
- [51] B. Penzenstadler, "What does sustainability mean in and for software engineering?," in *Proceedings of 1st Int. Conf. ICT Sustain.*, 2013.
- [52] C. Becker et al., "Sustainability design and software: the Karlskrona Manifesto," in *Proceedings of Int. Conf. Softw. Eng.*, vol. 2, pp. 467–476, 2015.
- [53] C. C. Venters et al., "Software sustainability: The modern tower of babel," in *CEUR Workshop Proc.*, vol. 1216, pp. 7–12, 2014.
- [54] B. Penzenstadler, "Where attention goes, energy flows-enhancing individual sustainability in software engineering," in *Proceedings of the Intl. Conf. on ICT for Sustainability*, 2020.
- [55] V. Grecu, R. I. G. Ciobotea, and A. Florea, "Software application for organisational sustainability performance assessment," *Sustain.*, vol. 12, no. 11, pp. 7–9, 2020.
- [56] J. A. García-Berna, J. M. C. de Gea, B. Moros, J. L. Fernández-Alemán, J. Nicolás, and A. Toval, "Surveying the environmental and technical dimensions of sustainability in software development companies," *Applied Sciences*, vol. 8, no.11, pp. 2312. 2018.
- [57] M. Razavian, G. Procaccianti, and D. A. Tamburri, "Four-dimensional sustainable e-services," in *EnviroInfo*, pp. 221-228, 2014.
- [58] O. Gordieiev, V. Kharchenko, and M. Fusani, "Evolution of software quality models: Green and reliability issues," in *CEUR Workshop Proceedings*, 2015.
- [59] R. Chitchyan, J. Noppen, and I. Groher, "What can software engineering do for sustainability: case of software product lines," in *Proceedings of 5th International Workshop on Product Line Approaches in Software Engineering, PLEASE 2015*, 2015.
- [60] C. Calero, M. Á. Moraga, M. F. Bertoa, and L. Duboc, "Green software and software quality," In: Calero C., Piattini M. (eds) *Green in Software Engineering*. Springer, Cham, 2015.
- [61] M. R. Shaharudin, J. Akbar, N. N. Zainal, S. F. Hassam, A. I. Zainoddin, and M. F. M. Nizam, "Factors that influence green practices adoption amongst logistics services providers," *Int. J. Supply Chain Manag.*, vol. 7, no. 6, pp. 242–253, 2018.
- [62] M. N. O. Sadiku, O. D. Olaleye, and S. M. Musa, "Sustainable engineering: an introduction," *Int. J. Adv. Sci. Res. Eng.*, vol. 5, no. 6, pp. 70–74, 2019.
- [63] B. Galleli, F. Hourneaux, and L. Munck, "Sustainability and human competencies: a systematic literature review," *Benchmarking: An International Journal*, 2019.
- [64] E. Jagroep, *Green software products*. PhD diss., Utrecht University, 2017.
- [65] P. Bozzelli, Q. Gu, and P. Lago, *A systematic literature review on green software metrics*, VU University, Amsterdam, 2013.
- [66] A. Kipp, T. Jiang, and M. Fugini, "Green metrics for energy-aware IT systems," in *Proceedings of Int. Conf. Complex, Intell. Softw. Intensive Syst. CISIS 2011*, pp. 241–248, 2011.
- [67] K. Erdélyi, "Special factors of development of green software supporting eco-sustainability," in *Proceedings of SISY 2013 - IEEE 11th International Symposium on Intelligent Systems and Informatics*, 2013.
- [68] E. Kern, M. Dick, S. Naumann, and T. Hiller, "Impacts of software and its engineering on the carbon footprint of ICT," *Environ. Impact Assess. Rev.*, vol. 52, pp. 53–61, 2015.
- [69] Z. Bukhari, J. Yahaya, and A. Deraman, "Metric-based measurement and selection for software product quality assessment: Qualitative expert interviews," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, pp. 223-231, 2019.
- [70] N. H. Z. Abai, J. Yahaya, A. Deraman, A. R. Hamdan, Z. Mansor, and Y. Y. Jusoh, "Integrating business intelligence and analytics in managing public sector performance: An empirical study," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 9, no. 1, pp. 172-180, 2019.
- [71] S. K. Sharma, P. K. Gupta and R. Malekian, "Energy-efficient software development life cycle - An approach towards smart computing," in *Proceedings of 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, 2015.