Applied Mathematics and Computational Intelligence
Volume 13, No. 4, 2024 [113-129]

An Assessment Mechanism for Integrated Software Sustainability Evaluation Model via Evaluation Theory

Ruzita Ahmad[1*], Mohammad Hafiz Ismail[2], Shukor Sanim Mohd Fauzi[3], Tajul Rosli Razak[4]

[1,2,3,4]College of Computing, Informatics and Mathematics, Universiti Teknologi MARA (UiTM) Perlis Branch, Arau Campus, 02600 Arau, Perlis, Malaysia

[*] Corresponding author: ruzitaahmad@uitm.edu.my

Received: 14 October 2024
Revised: 3 November 2024
Accepted: 5 November 2024

ABSTRACT

Evaluation of software sustainability aids in decision-maker's identification of the specific actions needed to guarantee sustainability for current and future generations. The prior approach to assessment focused on how the business environment was changing and using the high-quality software sustainability evaluation model (SSEM) affected those changes. Numerous well-established quality models, concepts, and understandings impacted on SSEM trends. These act as frameworks for developing software evaluations, the outcomes of which are applied to the assessment of generic software procedures. Therefore, this research aimed to use Evaluation Theory (ET) to create an assessment mechanism for an integrated Software Sustainability Evaluation Model (i-SSEM). This model encompasses evaluation criteria, targets, assessment processes, data-gathering techniques, synthesis techniques, and yardsticks. Nine criteria are presented in this study to evaluate software sustainability encompassing functional adequacy, dependability, performance efficiency, usability, security, compatibility, maintainability, portability, and impactibility. The use of the Quality Function Deployment (QFD) methodology, effectively classifies the recommended criteria into sustainable dimensions. A Goal Question Metric (GQM) is used to establish the software criteria by precisely specifying the aim, perspectives, and viewpoints of an evaluation of the sustainability aspects. By highlighting the unique evaluation mechanism for software products and processes and utilizing both quantitative and qualitative measurement techniques, this model improves the current SSEM.

Keywords: Assessment model; evaluation criteria; software sustainability evaluation; analytical-hierarchy process; weighted-sum method

1 INTRODUCTION

In modern software development, the assessment of software sustainability is critical because it ensures that the software meets the essential sustainability criteria. It is beneficial to provide decision-makers with an evaluation [1]. The goal is to help the stakeholders determine what needs to be done to guarantee software sustainability for the current and upcoming generations. Software sustainability evaluation assists the developer in understanding software sustainability requirements. As previously mentioned, software sustainability requirements require the integration of each sustainability dimension, including environmental, economic, and social aspects

[2]. Therefore, the evaluation of software sustainability encourages stakeholders to apply the software criteria in their development processes to achieve software sustainability and understand its effects. Previous research revealed that software sustainability evaluation models were assessed using the reasoning framework [3], the bottom-up approach [4], and the survey-based approach [6]. Most earlier models used the sustainability indicators' recommendations as a reference and mapping to evaluate the software sustainability criteria. According to this trend in assessment mechanisms [5], the sustainability achievement results previously presented in the general statement have been enhanced to include detailed descriptions, making them easier for the assessor to understand. Only [6] has used a survey-based methodology to contribute to the sustainability evaluation; the findings are displayed as a list of percentage values for every survey question.

The results indicate that, in the current business context, the provision of high-quality software sustainability measurement was not given sufficient weight by the assessment technique employed in the previous study. According to the [6], employing the evaluation requirements effectively is the key to ensuring an inclusive software evaluation model. However, this claim is supported by [7], which claims that the requirements of software sustainability are essential to the evaluation's success. Thus, to establish the quality of the evaluation, any conventional evaluation models or theories must be cited. However, their tailored assessment procedures are centered on general software process assessment, even if the prior software sustainability evaluation models referenced many common quality models, theories, awareness, and understanding as guidance to support software assessment. Since the developer's suggested models do not emphasize the integration of sustainability dimensions, this problem results from their neglect of sustainability requirements. The assessment method was originally built around "what" needs to be measured rather than "who," "when," "where," "why," and "how" in order to overcome this problem. As a result, the assessment procedure is predicated on the suggested software standards that have been determined in the chosen sustainability dimensions. These approaches do, however, recognize that the software sustainability assessment methodology they have proposed is an integrated evaluation model.

Regrettably, the purpose of the assessment was not made clear, so this limitation prevented them from evaluating the evaluation criteria for each sustainability dimension separately. Although the final assessment results in earlier work provided the assessor with guidance in the form of a continuous action and improvement plan [6]. Thus, the measurement scales and rates that are needed to report the sustainability achievement level for each evaluation criterion, dimension, and overall software assessment have not been provided. Rather, most of them executed the evaluation mechanism within the whole assessment process, providing the overall results statement and several explanations of the evaluation outcomes [8].

This study should address another issue related to assessment mechanisms in software sustainability evaluation models. To conclude, to attain sustainability in the new era, software development must be controlled and monitored using an extensive software sustainability evaluation model. It is necessary to build a thorough software sustainability assessment model on several standards to use it as a guide for software development. These standards support the evaluation model's central idea, serving as its major appliances. This is accomplished by building a comprehensive and comprehensive software evaluation model using approaches, methods, and tools. The following sections will cover the assessment of an i-SSEM using evaluation theory, a standard model, a tool, a technique, and a method.

2 INTEGRATED-SOFTWARE SUSTAINABILITY EVALUATION MODEL (I-SSEM)

The primary objective of the i-SSEM model is to integrate the environmental, economic, and social components of software sustainability by outlining the requirements for each dimension and providing a comprehensive evaluation mechanism for each criterion separately. The evaluation theory [9], which acts as the foundational theory and modifies the GQM approach, is used to build the i-SSEM model. Six (6) elements make up the evaluation theory: a yardstick, evaluation criteria, target, assessment process, and methods for obtaining and synthesizing data. Through the Systematic Literature Review (SLR) and empirical (exploratory) research, the theory was modified for this study. The literature was initially used to identify the crucial procedures, actions, methods, and standards. After that, a study was carried out using the survey technique to find out what Malaysian software practitioners thought about sustainable software development. The investigation's findings also aid in educating practitioners about software sustainability standards and procedures.

The Quality Function Deployment (QFD) technique was applied to systematically group the discovered criteria into sustainable aspects. This approach tries to highlight the effectiveness and efficiency of a systematic assessment mechanism [10], where the assessment procedure is essential to meeting the needs of software sustainability. The software criteria are categorized and organized using the QFD in accordance with their capacity to produce a high impact on sustainability aspects. Therefore, it makes sense to modify the QFD approach to expedite a methodical software assessment. Yoji Akao invented the QFD method in Japan in 1966. The Voice of the Customer (VoC) can be more methodically converted into new goods with the help of this strategy, especially when the House of Quality (HOQ) is involved.

Each criterion's measurement was created by using the GQM that was suggested by [10]. The question and metric, which are the primary tools in the i-SSEM assessment mechanism, were then developed using the aims as a guide. Data were gathered by document review, interviews, and observation by the respondents who completed the self-assessment task. Both quantitative and qualitative methods were employed by the i-SSEM model to determine the assessment measurement. Whereas the qualitative evaluation concentrated on the software process, the quantitative measurement was directed towards the software product. The multiple-criteria assessment is a component of the qualitative measurement, where the evaluation criteria may have varying effects on the development.

Subsequently, these synthesis methods used two MCDM approaches, AHP and WSM, to guarantee the precision of the software sustainability findings. Each criterion's global weight was determined using the AHP. Since software sustainability has varying effects on the environment, the economy, and society, this approach enables decision-making on all software sustainability criteria. Using both qualitative and quantitative measuring techniques, the AHP integrated the data to compare the effects of each criterion in an independent manner. WSM was used by allocating scores according to the weight values for the assessed criteria of the i-SSEM model. This approach makes decision-making easier by figuring out which criterion in each dimension has the best score and producing a score for the total accomplishments of the software project that is being assessed. Finally, the sustainability level was ascertained by contrasting the total performance score with the sustainability achievement index that was obtained from ISO/IEC 15504 [11]. However, AHP and WSM adaptations will not be presented holistically in this article.

2.1 Evaluation Theory: The Theories Underpinning Software Sustainability Evaluation Model

An effective evaluation model should be able to facilitate the assessor in decision-making, help the assessment accomplish its goal and serve its purpose, and offer continual opportunities for future development [12]. Moreover, bringing evaluation theory's constituent parts to use can result in the creation of an all-encompassing evaluation model that offers the user long-term results. This theory consists of the target, criteria, yardstick, data-gathering method, synthesis technique, and assessment mechanism [13]. Evaluation and improvement of software processes have made extensive use of the assessment theory. It is intended to assist organizations in evaluating long-term results, input, and output to facilitate continuous advancement and enhancement.

In [13] emerges subsequent, who apply the ideas to their research on the assessment of engineering design principles and lightweight software process assessment techniques in their measurement process. It is important to apply the fundamentals of evaluation theory to provide a thorough framework for software evaluation. The [13] applies the goal measurement for the introduction to the theory of evaluation is to help software practitioners create evaluation methods, develop the elements of the evaluation model, and adopt or adapt evaluation theory components to the demands and specifications of the evaluation model. A model that is deserving of review should be able to support the assessor in decision-making, help the assessment accomplish its goals, and offer ongoing actions for future development. Furthermore, using the evaluation theory's components can result in the creation of an extensive evaluation model that offers long-term results for the user.

2.2 Standard References for Software Evaluation: ISO/IEC 15504

In the sphere of the software industry, this framework the ISO/IEC 15504 is referred to as the model for process management and plays a vital role in advancing software process assessment. Two contexts of application are provided by the ISO/IEC 15504: process improvement for the organization's evaluation and capability determination for the supplier's evaluation. This reference model can be used by both user scopes to plan, manage, oversee, regulate, and enhance industry activities, comprising the development, procurement, operation, supply, and evolution to support their products and services. Nonetheless, these standard references provide users with instructions to establish process capability and incorporate a conformant process evaluation into a program for process improvement [11]. As an outcome, the technology company uses this strategy to evaluate internally how well they can accomplish the goal and enhance their workflow. This study adapts ISO/IEC 15504 to create the guidelines for sustainable software. The guidelines, which refer as the sustainability index, are used to illustrate the degree of sustainability achievement concerning the improvement plan and the improvement process. The level attainment of sustainability was developed using the six-maturity level of each process in ISO/IEC 15504, with each level's description tailored towards the goal of software sustainability. Additionally, the result of the conformant process assessment is characterized and analyzed in terms of the risks, weaknesses, and strengths associated with software development against an organizational unit using the ISO/IEC 15504 index level [11]. This sustainability index aids assessors in making decisions about ongoing action and enhancements by evaluating how well processes work toward achieving sustainability.

2.3 Goal Question Metric (GQM)

In Basili and Weiss developed the GQM methodology in the 1980s. It allows the user to create a hierarchical framework that acts as a guide for measurement by identifying the objective, question, and metrics. The approach entails articulating specific objectives, followed by inquiries that designated recipients must respond to facilitate the attainment of these goals. The metrics are established as a foundation for measurement. Formulating questions intended to evaluate the specified criteria is the foundation for creating metrics. One or more metrics are used to evaluate a single question to meet the predefined goals. With a goal, a question, and measurements arranged in a hierarchical structure, the GQM paradigm serves as a framework for users.

2.3.1 Goal

Conceptually, a goal is the main thing that has to be accomplished. Establishing the aim, viewpoint, and point of view for the circumstances can help define goals. It is organized according to a systematic methodology and presented using Basili's templates for simpler referencing. The first templates, which are made up of "purposes," aim to define, evaluate, forecast, and inspire various aspects of the topic under investigation—a process, a product, or a model—in order to make its qualities—like comprehension, evaluation, management, design, learning, and improvement—clear. "Perspective," is the following component, describes the necessity of analysing particular problems or aspects pertaining to expenses, efficiency, accuracy, flaws, modifications, product measurements, dependability, etc. It is important to do this analysis from the perspectives of various stakeholders, such as users, developers, managers, clients, and corporate organizations. The third aspect, “environment,” specifically addresses contextual variables such as process factors, people factors, problem factors, methods, tools, constraints, etc. [14].

All elements were used to determine the question and metric required to accomplish the specified goal. This study involves modifying Basili’s templates to align in developing sustainable software. The updated templates, which include a section on "purpose," aim to clarify the intended purpose of the subject being studied by characterizing, assessing, forecasting, and motivating various aspects of the process, product, or component. This includes comprehending, assessing, managing, engineering, learning, and improving the subject being examined. The second element, “perspective,” refers to the analysis of specific aspects or characteristics, such as cost, effectiveness, accuracy, defects, modifications, product metrics, reliability, etc., from the viewpoint of different stakeholders, including users, developers, managers, software engineers, customers, and maintainers. The third component, "environment," focuses on a variety of contexts, including economics, society, persons, technology, organizations, and the environment. The templates that have been altered to determine the goal within the specified measurement are shown in Table 1. The original templates keep the objective and point of view but modify the environmental element to align with the social, economic, and environmental dimensions of sustainability.

Table 1: Adapted Template to Define Goal

Elements	Description
Purposes	To characterise, evaluate, predict, and motivate the process, product, model, and metric in order to understand, assess, manage, engineer, learn, and improve it
Perspective	To examine the cost, effectiveness, correctness, defects, changes, product metrics, reliability, etc. from the viewpoint of the developer, manager, software engineer, user/customer, maintainer, etc.
Environment	The context of the environment, economy, society, individuals, technology, organizations, etc.

2.3.2 Question

The questions are presented at the operational level, which is the second hierarchical level. Based on various built-in characterization models, a series of questions is used to characterize the process of evaluating or accomplishing a particular aim. One goal is achieved by creating multiple questions. The categorization of the questions is in between aims and metrics. To aid in comprehension among the recipients, the questions are provided in a straightforward manner that is neither abstract nor overly complex.

2.3.3 Metric

The metric is also referred to as quantitative or qualitative measurement. This is followed by the type of data, which can be described in two forms: objective (quantitative) and subjective (qualitative). The measurement takes into account both the target and the viewpoint. For example, one can assess factors such as the number of document revisions, the duration of time that staff members allocate to a task, and the extent of a program's coverage. The items being measured, and their perspective are qualitative data. Examples of qualitative data include the legibility of a text, the degree of satisfaction, awareness, and similar factors. The quantitative metric allows various individuals to measure it and produce the same results. However, depending on who utilizes the measurements, the qualitative metric yields different results. On the other hand, qualitative measures work better with unstable or ambiguous items, whereas quantitative metrics work better with more developed objects. As a result, each question may have a different measure defined by the rules in the GQM-generated metrics [14]. A single measure should be able to address multiple questions related to the aim, rather than just one. After the input data is collected from the measurements, it is analyzed and synthesized using an appropriate technique to produce useful outcomes.

As previously explained, the synthesis approach is the focal point of concern in the assessment mechanism, where the obtained results are regarded as the ultimate outcomes. At the end of the evaluation, the yardstick, also known as the final results, provides the assessor with a clear indication of the measurement's achievement level. Hence, it is necessary to employ a synthesis approach to convert the data acquired during the measurement phase into a practical format that can be utilized for decision-making. Hence, the utilization of multi-criteria decision-making (MCDM) techniques, such as AHP and WSM, is appropriate for this objective. The next section provides more analysis of this matter.

2.4 Analytical Hierarchy Process (AHP)

AHP is a popular MCDM technique that offers the user many advantages in terms of data synthesis. The global weight for the assessment criterion used in the measurement is what AHP seeks to determine. Creating the pairwise comparison (PWC), evaluating the PWC, synthesizing the PWC, conducting the consistency analysis, and finally obtaining the global weight are some of the stages involved in generating the global weight. In addition, AHP is a measuring theory offered by PWC that creates priority scales based on expert assessment. The decision-makers can compare the alternatives with relative ease and apply weight to each evaluation criterion in an assessment thanks to PWC. The AHP is adaptable, thus organizing evaluation criteria into a hierarchy tree to display comparative views is a straightforward process. Decision-makers can quickly determine how many comparisons were made during the measuring process to the hierarchy tree. On the other hand, this approach offers the formula for calculating the total number of comparisons by AHP. In addition, decision-makers can systematically synthesize data using the AHP approach. This process converts the weight derived from human judgment into measurable values that can produce reliable findings and aid in decision-making.

2.5 Weighted Sum Method (WSM)

The most fundamental and traditional mechanism used in the MCDM method is the Weighted Sum Method (WSM). It is most frequently applied to situations involving only a single dimension. One popular technique for determining the final grade values for multiple criteria issues is simple additive weighting. The WSM method makes use of a set of data that comprises a variety of choices and standards. The formula that is given makes it possible to choose many alternatives and criteria in the best possible way to make decision-making easier. This is accomplished by giving each criterion a weight determined by the input of pertinent stakeholders.
The best alternative score is displayed as the WSM result. The score for each criterion for each declared alternative in the measurement procedure will be ranked using this method. Each alternative's total value equals the sum of its products, which is 1. The WSM method is usually used in combination with the AHP method to provide accurate results for the best selection criteria from a set of alternatives in supporting decision-making.

3 METHODOLOGY: AN EVALUATION MODEL OF AN I-SSEM

Six (6) elements make up an assessment mechanism component in i-SSEM: a yardstick, evaluation criteria, target, assessment procedure, and methods for obtaining and synthesizing data. The discussions of each component are given in the ensuing subsections.

3.1 Evaluation Criteria

The Goal-Oriented Software Sustainability Evaluation Criteria (GOSSEC), which was developed based on the conclusions from the theoretical and exploratory investigations, is the first part of the i-SSEM paradigm. The theoretical study’s conclusions encompass the primary realm of sustainable development. The study also defines criteria for software sustainability and looks at a number of software quality models and standards. The results of the exploratory study were derived from a survey given to software professionals in Malaysia. Four main steps were involved in developing GOSSEC: first, the establishment of software sustainability criteria; second, the classification of these criteria into sustainability dimensions; third, the formulation of software sustainability criteria; and finally, the modification of GQM (goal, question, metrics) to create goals, questions, and figures for each criterion. The GOSSEC underwent verification employing the expert review process to guarantee its accuracy, thoroughness, and clarity. Academics and software practitioners with knowledge and experience in software evaluation and sustainability domains are among the experts who participated. The following subsections elaborate on each of the GOSSEC phases.

3.1.1 *Phase 1: Identification of the* **Sustainable** *Software Criteria*

According to the findings of the theoretical and exploratory studies, researchers typically consult the Brundtland Commission Report in addition to drawing on their own experiences, theories, perspectives, and understanding of sustainability to determine the relevant criteria. When establishing the criteria, the researchers also referred to a number of standards and quality models, including ISO/IEC 25010, FURPS, McCall, Boehm, and ISO/IEC 9126. As a result, the exploratory investigation was used to apply and test the recommended criteria that were identified from the theoretical study. According to the results of the exploratory study, a total of 55% of the respondents used ISO/IEC 25023, ISO/IEC 25010, and ISO/IEC 9126, among other standard quality models, to determine the sustainability criterion. They claim that in addition to other factors like the software category, integrity level, and user needs, the system's, product's, or a component of the product's inherent features determine the credibility of these standard references. Moreover, they contended that these standards offer crucial benchmarks that might significantly improve software's long-term durability. Eight (8) software sustainability criteria were therefore developed following an exploratory investigation and analysis of the literature reviews: functional suitability, performance efficiency, dependability, compatibility, maintainability, security, usability, and portability.

The social, economic, and environmental facets of sustainability will be impacted by any software developed and used. As had already been established, the degree to which information and communication technologies are employed is referred to as "impact" in the area of software sustainability. Some research indicates that the sustainability of software was impacted by the third-order effect, which is the long-term influence caused by the cumulative impacts (of the first and second-order effects). The "impact" issue needs to be brought up in order to make sure that software practitioners are aware of how software development impacts the social, economic, and environmental aspects of sustainability. Thus, this study proposes "impactibility" as a software sustainability criterion.The ISO standards and software quality models, which recommended terminating the word with "-ility," were cited when using this phrase as one of the criteria.

When building sustainable software, the "impactibility" refers to the degree of user acceptance. It evaluates software practitioners' awareness and adoption factors about the environmental, economic, and social dimensions in addition to their acceptance of controlling the risk of sustainability. The idea of impact, which refers to a human interaction and recognition of a procedure or circumstance within a particular context, such as the three sustainability dimensions, is directly related to the issue of human/user acceptance, as previously discussed. Measuring these aspects is necessary to achieve sustainability since they have interrelated implications that begin at the software development stage. Consequently, it is believed that user acceptance is a suitable and significant as the sub-criterion of "impactibility".

Software practitioners in Malaysia have tested and applied the "impactibility" criterion. In their positive response, every respondent (100%) suggested "impactibility" as a new standard to help achieve software sustainability. Developing software that can yield beneficial outcomes for the environmental, economic, and social dimensions is important, as indicated by the fact that 95% of respondents believed that human/user approval influences the criterion. Consequently, functional suitability, dependability, performance efficiency, usability, security, compatibility, maintainability, portability, and impactibility were the nine (9) software sustainability criteria that were found in this study.

3.1.2 Phase 2: Organisation of the **Sustainable** *Software Criteria*

Sorting the software sustainability criteria into groups based on how well they can contribute to sustainability dimensions is the aim of this organization. The criteria were arranged according to sustainable dimensions by using the results of exploratory and theoretical research. Various standards, quality models, experts, and the experiences, beliefs, and perspectives of the researchers and practitioners about sustainability were taken into consideration when organizing the criteria in Phase 2. Putting the criteria in the right order within the sustainability aspects will increase the positive effect and produce software that is long-lasting in the end. However, the arrangement of the software sustainability criteria will not be covered in detail in this paper. Using the QFD based on the metric measures of the ISO/IEC 25023 standard, each detected criterion that has been organized into a series of quantifiable sub-criteria was aligned to each sustainability dimension. The metric was selected by software practitioners due to its capacity to assess software from an environmental, economic, and social standpoint. Responses were recorded on a seven-point Likert scale, with the interval values ranging from "extremely disagree" to "extremely agree."

The QFD tool was then used to do a systematic organization using the data that was gathered from the respondents in the exploratory study. The Voice of the Customer (VoC) can be efficiently translated into creative goods through the application of the QFD, a systematic quality strategy. The top (HOWS) denotes the software sustainability requirements, while the VoC (WHATS) symbolizes the sustainability characteristics that match with the relevant technical answer. The importance weights or rating scales provided by the stakeholders are included into the modified House of Quality (HoQ) framework, which is used to illustrate these linkages. The rating scales are represented by the connection matrix in the center of the HoQ. The QFD method for the criteria organization was then used to map these results, accounting for each sustainability dimension. Following that, a number of professionals verified the analysis's findings during the verification session. The recommendations offered by the experts during the verification session were subsequently taken into consideration while revising the criteria organization's results. The outcome of the criterion arrangement for this investigation is shown in Figure 1.

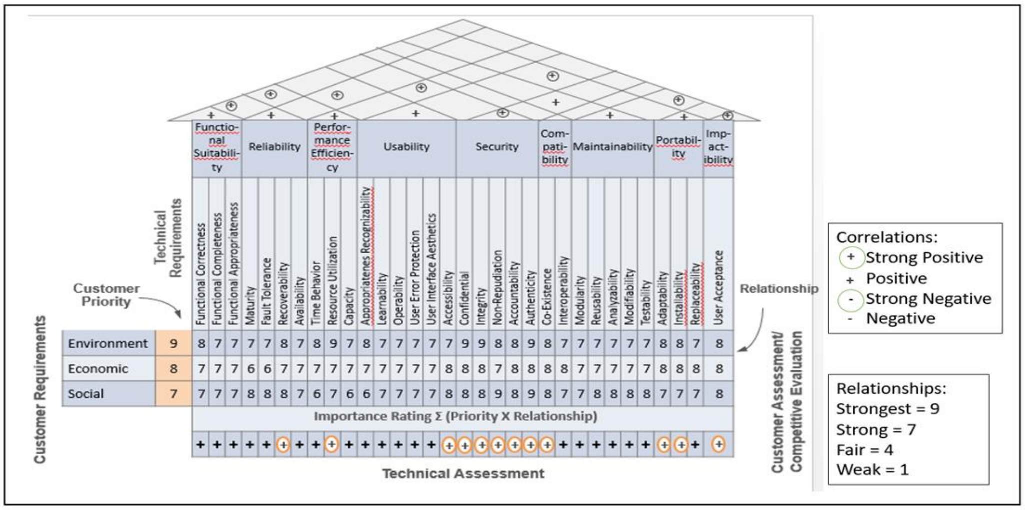

Figure 1: Final criteria organization

According to Figure 1, several factors, including dependability, performance efficiency, security, compatibility, portability, and impactibility, have been scored as strongly positive. Positive ratings were given to the remaining categories, which included functional suitability, usability, and maintainability. The weight that participants assign to the HoQ matrixes is implied by the QFD's automatic importance rating calculation, which is based on the total customer priority.

3.1.3 Phase 3: Definition of the **Sustainable** *Software Criteria*

Finding the specification of software sustainability criteria is the main goal of this phase. Each criterion was defined by the organization using the outcomes of the Goal-Oriented Software Sustainability Evaluation Criteria (GOSSEC) second phase [15]. Based on the ISO/IEC 25023 standards, the software sustainability requirements for the software product were created. These criteria include functional suitability, dependability, performance efficiency, usability, security, compatibility, maintainability, and portability. Furthermore, an exploratory study was conducted to apply and assess the theoretical study's definition of the "impactibility" criteria for the software process. The definition of the identified software sustainability criteria in this study has been verified through the verification session.

3.1.4 Phase 4: Determination of Goal

The GQM templates were utilized to ascertain the objective of every criterion that was identified. To solve the shortcomings of earlier works, the modification of GQM is essential. By outlining the goals, viewpoints, and context of software sustainability concerning the social, economic, and environmental spheres, it accurately establishes the objective. The goal of software measurement is guided by the GQM can solve the drawbacks of previous studies that just concentrate on determining what has to be measured. GOSSEC comprises thirty-two (32) sub-goals and nine (9) goals, which are determined by the software sustainability criteria and sub-criteria. Earlier, software sustainability criteria were evaluated holistically, with specific software products and processes being evaluated in respect to the environmental, social, and economic aspects. Figure 2 depicts the goal structure of GOSSEC's software sustainability criteria.

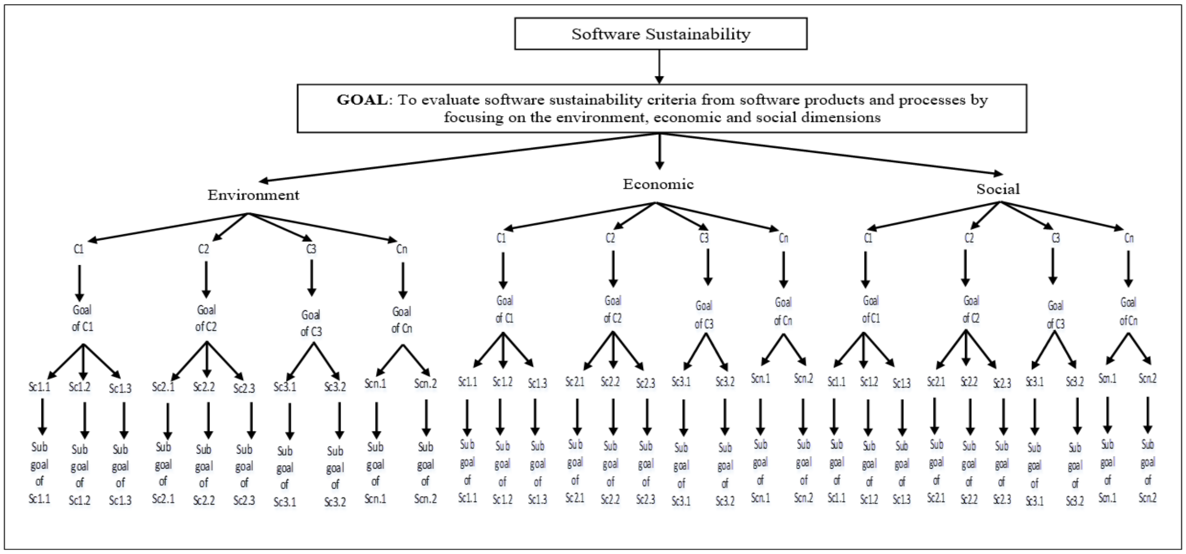

Figure 2: Goal structure of software sustainability criteria

3.2 Evaluation Target

Determining and justifying the evaluation aim is essential, as is describing the target environment. Software items and procedures were included in the evaluation aim of this study. Evaluators can better comprehend the type of software that needs to be evaluated by defining the target evaluation. The overall objective of creating software for sustainability was split into three (3) qualitative metrics and eighty-six (86) quantitative metrics in this study, which focused on software processes and products, respectively. As a result, eighty-nine (89) of the nine (9) defined software sustainability criteria were employed as evaluation metrics. Measurement techniques included both qualitative and quantitative approaches.

3.3 Data Gathering Technique

Numerous methods are used to gather the input data, such as document examination, interviewing, and observation. According to the exploratory study's findings, software practitioners in Malaysia most frequently employ these numerous data-gathering strategies because they can enhance the comprehension of the assessment team and better validate the assessment outcomes [16]. There are two categories of documents in the document review: direct and indirect. While the indirect documents are the results of other activities, the direct documents are the concrete results of the practice's application. Conversely, the methods of observation and interviewing allow the assessment team the chance to engage with a qualified individual directly and gain further insight into the subject matter [16].

3.4 Assessment Process

Pre-assessment, assessment, and post-assessment are the three (3) primary stages of the assessment process, which consists of several tasks. Figure 3 shows the assessment processes for the i-SSEM model and is further detailed in the next subsection correspondingly.

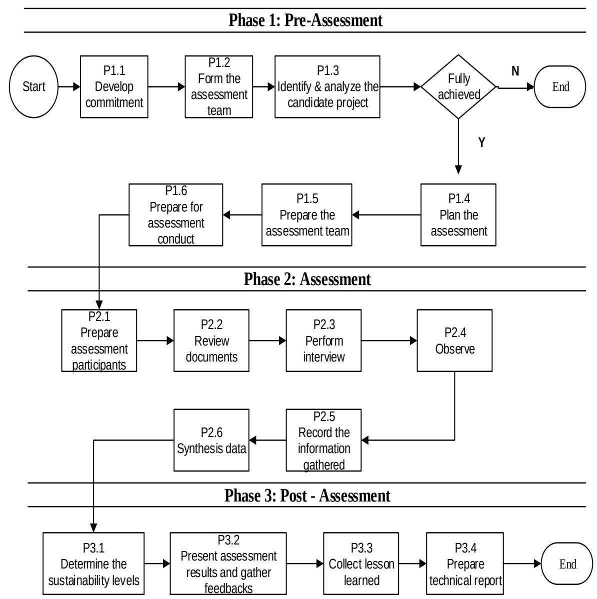

Figure 3: Assessment Processes for the i-SSEM model

3.4.1 *Phase 1: Pre-Assessment*

The pre-assessment entails the stakeholders establishing a commitment to use the self-evaluation method and reaching a consensus on how to conduct the assessment. The assessment team is established by deliberately selecting competent individuals who are eager to engage in the review process with commitment and consensus. Next, the task entails identifying and evaluating a suitable potential project for assessment. Subsequently, the assessment planning process is executed to outline the associated tasks, necessary resources, and time frame. At the end of this stage, the evaluation is appropriately organized. Meanwhile, project documentation is prepared for the upcoming phase.

3.4.2 *Phase 2: Assessment*

The primary stage of the evaluation process is this assessment. Using the provided assessment form, the main task entails assessing the software development of the software project in the case study. Interviews, observations, and document reviews are used to do this. The information pertaining to the case study project is used to answer the input for the quantitative measurement, which is the concrete output that results directly from the application of practice. Interviews, document reviews, and observational methods are used to collect the data. Participants in the measurement are required to rate their acceptance of the software sustainability activities by giving each practice a number since the measurement's input is obtained through observation and interview methods. Next, the participants are asked to assign a score to the important criteria related to the environmental, economic, and social dimensions

3.4.3 *Phase 3: Post Assessment*

The post-assessment is the last stage of the software process assessment. In this stage, the results of the evaluation procedures are acquired.

3.5 Synthesis Technique

Once the necessary data are gathered, the synthesis technique is used to synthesize them. By taking weight allocation into account, the synthesis technique is applied in the i-SSEM model to improve decision accuracy and consistency. The measures that are involved are essentially a combination of the best MCDM synthesis approaches, like AHP and WSM, both quantitatively and qualitatively. First, the global weight of each assessment criterion is determined using the AHP approach. The quantitative data scores are computed using the adopted metrics, which are also referred to as the score of evaluation criteria (i-child). The scores of their parents are then determined using these scores. To find the score for the qualitative data evaluation criteria (i-child), utilize the assessment form. Ultimately, the software project's parent/root score is determined, and the degree of accomplishment for every assessment criterion, dimension, and overall sustainability achievement is carried out.

3.6 Sustainability Index: Yardstick

To help users map the software assessment's outcomes toward sustainability accomplishment, this study proposes a sustainability index. The suggested index is based on the ISO/IEC 15504 standard, commonly known as software process improvement and capacity determination. The description level improves the user's comprehension of sustainability attainment by providing a thorough explanation of each accomplishment level. This index allows for the final results to be determined by taking into account the sustainability dimensions, the software project's overall success, and the degree of achievement of each criterion. In conclusion, by consistently planning and acting to improve software development techniques, this index helps users improve their software projects. Table 2 displays the index of sustainability.

Table 2: Sustainability Index

Rating Scale	**Achievement Level**	**Level Description**
Fail / Not Achieved (0–5%)	Level 0 Incomplete	Sustainable achievement has failed. There are no recognizable work products or process outputs.
Strongly Poor / Very Slightly Achieved (> 5%–10%)	Level I Performed	The process's purpose is generally achieved. The achievement may not be thoroughly planned and tracked. There are identifiable work products for the process or product, and these testify to the achievement of the purpose.

Poor / Slightly Achieved (> 10%–25%)	Level II Managed	The process delivers work products according to specified procedures and is planned and tracked. Work products and processes match the specified standards and requirements.
Adequate / Partially Achieved (> 25%–50%)	Level III Established	The process is managed using a defined process that is based on good software engineering principles. The process implementations are approved, tailored to a standard version, and documented to ensure sustainability.
Good / Largely Achieved (> 50%–85%)	Level IV Predictable	Within a specified sustainability dimension, the defined process is performed consistently in practice. Each dimension is evaluated to achieve the process goals.
Excellence / Fully Achieved (> 85%–100%)	Level V Optimising	The process's performance is optimised to meet current and future requirements, as well as to achieve the sustainability goal.

4 DISCUSSION AND CONCLUSION

The evaluation theory served as the basis for the construction of the six-component i-SSEM model. In addition, the GQM approach was modified to ascertain the objective of every criterion through the accurate identification of the aims, viewpoints, and points of view in the environment about the attainment of software sustainability. The measurement criteria were then taken from ISO/IEC 25023 and presented for each criterion, separately, as questions and metrics. This model provides a synthesis method that uses the AHP and WSM methodologies to provide correct results. Ultimately, the outcomes were showcased for every assessment criterion, aspect, and total software accomplishment. In order to ascertain the sustainability level and ongoing improvement, these outcomes were specifically referred to as the sustainability achievement index.

Generally, this research could have a significant influence on software engineering, particularly in the areas of software sustainability and evaluation. It can offer a strong basis for the next studies that seek to progress software development toward sustainability. Practically, this strategy offers significant advantages. Software engineers are very knowledgeable about both the methods they employ in their projects and the sustainability of the software they create. Furthermore, by providing appropriate instructions, this model can help the assessor in doing the program evaluation. Because it delivers a novel perspective on software sustainability and its long-term viability, this model benefits academics by improving their studies and boosting their evaluation.

4.1.1 *The Development of GOSSEC*

Particularly, theoretical and exploratory research as well as the improvement of expert evaluations' comments and proposals had an impact on the GOSSEC's development. A set of requirements for software sustainability was created as a result of the study's findings [15] These standards help to direct software developers in creating software that prioritizes the social, economic, and environmental aspects. Six measuring aspects are offered by the GOSSEC, and the criteria are assessed according to "what, who, where, when, why," and "how" to measure. These contributions were made use of by modifying the GQM technique to precisely ascertain each criterion's measurement target and by implementing the QFD tool to methodically arrange the criteria into sustainable dimensions [15]. The QFD is capable of rearranging the software criteria into the social, economic, and environmental. For every criterion that the HOQ structure represents, the QFD aids in the construction of the correlation matrix. This tool supported the elements of measurement (the WHATS), where the requirements are matched with the appropriate technical response at the top (the HOWS). Moreover, the instrument methodically arranged the sub-criteria into sustainable aspects while considering its assessment standards. The GQM approach employed the components of measurement, namely "what, who, when, where, why, and how," to evaluate each criterion and sub-criterion. Basili's templates were modified according to the research findings in order to enhance the measuring elements. This significantly assists in overcoming the constraints of prior models, which mostly focused on quantifying "what" rather than "who, when, where, why, and how". The inclusive goal definition process may oversee the measurement mechanism for formulating questions and evaluating metrics for software sustainability assessment.

4.1.2 *The Highlighted Synthesis Technique*

The incorporation of the synthesis approach in *i*-SSEM enhances the assessment mechanism in this model in comparison to the prior software sustainability evaluation model documented in the literature. The i-SSEM model heavily relies on data synthesis approaches such as AHP and WSM, which are critical for achieving accurate results. The accuracy of the results was attained by considering the weight provided by the evaluators during the evaluation process. This is critical because the assessment covers several variables, each of which may have different levels of significance for software sustainability. The AHP was used to calculate the overall weight of each criterion. This process required obtaining a final consensus from all assessors participating in the assessment. The WSM technique aids in calculating the ultimate score for each specific criterion, as well as contributing to the overall success of the assessment process. The integration of these two methodologies in data synthesis enhances the comprehensiveness of the assessment mechanism carried out by the *i*-SSEM model, leading to accurate results.

4.1.3 *Recommended Sustainability Index*

A crucial contribution to this research was the sustainability achievement index. When performing the software sustainability assessment, this index serves as a benchmark. It is typically used as the baseline in software process measurement and was derived from ISO/IEC 15504, commonly known as Software Process Improvement and Capability Determination. Beginning with "Level 0: Incomplete, Level I: Performed, Level II: Managed, Level III: Established, Level IV: Predictable, and Level V: Optimising,". The sustainability accomplishment index presents six levels of sustainability achievement. By using these indices, the assessor may continue to their organization's continuous enhancement with a focus on the sustainability achievement level. This contribution is advantageous because it rectifies the inadequacies of earlier research, which did not provide a sustainability index to help users relate the outcome to the sustainability achievement. As a result, this work offers the i-SSEM model as a conventional sustainability assessment approach for creating highly qualified software in the contemporary corporate setting. The i-SSEM model offers an independent evaluation mechanism for each of a set of software sustainability criteria, which challenges the constraints and requirements of the earlier models. To solve the shortcomings of the earlier study, a comprehensive assessment mechanism for software sustainability is needed. Thus, the Evaluation Theory components supported the i-SSEM model, while ISO/IEC 25023 and the impactibility criteria recommendation supported the measurement criteria to support the evaluation of user acceptance towards software sustainability practices.

ACKNOWLEDGEMENT

We would like to thank the Universiti Teknologi MARA, Shah Alam, Malaysia, for providing the funds for this research through the Geran Penyelidikan Myra (GPM) entitled "An Enhancement of Integrated-Software Sustainability Evaluation Model Towards Service-Oriented Architecture Paradigm, 600-RMC/GPM LPHD 5/3 (152/2021)."

REFERENCES

[1] T. Kehrer and B. Penzenstadler, "An Exploration of Sustainability Thinking in Research Software Engineering," in 7th Intl. Workshop on Requirements Engineering for Sustainable Systems (RE4SuSy), pp. 1-8, 2018.

[2] C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. M. Easterbrook, B. Penzenstadler, N. Seyff, and C. C. Venters, "Requirements: The key to sustainability," IEEE Software, vol. 33, no. 1, pp. 56–62, 2016.

[3] E. Kern, D. Markus, S. Naumann, G. Achim, and T. Johann, "Green Software and Green Software Engineering – Definitions, Measurements, and Quality Aspects," in ICT4S ICT for Sustainability. Proceedings of the First International Conference on Information and Communication Technologies for Sustainability, ETH Zurich, pp. 87–94, 2013. Available online: ecollection.library.ethz.ch.

[4] K. Heiko, D. Domis, T. Goldschmidt, and P. Vorst, "Measuring architecture sustainability," IEEE Software, vol. 30, no. 6, pp. 54-62, 2013.

[5] K. Heiko, "Sustainability evaluation of software architectures: A systematic review," in CM SIGSOFT Symp.—ISARCS on Quality of Software Architectures—QoSA and Architecting Critical Systems—ISARCS, QoSA'115, 2015.

[6] Software Sustainability Institute, "Software Sustainability Evaluation," 2016. [Online]. Available: http://www.software.ac.uk/.

[7] J. Sachs, C. Kroll, G. Lafortune, G. Fuller, and F. Woelm, Sustainable Development Report 2022, Cambridge University Press, 2022.

[8] E. Kern, L. M. Hilty, A. Guldner, Y. V. Maksimov, A. Filler, J. Groger, and S. Naumann, "Sustainable Software Products - Towards Assessment Criteria for Resource and Energy Efficiency," Future Generation Computer Systems, pp. 199-210, 2018.

[9] E. Kern, L. M. Hilty, A. Guldner, Y. V. Maksimov, A. Filler, J. Groger, and S. Naumann, "Sustainable Software Products - Towards Assessment Criteria for Resource and Energy Efficiency," Future Generation Computer Systems, pp. 199-210, 2018.

[10] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, and A. Trendowicz, "Linking software development and business strategy through measurement," IEEE Computer Society, vol. 43, no. 4, pp. 57–65, 2010.

[11] ISO/IEC15504, "Information Technology Software Process Assessment," Geneva, Switzerland: International Organization for Standardization, 2004.

[12] G. A. Szalkowski and C. Johansen, "Defining and measuring the effects of digital technologies on social sustainability: A systematic literature review," Sustainable Development, vol. 32, no. 3, pp. 1678-1699, 2024.

[13] M. Scriven, "The Nature of Evaluation Part II: Training," Practical Assessment, Research & Evaluation, vol. 6, no. 12, 1999.

[14] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, and A. Trendowicz, "Linking software development and business strategy through measurement," IEEE Computer Society, vol. 43, no. 4, pp. 57–65, 2010.

[15] R. Ahmad, F. Baharom, and A. Hussain, "GOSSEC: Goal Oriented Software Sustainability Evaluation Criteria," Journal of University of Babylon for Pure and Applied Sciences, vol. 27, no. 1, pp. 387-407, 2019.

[16] R. Ahmad, "Software Sustainability Practices and Awareness Amongst Software Practitioners in Malaysia: An Exploratory Study," in 2022 2nd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS), Yogyakarta, Indonesia, 2022, pp. 192-197.