

SQ-Framework for Improving Sustainability and Quality into Software Product and Process

Kamal Uddin Sarker¹

Aziz Bin Deraman²

Faculty of Ocean Engineering
Technology and Informatics
University Malaysia Terengganu
Terengganu, Malaysia

Raza Hasan³

Department of Information
Technology, Malaysian University of
Science and Technology
Selangor, Selangor
Malaysia

Ali Abbas⁴

Department of Computing
Middle East College
Knowledge Oasis Muscat
Oman

Abstract—Sustainability is one of the most important quality factors and it integrates some other quality factors in the product too. Moreover, it makes an effective workflow and improves user satisfaction. A manager can meet the target by controlling a project but sustainability is more versatile. Quality factors are the measuring criteria of a product while sustainability drives to make the quality product, efficient project, and successful organization so it is a package of strategy, tasks, processes, technologies, and stakeholders. It is observed that there is a lacking of sustainability practice in software engineering like other engineering communities. There are many software developing models that exist with limited scope in quality control for sustainability. Given the aforementioned viewpoint, this research proposes a new software project management framework, “SQ-Framework”. Its hybrid structure consists of the features of methods, quality models, and sustainability. The execution guidance of “SQ-Framework” is provided according to “Karlskrona manifesto”. A manager can use the framework to improve the management process of a project, a developer can integrate quality factors with sustainability into the product, an executive could be motivated to integrate quality and sustainability strategy in the organization, and the users get inspiration to practice sustainability.

Keywords—Software project management; sustainable project; sustainable product; sustainable and quality model; system development methodology

I. INTRODUCTION

A project is a temporary endeavor that generates a product or service within a fixed budget and time. Project management is a systematic approach that has a set of tasks, processes, guidelines, technologies, and stakeholders to meet the goal of the project on time, in budget with better user satisfaction. User satisfaction is measuring based on the quality criteria of the projects and sustainability is one of them. The evolving concept “quality” is varied according to the application domain and quite tricky to define. A product is known as a qualified product when it has distinct features than a similar type of other products. It aggregates multiple measuring attributes of this domain. It is a misleading and risky word and multi-attributes measuring system required to qualify a software system [1]. If without domain knowledge no one can define a terminology; without proper definition, you can not measure the product's quality, and without knowing measures can not do quality control of a system [2]. The software functionalities

are expressed in quantity (e.g. source line of code or functional points) that specify the size of software and non-functional attributes are qualitative. Software quality measurement includes both in measures (completeness of the system and user satisfaction factors) and project management integrated quality attributes to a software product. Quality management is a process that aims to make sure the existence of quality factors in a soft product. Quality measures of the quality models and standardization companies guidelines help to gain quality products. The initial measurement of a system depends on the expected quality standard of the customer or user. Moreover, it also meets regulatory quality requirements from the developing teams. Quality management is not only limited to testing activities before released to the market but also maintains critical evaluation processes in different phases [3]. It also guides to develop a quality management culture in the enterprise [3]. Quality management consists of quality planning: method of measuring quality goals, quality control: defect identification and correction, and quality assurance: a set of actions in process management to ensure quality.

Sustainability becomes important in global concern due to the huge consumption of energy for the industrial revolution of the 20th century and in 1930s economists developed sustainable models for non-renewable resource management. Day by day its' value is adding in our life due to technological advancement and smarter lifestyle. In the 21st century, its importance is extending in personal, social, and corporate life. Mitchell Grant simply defined “meet present needs without compromising future needs” [4]. Sustainability ensures the wise-utilization of environmental, social, and economic resources that could offer the same for an upcoming generation [5]. Social sustainability promotes wellbeing by developing processes and structures in society. Sustainable business culture ensures human rights, fairness, diversity, and wellbeing practice in the enterprise. Reducing wastage of time, effort, and money of an organization is called economic sustainability; moreover, it suggests implementing technological applications to make the highest productivity. Environment-friendly farming and foresting practice by using sustainable energy and technology is the practice of environmental sustainability. Innovation in applications and software engineering for financial, social, and environmental sustainability is a common practice. But today researchers focus on integrating green soft features in the system and sustainability practice in the project.

Sustainability is a special quality factor in an application to reduce energy consumption by stopping the unnecessary processing cycles, implementing power-saving mode, applying efficient data structure and algorithm, and using green technology. Information technology (IT) engineers can bring Innovation in software with sustainability features and play the golden role in reducing carbon emissions. Business transformation with green information and communication (ICT) enhances economical sustainability know as ICT for sustainability. The ethical practice of ICT can improve social harmony and cultural exchanges for peace and happiness too. This article shows more importance of sustainability because it integrates other quality factors like re-usability, efficiency, and cost-effectiveness. The proposed framework illustrated a way of integrating sustainability with other quality factors without conflict; as well as the way of practice and keep up the quality of product and project. it brings innovation in cost-cutting, enhancing competitive advantages, and adding the value of software project [6].

“Introduction” consists of the working area of this research and the “literature review” presents the importance of research on sustainability and quality control practice in software industries. It also includes a comparison study on practicing quality models of the software product. Section-3 illustrates a framework for quality control and sustainability practice in software farm that is a hybrid framework of system development methodology and quality model. The aim of this framework is enhancing sustainability practice into the product, and project besides keeping quality control; so it is named "SQ-Framework" to The way of implementation for the proposed model is expanded and elaborated in the followed section immediately. Section five consists of a comparative analysis of the framework with system development approaches and quality models. Section 6 carries the guidelines for using the framework according to the “manifesto”. The last section makes the conclusion of the study with recommendations on future studies.

II. STATE OF THE LITERATURE

A. Role of Technology in Project Management

Advancement of technology changes the processes and methods of project management. Project management institute (PMI) has been publishing regular bulletins on project management opportunities, challenges, and technologies. The latest analysis showed higher importance in technical, leadership, business, and digital skill for measuring talent in project management [7]. It noticed that a single business, project, or big idea is not enough in this wrap-speed word to keep an organization at the top constantly. Moreover, your brilliant strategy or amazing product idea could not be a success for supply chain disruption or new technology, and projects can fail fast. Disrupting technology like AI and machine learning can be run by only smart people. The rapid technological change increases challenges into the process of conversion idea to reality. Moreover, it suggests three tenets for resolving the aforementioned challenges: adaptability or agility in process, regular training, and automatic design skills. Project management wants to turn their idea to reality but there is no super-secret formula to make project success; so change

management approach, design thinking approach, hybrid management approach, and agile approach become more popular [8].

B. Reasons for Project Failure

Researchers, academicians, professionals, standardization organizations, and certification vendors of this domain are working to improve IT project success rate but till now about 45% project is challenging, 36% success, and 19% fail (2018) [9]. Recently, the success rate is increasing by benefits realization management (BRM), which is a powerful project handling approach and it aligns projects, programs, and portfolios to an organization’s overarching strategy [10]. A project does not fail for a single task, person, or process; but there is a cause-effect relationship among task, person, and process. A task is defined and executed by a person or team and the task is executed according to the predefined process so the outcome of the task could not be measured by individual parameters. A project may fail if there is a lack in i) the project and organizational strategy, ii) established accountable result measuring, iii) unambiguous checkpoint or consistent process, iv) consistent methodology for planning and executing, v) stakeholders’ involvement in requirement elicitation and change, vi) utilization maximum effort and vii) effectively use of tools and technology [11]. A quality model or framework helps to ensure quality in all aspects with factors: completeness, accuracy, efficiency, security and reliability, sustainability, usability-accessibility, portability, maintainability, etc.

C. Quality Models in Project Management

Factor Criteria Model (FCM) is considered the first quality model in software engineering [11] that is developed in 1977 by Air Navy [12]. FCM consists of 11 quality factors that are mapped into three major phases (operation, revision, transition) of the software development and each factor is mapped with multiple criteria of 23. FCM is also known as McCall’s software quality model. After one year (1978), Boehm’s model [13] is developed with a hierarchy structure with 7 top-level quality factors and 15 bottom-level quality factors, one higher factor is linked to 2 or more lower-level factors. It’s clustering consist of three major areas: portability, maintainability, and utility. International Organization for Standardization (ISO) update its generalization model ISO 9000 by ISO 9126 in 1991 [15], but full adaptation was completed in 1992 for software quality measurement [14]. In 1995, Dromey’s quality model is proposed that distinguished a software to multiple product-properties and recommend for adding quality attributes to each product-property list. It is standing on three principles i) setting high-level quality attributes ii) identification product properties that affect quality, iii) linking on product properties and quality attributes [16]. In 2001, ISO restructured the quality view with updated version ISO 9126 - 1:2001 [17]. ISO sets guidelines for measuring software characteristics and international standard measurement of software quality into four sub-domains: “ISO/IEC 9126-1 (ISO/IEC, 2001a)”, “ISO/IEC 9126-2 (ISO/IEC, 2003a)”, “ISO/IEC 9126-3 (ISO/IEC, 2003b)”, and “ISO/IEC 9126-4 (ISO/IEC, 2001b)” respectively for "define and update the model", "define attributes of external measures", "define attributes of internal measures", and "define

the quality on uses". It refined the six main quality measures to 3-5 sub-lists without overlapping. FURPS is an acronym that stands for functionality, usability, reliability, performance, and supportability and each of these has a set of quality attributes presented by a quality model by Robert Gready [18]. Mobile devices were not like today in the developing periods of these models and sustainability was not a serious concern of this domain; moreover, these models show important on the product not process (see details in Appendix-1).

IBM rational software extended this by the name FUPRS+ (2000) with integrating requirements on design, implementation, and interfaces [19]. Software Assurance Technology Centre (SATC), developed a model SATC's quality model in 1996 to support NASA that assisting manager for cost-minimizing and identifying testing quality within four goals: requirements quality, product quality, implementation Effectively, testing effectively [20]. Dromey's model is extended with a hierarchical structure for explicitly specify object-oriented design is known as Bansiya's QMOOD Model and it focuses on the identification of qualified design components, patterns, characteristics, and matrices [21]. Kazman et al integrate the quality factors into the software life style architecture [22] and our study motivated that logically quality factors should work together. Capability Maturity Model Integration (CMMI) gives priority to the organization level too besides project-oriented to ensure the quality of a system. It consists of five maturity levels for integrating and reviewing quality aspects of a system. It is also aligned to ISO 9001 standards; moreover, it promotes the Software Engineering Institute (SEI) of the USA. While ISO 9001 performs quality actions in software development and maintaining stages, CMMI's framework focuses on the continuous improvement of a software process with explicit information [23]. ISO is the world standard organization and CMMI is developed by SEI at Carnegie Mellon University in Pittsburgh and the main difference is "ISO is an audit standard" and "CMMI is a process model" [24]. ISO, CMMI, and IEEE are pioneer organizations in software quality standardization besides national standardization institutions of technological advancement countries. ISO certifies the software firms according to ISO9001 standards, and IEEE has a computer society for certification, ISO and API consulting for software developing companies certification, and CMMI has certification for every CMMI maturity level. This research encourages enhancing business and projects' ability and quality by following standard models, methods, and guidelines without recommending to get the certification.

D. Methodologies for Software Project Management

A methodology defines and mentions the work process and management flow every single part of the process. Sometimes, it assumes that more specification kills project execution time and increases project time but imagine "If any error appears after few days that need to recover from the foundation part"; what dangerous the situation is? And the consequence is losing the trust of customers. A short-cut is an instant success but not good for long-term goals. Quality assurance ensures a perfect balance among technology, process, and people to produce a quality product or service, and methodology creates a combination of these three ingredients. The wrong choice of

process methodology is a major risk that can appear during software development [25]. Method, process, model, and framework are upgrading continuously. Scrum is a popular methodology in agile families but the scrum team often overlooks the quality assurance activities due to the tight schedule or early delivery thought agile is not anti-methodology or against the quality practice [26]. The author [26] also demands quality assurance to work with the scrum team to clarify the goal, responsibilities, way handling issues, setting up monitoring and controlling strategies, and finally stay on track to achieve the goals. So, it is an additional activity that going to resolving by the proposed framework.

PRINCE2 (Project in control environment) is a project management methodology developed by the UK government and widely recognized in government and private sectors. It sets roles for the manager, customer, and supplier that uses the "PRINCE board" to accept inputs from users, suppliers, and experts. Series of sequential activities are recorded in PRINCE2 quality practice register: quality identifier, product identifier, product titles, method, roles and responsibilities, planned date of quality check, planned date of sign-off of quality check, the actual date of quality check, the actual date of sign-off of quality check, result, and quality records [27]. It is generic and adaptable to any project that is embraced with ISO 9000 standards, but it is not exact for software project management [28]. The waterfall model is the first process model and commonly used. It has sequential phases and the previous phase provides feedback to the subsequent phase. Quality assurance is predefined at the project initiation stage and practiced in all phases. QATestLab is applying the Black Box testing approach in the waterfall model for quality assurance [29]. It shows the value for quality plans, standards documentation, reviews, verification, demonstration, and quality assurance implementation to every phase. The spiral model has the most influence on risk management and it is right for handling projects with complex functional dependencies [30].

E. ITC Product and Sustainability

The information factories do not spew out carbon smoke but they are not bereft of environmental impact for explosion demand of energy. Already, a demand of 200 terawatt-hours (TWh) electricity (2018) is recorded for data centers every year that is more than the energy consumption of some countries like Iran [31]. But the demand will reach 1000 TWh according to the best case practice by 2030 [32]. The same research showed that 2500 TWh would be the least demand by 2030 for ICT productions, networking, data centers, and consumer devices. So, sustainability becomes a series matter of fact in Information and Communication Technology (ICT) similarly engineering and environment studies [33]. In addition, according to the "Ericsson Energy and Carbon Report 2015": mobile subscribers will be 9.5 billion where 55% of mobile data will come from video streaming data by the end of 2020 [34]. Less than 2% of greenhouse gas (GHG) emissions happen for ICT and it could be an enabler to reduce 98% of GHG that emission is not related to ICT [35]. It is also noticeable that 8% energy of the European union's (2015) is consumed by ICT services and subscribers' devices [36]. Definitely, the number of data centers, ICT infrastructures, networking area,

subscribers, the volume of information, have weight data (video), sensor data for Internet of things (IoT), and so on inventions will increase the demand for electricity in ICT sectors. A software engineer can carry out sustainability in a system by choosing comparatively less power consuming technology, applying the effective algorithm and data structure, integrating power-saving features, and reducing unnecessary workflow. On the other side, the new invention of technology “blockchain” consumes 12.76 Twh per year, becomes a new concern of sustainability [37]. However, ecological practice in electronic wastage management by proper destroying, reusing, and recycling is another dimension of sustainability.

F. Sustainability and Quality in Software Product

A software project has three major dimensions for improving quality with sustainability practice (Fig. 1).

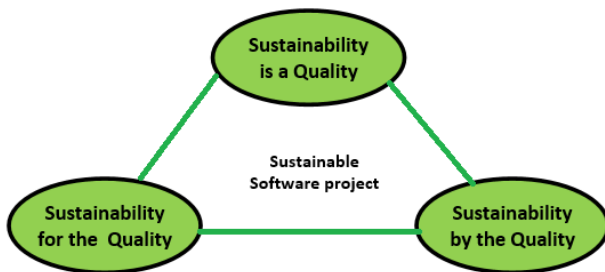


Fig. 1. Sustainability-Quality Relationship.

1) *Sustainability is a quality (sustainability practice in the product)*: According to David Bicknell, attention needs all phases from requirement analysis to deployment and operation phase of a system developing process cycle to create an energy-efficient application [38]. An application could be a sustainable system when it is featured with the auto switch to power saving mode when not need to display anything; auto shutdown actions for dormant applications; turn off wifi, Bluetooth, and global positioning features in a stable place; compression data transmission format; reduce screen brightness and screen timeout; sleep mode activation; reduce data transition steps; increasing resource sharing features; reduce log files; reducing unnecessary historical data; optimal power-consuming programming language, and omitting frivolous features, etc. it becomes more important for mobile systems such as robots, networking nodes, sensors, laptop, and mobiles. Technique and technology both are equally important to integrate green features. The project team should select the comparatively low power consuming data structure (green data structure) [39] and optimal power-consuming algorithm such as Low power Konsumption Algorithm (LOKA) is implemented to end-device of the ZigBee protocol [40].

2) *Sustainability by the quality (sustainability practice in the project)*: Most of the researchers noticed three dimensions for sustainability practice called the “triple bottom line approach” [41]. It is not common that most of them considered three factors of green project management, environmental sustainability practice, and financial sustainability practice. Project strategies and perspective could make suitable for

sustainability practice and it differs from project to project [42]. Some projects focus on social factors [44], while someone considered for financial [43], and others practice by environmental sustainability [43]; so sustainability practice is influenced by the subject of projects. In a software project, sustainability practice could be done by minimizing wastage of time, cost, and effort; maximize re-usability of existing assets: hardware, software, design, and code; and efficient resource allocation. Sustainability practice is not an individual action for a certain task or process of a project so linked with all phases, tasks, and stakeholders and it could be implemented by various methods [45]. This is a strategic level decision of an organization that is executing the project and projects’ sustainability practice should be aligned to the organization. A manager could achieve the target by controlling a project but sustainability is more versatile than just control [46]. Sustainability not only an act but also practice in life and all stakeholders’ participation is important [47].

3) *Sustainability for quality (sustainability by the product)*: An application in a wellbeing tool for humanity and a software project should consider the security concerns inside the system. A product is developed today but it could be backdated after two years for technological advancement that is not financially sustainable. If there is any lacking of protection the system could be abused by culprits that might make a problem for someone. So good software could contribute to sustainable social development by saving expenditure and time, ensuring privacy and security, and making social awareness. An application has an effect on the users’ mental and physical health, an embedded system has radiation effects, and kids’ game might be a cause for addiction (isolated from social gathering [48]). A smart device could broadcast radiation of non-ionizing radio-frequency that is hazardous for health [49].

III. S-Q FRAMEWORK

1) *Stakeholders*: This model distributes the stakeholders in two major groups called internal (hired employee) who mainly responsible to execute the project and external stakeholders (not working in the developing company) who will support (client, user, sponsor, government agencies) to execute properly. Two types of internal stakeholders are proposed in the framework due to enjoy outsourcing facility or economical employees from the corner of the world (virtual). The employees who are working in the office are remarked as physical mode internal stakeholders. The external stakeholders are not specified due to the enjoy flexibility of mode according to the applications and customers.

2) *Quality factors (QFs)*: There are plenty of quality factors and these are not specified in the model to keep adjustment facility according to the requirements and nature of an application. But the Appendix-1 shows the collection QFs from well know quality models with sustainability, re-usability. While a mobile application is highly biased by power-saving features, a computer user ignores the same features on the desktop. This model proposed to develop a guideline to

integrate the elements from the universal set of quality factors (QFs) (universal set is the superset of all models and sustainability factors). All quality factors are not equally essential to implement for all software except accuracy, security, and efficiency. The importance of a factor depends on the nature of applications such as sustainability is one of the most important for mobile applications. Usability and accessibility are essential for online web applications for mass people.

3) *Green environment*: Sustainability practice is not a single task or unique action in an organization or even in an application. It is part and parcel of personal and corporate practice in a business organization or industry. A software engineer could practice in office management and add to soft products. The green background of the framework (Fig. 3) shows a sign of sustainability and it bears the importance of practice in a project management workplace. An organization should encourage to practice sustainable technology (hardware and application) and a carbon-reducing office environment. Re-usability of historical soft documents (design, code), using tools (information system), and implement sustainability features in the software projects. Moreover, the sustainability motto of the organization can motivate external stakeholders to accept green soft products.

4) *Phase replace by task*: The project is divided into a series of tasks (task1, task2,, task_n) and these are closely adjacent because there is not significant transition between two tasks. It also stands for the flexibility of tasks so that the manager can adjust among tasks, teams, and resources of adjacencies. The size of the task is increasing means it is carrying the historical information and highly dependent on the earlier actions, and if anything is needed it can go back to get or update. It also gives the flexibility in the task for a minor period (margin) of adjustment. Tasks are not mentioned (like traditional phases) for keeping logical adjustment facility according to the application, resource allocation, and risk management. Though task_{n+1} is bigger than task_n in the diagram, the workload of task_{n+1} could be less than task_n because it carries the weight and dependency of previous stages.

5) *Sub-team*: A team could be developed from internal and external stakeholders who can work in the physical and virtual environment. Team with the required resource is tight to each other carries information that there is no transition gap between two adjacent teams. A team can get feedback from its neighbor and a team resource could be a common element of adjacent two teams. This scope has enhanced interaction and communication between two groups and mutual understanding will improve.

6) *Management*: A two-layer management body exists in the SQ-Framework to separate strategic activities and regular activities. A project manager is a key person to execute a project who has to report to the executive level. He/she has to work along to the mission and vision of the organization and business goals of the company too. Managers cooperate and execute organization level strategy of standardization to a project that helps to effectively handle his/her regular work such as manage, organize, coordinate, and keep control of it. A manager works in shell management who directly deals with internal and external stakeholders.

7) *Quality control spin*: The executive body is responsible to check assigned tasks of each team and the manager is the vital person of a project. Instance decisions, communication with stakeholders, risk management, change control, task review and approval, and quality assurance are the main activities. Moreover, it supports the core management for standard documentation development, practicing, assessing, and update regularly. Policy procedure and guidelines development for building a standardization record-keeping system and practice depends on core management. Other stakeholders will provide required input and feedback with accurate information. Information system (tool) and control language (e.g. descriptive logic) are proposed for record management and standardization respectively. Predicate logic, descriptive logic, or control language can reduce the ambiguity of an information system and it is familiar to the technical person of a software project.

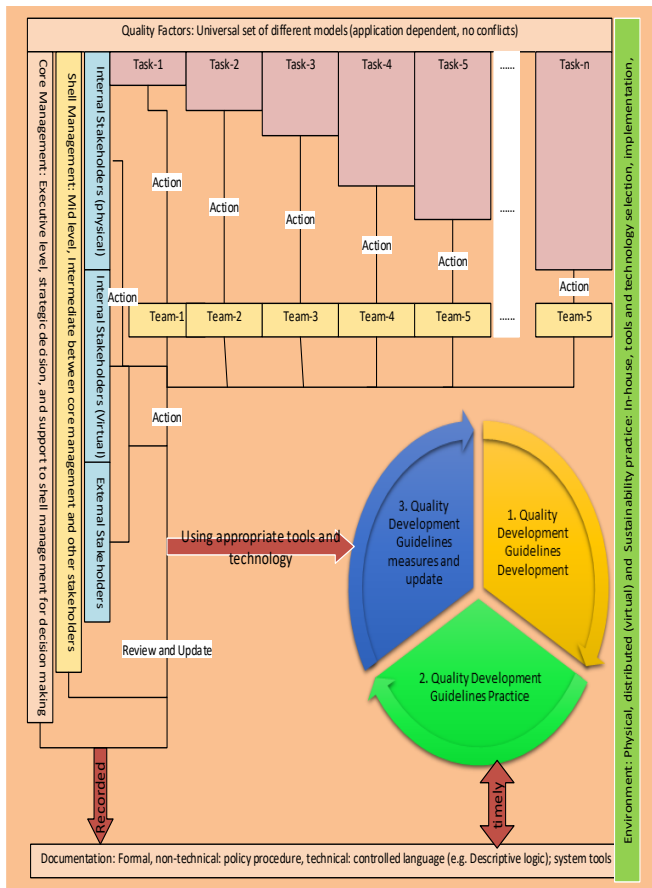


Fig. 2. S-Q Framework.

IV. FRAMEWORK EXECUTION MODEL

A 4-phase execution process model (Fig. 3) shows the way of SQ-Framework implementation by four sequential steps (left to right). These phases are connected with a logical sequence of actions but within a phase, the project team can execute parallel operations. The quality of a product is measuring in a particular stage but quality should be ensured in every stage. So quality practice will start from the organizations' strategy and the outcome will 100% fruitful only when the product is utilizing properly.

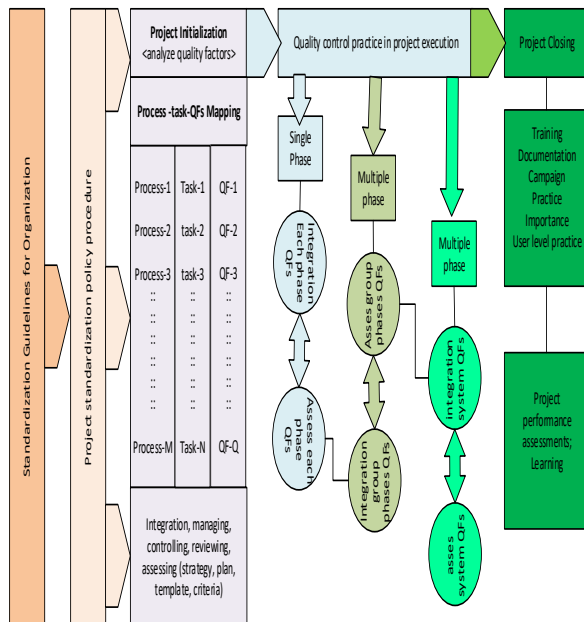


Fig. 3. Framework Execution Process.

Phase-1 (Definition): The executive body of an organization develops the strategy for the organization and is executed by subordinates. A project manager launches and executes the project according to the strategy of the organization. The manager can willingly accept and practice the sustainability and or quality guidelines of standardization but it must be practiced when the organization has influence. This phase asks to adapt and practice the standardization guidelines but not recommend for certification because any can practice without certification.

Phase-2 (initiation): This is the project initiation stage that analyzes the project scope, feasibility study, planning (hardware-software resources, time, cost, human resources, communication, acceptance, risk. etc.) for a traditional practice.

But, this phase is expected to do more with the strategy of integrating quality factors (QFs) and sustainability factors; specification of the way of implementation, measuring and assessing. Moreover, it suggests making the relationship with the process, task, and stakeholders of the project.

Phase-3 (execution): This is the quality implementation action into the product to each task and it would be finalized after a successful quality test. A group of qualified independent tasks could be integrated for the system and it has complete a successful integration test. This process will continue until qualified for the system test.

Phase-4 (closing): Phase-1 to 3 already enjoyed sustainability practice to the organization, for the project and integrated into the system; now this is the crucial phase for involving customers into the system and getting long-term advantages. It is also time to assess the performance of the system, learning from the project, and archiving the information for enjoying re-usability.

V. FRAMEWORK EXECUTION GUIDELINES

A. Karlskrona Manifesto

The Karlskrona Manifesto for Sustainability Design (KMSD) was the outcome of the Third International Workshop on Requirements Engineering for Sustainable Systems (RE4SuSy) at RE'14 in Karlskrona, Sweden [50]. A group of researchers from various disciplines brings the manifesto to leading an interdisciplinary sustainability research platform. The summary of Karlskrona Manifesto [51] is defined in Table I according to the nine principles classification of Oyedeji et al [52]. This specification will guide to create a relationship among stakeholders, task, process, and Karlskrona principle(s).

B. Guidelines of utilization SQ-Framework according to Karlskrona Principles (KP) and SQ-Framework Execution Model

SQ-Framework does not mention the phases of software project handling but any project has a set of logical sequences of steps with zero or more iteration that depends on the importance of phases. Documentation, management, quality control, communications are common activities and very much important for project management. Sustainability practice becomes one of them. But for better reflection, a set of the task is mentioned in Table II and KP of Table I is mentioned where it has more influence on each phase of the SQ framework execution model (Fig. 3). In Table II, QFs stands for quality factors and it carries a set of elements that are required according to application form the universal set of QFs (Appendix-1).

TABLE I. SUMMARY OF KARLSKRONA PRINCIPLES (KP)

Karlskrona Principle (KP)	Description
Sustainability meets the needs of the future without compromising current demands (KP1).	KP1 is a brief definition of sustainability that could be achieved by introducing Innovation science and technology, management process, and the lifestyle of human beings. It assumes that against the trade-off mindset of the greedy approach but better in long-term business aspects. Aim of the sustainability practice is not only getting advantages now but also the future.
Sustainability for long-term and continuous practices (KP2).	Advantages of sustainability can enjoy after long-term practice and it could be measured for a long period with multiple scales, indicators, and aspects. This practice should be continuous and mutual participation.
Sustainability is a systematic approach (KP3).	Sustainability is a systemic set of actions and processes that can execute collectively. Its' need a common background and common platform to design and implement. Its property makes a relationship as an organization to society, society to the nation, nation to global.
Sustainability has multiple dimensions (KP4).	Sustainability practice could not be possible by a single property or dimension because naturally, it has a relation with finance, environment, energy, and social values. According to a circumstance we have to analyze sustainability outcomes from these aspects.
Sustainability with multiple disciplines (KP5).	Sustainability practice includes people from multiple societies and backgrounds so working in this area is becoming challenging for human interaction. It also addresses the challenges of multiple disciplines and perspectives.
Sustainability is independent of the purpose of the system (KP6).	This principle represents the importance of sustainability and the emphasis on the value of the practice. Even initially is not yet mentioned for a particular task or sustainability is not primary focus but it has to be considered.
Sustainability is a wider context (KP7).	Sustainability is a part and parcel of a system and its surrounding environment too. It encourages to enhance the scope of dimensions and area of practicing or integrating elements that could be part.
Sustainability is a precondition to system design (KP8).	This is an amendatory condition that shows the significance study of sustainability in system design and development. Sustainability is an essential enabler for system design from different perspective levels and abstractions that could increase participation.
Sustainability action in multiple levels (KP9)	Look alternative better choice in every level of action and each process of a system to enjoy the most leverage on a system. It encourages comparison analysis on multiple options to accept a better.

TABLE II. MODEL UTILIZATION GUIDELINES ACCORDING TO “KARLSKRONA PRINCIPLES (KP)” AND “SQ- FRAMEWORK EXECUTION MODEL”

Project tasks <variable numbers according to Fig.3. The manager has to choose based on the nature and functions of the software project>	Phases of SQ- Framework execution model from Fig. 3			
	Phase-1 <identification, analysis, define and select for the project based on the organization's guidelines and individual attempt>	Phase-2 <map task, process, Sustainability, and quality factors (QFs) and select a better option from alternatives>	Phase-3 <implement according to the phase-1 and phase-2 but do not miss the scope of enhancement for quality>	Phase-4 <practice, learn, update for long term-goal: ensure participation, continuation, practice>
Requirement elicitation	KP1 & QFs <discover, classification, specification, prioritization>	KP2, KP3, KP6, KP9 & QFs <fixing for each requirement, what and how to implement>	KP2 & KP6 <implementation QFs based on Phase-1 & 2 for all functional & non functional requirements >	KP2 & KP9 <Verification, correction, learning for next time >
Feasibility study	KP1 & QFs < Analyze and setting for social, economic, technological, and select KP9>	KP2-5 & DFs <Fixing and keeping alternatives for each factor by KP6 & KP7 >	KP4-6 <implementation QFs based on Phase-1 & 2 for all aspects >	KP2-5 <Verification, correction, integration, preparation for next project>
Planning	KP3 & QFs < Allocation for time, risk, cost, resource, etc.>	KP3, KP8, KP9 & QFs <allocate for each task and process to maximum utilization>	KP3-5, & KP9 <execute QFs based on Phase-1 & 2 for all task according to plan >	KP3 <Verification, correction, realization and tools selection >
Design	KP8 & QFs < Identify by KP3-KP7 for the whole system too>	KP4, KP5, KP6, KP9 <set relationship of the task, process, and QFs>	KP6 & KP8 <implementation QFs based on Phase-1 & 2 for all functional & non functional according to design >	KP8 <Verification, correction, upgrading KP1 focusing on product and process>
Development & unit testing	KP7 & QFs <tools, process, template, time, person, method, specification >	KP3, KP6, & QFs <method and process of unit test and purposes fixing>	KP7 <implementation QFs based on Phase-1 & 2 for all task and verify>	KP7 <Verification, correction, upgrading phase-1 >
System Testing	KP7, KP8 & QFs < template, time, person, method, specification >	KP4, KP5, KP9 & QFs <method and process of unit test and purposes fixing>	KP4, KP5, KP9 <integrate QFs based on Phase-1 & 2 for all requirements and verify according>	KP7 <Verification, Completeness, correction, upgrading phase-1 and KP1 >
Deployment & Maintenance	KP4, KP5, and QFs <method, training, participation>	KP4, KP5, KP9 & QFs <regulation, procedure, training, participation>	KP4, KP5, KP9 <confirm utilization QFs based on Phase-1 & 2 for all by user >	KP1-9 < Realization stakeholders involvement and upgrading KP-1 >
Documentation	KP1-3, and QFs <specify technology, methods, process>	KP3-5, KP9 & QFs <fixing relationship with tool, format, and structure>	KP3 <ensure QFs based on Phase-1 & 2 for documentation standard>	KP3 <Alalysis and upgrade record-keeping, accept tools and techniques>
Management	KP3, KP5, & QFs <identification and specification task, person, process, procurement>	KP1-9 & QFs <fixing policy procedure to manage task, person, process, procurement >	KP3 <track, monitor, control for QFs based on Phase-1 & 2 for declaration>	KP1-9 <analyzised method, process, task and procedure, realize and upgrade>

C. SQ-Framework Implementation Algorithm Model

START

Phase-1 (Definition)

- 1.1 Study the system requirements (functional (user & expert), non-functional (expert), sustainability (expert))
- 1.2 Set the guidelines: Policy-procedure according to organizations' objectives and strategy
- 1.3 Apply KP1: Make clear with explicit specification by definition
- 1.4 Analyze and finalize system requirements (functional, non-functional, sustainability)

For the product, process, and human being:

Apply KP2 to select technology, KP3 for planning, KP4 for the feasibility study, KP5 for managing (task, process, human), KP6 for smart light, desk, data analysis, etc., KP7 for an individual (energy-saving, efficient working process), KP8 for organization policy and design aspect, and KP9 for the selection of better process/tool/technology.

End of 1.4

End phase-1

Phase-2 (Initiation)

- 1.1 Develop relationship for a task, process, and quality factors(QFs)
 - 2.1.1 Apply KP6 for each task
 - 2.1.2 Apply KP6 for each process
 - 2.1.3 Apply kp6 for each QFs (resolve conflict e.g. faster service & outstanding visualization)

Apply KP7 for generating relationship of 1.1, 1.2, & 1.3

- 1.2 Allocate resource for 1 according to phase-1 (time, effort, money, asset, stakeholders)
- 1.3 Develop quality measuring, assessing criteria, method of assessing, and template for each task, process

End of phase-2

Phase-3 (Execution)

- 3.1 Apply execution for
 - 3.1.1 Individual task and the respective process by following appropriate $KP_i (i=1,2,3,\dots,9)$
 - 3.1.2 Perform assessment (QFs) for each
 - 3.1.3 Apply for a group (3.1.1-2) of related tasks and the respective process by following appropriate $KP_i (i=1,2,3,\dots,9)$
- 3.2 Apply for n number of 3.1 until the finish

End of phase-3

Phase-4 (Closing)

- 4.1 Justification by $KP_i (i=1,2,3,\dots,9)$

For the process, task, product (Aspect: social, economic, environment, QFs)

Analyze KP1 based on assessment and If a required update
- 4.2 Justification for maintainability, reusability, and training by documentation standard by $KP_i (i=1,2,3,\dots,9)$ if the required update in KP1
- 4.3 Justification for user participation for product $KP_i (i=1,2,3,\dots,9)$ and if required update KP1
- 4.4 Analyze the project and accept learning for next practice by $KP_i (i=1,2,3,\dots,9)$

End of Phase-4

END

VI. FRAMEWORK ANALYSIS

The SQ-Framework (Fig. 2) is a comprehensive model for sustainable and qualified software development. The framework is not rigid for a specific software domain and it consists of brief guidelines with "Karlskrona manifesto" (Table II). The application of SQ-Framework is presented with a model "SQ- Framework execution model" (Fig. 3) and an algorithm (5.3). Key sustainability attributes are specified by the analysis of the impact of social, economic, and environment to integrate into the framework. There are plenty of software quality models as well as software development life cycles but there is a gap though both are working for software quality control. The framework demolishes the gap and brings in a single platform called "SQ-Framework" that performs project management, quality control, and sustainability practice because individually these are not effective. This model considered cross-platform compatibility in sustainability design for all stakeholders and showed importance on "quality product", "quality process", and "quality management". It aims to make sure the integration of quality factors but not what is the meaning of each factor but all models ignored the importance of sustainability according to the demand of current ages (Appendix-1). But, a reality the quality and sustainability are very closed to each other when a manager works for sustainability automatic quality will improve and vice versa. But this research considered sustainability practice by the system, in the project, and in the product and modeled by Fig. 1.

Features of current applications or systems are not the same as five years back and not will be the same for five years later. Technological advancement changes the working environment and business demands. Common uses of robotics, industrial revolution 4.0 and automation, smart environment, and web video data are increasing rapidly and applications are also biased by those. So, sustainability becomes one of the significant QFs and it also accelerates to integrate other QFs. Industry people could use this framework for systematic software engineering practice. They can modify this according to their demand. It will motivate the executive body for adopting sustainability strategy in the business organization, inspire the client to practice and implement software for sustainable practice, a software development team can develop policy procedures and guidelines to integrate quality and sustainability factors into a system, and manager can improve management process.

VII. CONCLUSION AND FUTURE WORK

Sustainability and quality highlight each other and jointly both shine the product and developing company. Moreover, sustainability practice in the software industries is new and till now there is a scope for clarification for current sustainability perception. SQ-Framework showed where the scope for implementing sustainability and quality is, how to practice, and who will practice in the software industries. The concrete framework is helpful for engineering practice in soft products and measuring the quality. The milestone of this research is the development of SQ-Framework that reduces the gap of quality models and methodologies. The contribution consists of "identification importance of the sustainability"; "gap of

practicing quality and sustainability”; “the distance between quality model and system development methodologies”; “developing linking between quality and sustainability”; and describes with execution model, Karlskrona manifesto, and algorithm.

SQ-model expects explicit standard documentation, formal practice according to the recommendation of standardization organizations or owns developing standard quality strategy. That is not possible for Ad hoc basis or freelancing development as well as difficult for comparatively small companies who are developing small projects. But if any organization wants to be a standard company then SQ-Framework will guide, though the first time it has to do hard work. Moreover, it would be easier for an experienced manager to execute, modify, and practice.

This paper would guide the researchers to develop new methodologies according to the upcoming technology trend with the dimensionality reduction of complexity, standard documentation ontology, domain-oriented sustainability-quality models: For robotics, embedded system, communication systems, smart infrastructure with (internet of things) IoT, etc.

REFERENCES

- [1] Luigi Buglione, Some thoughts on quality models: evolution and perspectives, Acta IMEKO, vol. 4, no. 3, article 12, September 2015, identifier: IMEKO-ACTA-04 (2015)-03-1.
- [2] Demarco T., Controlling Software Projects: Management, Measurement & Estimation, Yourdon Press, 1982.
- [3] Sommerville, I. (2011). "Chapter 24: Quality Management". Software Engineering (9th ed.). Addison-Wesley. pp. 651–680. ISBN 9780137035151.
- [4] Mitchell Grant, (2020).Sustainability - <https://www.investopedia.com/terms/s/sustainability.asp> accessed: 6 May 2020
- [5] Gimenez, C., Sierra, V., & Rodon, J. (2012). Sustainable operations: Their impact on the triple bottom line. International Journal of Production Economics, 140(1), 149-159.
- [6] Calero, C. Piattini, M.IntroductiontoGreeninsoftwareengineering. GreenSoftw. Eng. 2015,3–27.
- [7] Pulse of the Profession 2020 (2020).Ahead of the Curve: Forging a Future Focused Culture. Project management institute. available. <https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2020> visites:10th May 2020.
- [8] Pulse of the Profession 2019 (2019). The future of work, leading the way with PMTQ. Project management institute (PMI). available:<https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2019>, visited: 10 May 2020.
- [9] Clancy, T. The Standish Group Report, Retrieved 10 MAY 2020, from <http://www.projectsmart.co.uk/reports.html>, Chaos report, 2018.
- [10] Pulse of the Profession 2017 (2017). 9th global project management survey, <https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2017>. visited: 10th May 2020.
- [11] Discenza, R. & Forman, J. B. (2007). Seven causes of project failure: how to recognize them and how to initiate project recovery. Paper presented at PMI® Global Congress 2007—North America, Atlanta, GA. Newtown Square, PA: Project Management Institute.
- [12] McCall J.A., Richards P.K. & Walters G.F., Factors in Software Quality, Voll. I, II, III: Final Tech. Report, RADC-TR-77-369, Rome Air Development Center, Air Force System Command, Griffiss Air Force Base, NY, 1977.
- [13] Boehm B.W., Brown J.R., Kaspar H., Lipow H., MacLeod G.J. & Merritt M., Characteristics of Software Quality, Elsevier North-Holland, 1978.
- [14] Suman et al, A Comparative Study of Software Quality Models. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5634-5638.
- [15] ISO/IEC, IS 9126:1991 - Information Technology - Software product evaluation – Quality characteristics and guidelines for their use.
- [16] Ranbireswar S. Jamwal, Deepshikha Jamwal & Devanand Padha, “Comparative Analysis of Different Software Quality Models”, 3rd National Conference, February 26 – 27, 2009.
- [17] ISO/IEC, IS 9126-1:2001 - Software engineering -- Product quality -- Part 1.
- [18] Grady, Robert B. 1992. Practical Software Metrics for Project Management and Process Improvement. Englewood Cliffs (NJ), USA: Prentice-Hall, p 282.
- [19] Kruchten, P. 2000. The Rational Unified Process: An Introduction, 2nd Ed. Boston (MA), USA: Addison-Wesley Professional. 320 p.
- [20] Hyatt, L., and L. Rosenberg, 1996 "A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality," NASA SATC, 1996.
- [21] Bansiya, Jagdish, Davis, Carl G., 2002. A hierarchical model for object-oriented design quality assessment. IEEE Trans. Software Eng. 28 (1), 4–17.
- [22] Kazman, R.; Nord, R. L.; & Klein, M. A Life-Cycle View of Architecture Analysis and Design Methods (CMU/SEI-2003-TN-026, ADA421679). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. <http://www.sei.cmu.edu/publications/documents/03.reports/03tn026.html>.
- [23] Mark C. Paulk (1994). A Comparison of ISO 9001 and the Capability Maturity Model for Software. Software Capability Maturity Model Project. Technical Report CMU/SEI-94-TR-12 ESC-TR-94-12 https://resources.sei.cmu.edu/asset_files/TechnicalReport/1994_005_001_435267.pdf.
- [24] Iso, Cmmi and Agile : A Comparison <https://www.vizteams.com/blog/iso-cmmi-and-agile-comparison/>.
- [25] Ajayi, W., Adekunle, Awodele, Akinsanya, Eze, & Seun, E. (2018). Software Development Top Models, Risks Control and Effect on Product Quality. Global journal of computer science and technology.
- [26] Vasile Selegean, (2015). Quality Assurance in Agile-SCRUM environment. today software magazine, issue-33. 2015. <https://www.todaysoftmag.com/article/1355/quality-assurance-in-agile-scrum-environment> visited: 12 May 2020.
- [27] Dave Litten. (2017). PRINCE2 quality control. Prince2 Primer. <https://www.prince2primer.com/prince2-2017-quality-control/> visited: 12 May 2020.
- [28] Simon Buehring, (2020). PRINCE2 benefits, advantages and disadvantages. <https://www.knowledgetrain.co.uk/project-management/prince2/prince2-benefits>. visited: 12 May 2020.
- [29] QATestLab, (2018).QA Activities in Waterfall Process.Knowledge Center. <https://qatestlab.com/resources/knowledge-center/waterfall-process/>. visited: 12 May 2020.
- [30] M. A. Akbar et al., "Improving the quality of software development process by introducing a new methodology—AZ-model", IEEE Access, vol. 6, pp. 4811-4823, Dec. 2017.
- [31] Nicola Jones, (2018). The information factories -Data centers are chewing up vast amounts of energy — so researchers are trying to make them more efficient. Nature 561, 163–166; 2018 available:<https://media.nature.com/original/magazine-assets/d41586-018-06610-y/d41586-018-06610-y.pdf> visited: 13 May 2020.
- [32] Andrae, A.S.G.; Edler, T. On Global Electricity Usage of Communication Technology: Trends to 2030. Challenges 2015, 6, 117-157.
- [33] Nicola Jones, (2018). How to stop data centres from gobbling up the world’s electricity. nature. 13th september 2018. <https://www.nature.com/articles/d41586-018-06610-y>. visited: May 15 2020.
- [34] Ericsson Energy and Carbon Report 2015. <https://www.ericsson.com/assets/local/about-ericsson/sustainability-and-corporate-responsibility/documents/ericsson-energy-and-carbon-report.pdf> (Accessed 13 May 2020).

[35] Malmodin, J., Bergmark, P. and Lundén, D. (2013), The future carbon footprint of the ICT and E&M sectors. Proceedings of the First International Conference on Information and Communication.

[36] Calero, C.; Piattini, M. Introduction to Green in software engineering. *Green Softw. Eng.* 2015, 3–27.

[37] Elbahrawy, A.; Alessandretti, L.; Kandler, A.; Pastor-satorras, R. Evolutionary dynamics of the cryptocurrency market. *R. Soc. Open Sci.* 2017, 1–16.

[38] David Bicknell. 2012. 8 ways to make your software applications more energy efficient. <https://www.computerweekly.com/blog/Green-Tech/8-ways-to-make-your-software-applications-more-energy-efficient> (visited 15 May 2020).

[39] Zahereel Ishwar Abdul Khalib, R. Badlishah Ahmad, Ong Bi Lynn, "Energy Efficient Scheduling Algorithm for Soft Real Time System with High Deadline Meeting Rate on Overload", *Applied Mechanics and Materials*, vol. 699, pp. 840, 2014.

[40] Vaquerizo-Hdez, D., Muñoz, P., R-Moreno, M. D., & F Barrero, D. (2017). A Low Power Consumption Algorithm for Efficient Energy Consumption in ZigBee Motes. *Sensors (Basel, Switzerland)*, 17(10), 2179. <https://doi.org/10.3390/s17102179>.

[41] V.K. Chawla, A.K. Chanda, S. Angra and G.R.Chawla. (2018). "The sustainable project management: A review and future possibilities" *Journal of Project Management* 3 (2018) 157–170.

[42] Martens, M. L., & Carvalho, M. M. (2017). Key factors of sustainability in project management context: A survey exploring the project managers' perspective. *International Journal of Project Management*, 35(6), 1084-1102.

[43] Khodadadzadeh, T. (2016). Green building project management: obstacles and solutions for sustainable development. *Journal of Project Management*, 1(1), 21-26.

[44] Silvius, A. J., & Schipper, R. P. (2014). Sustainability in project management: A literature review and impact analysis. *Social Business*, 4(1), 63-96.

[45] Aarseth, W., Ahola, T., Aaltonen, K., Økland, A., & Andersen, B. (2017). Project sustainability strategies: A systematic literature review. *International Journal of Project Management*, 35(6), 1071-1083.

[46] Kivilä, J., Martinsuo, M., & Vuorinen, L. (2017). Sustainable project management through project control in infrastructure projects. *International Journal of Project Management*, 35(6), 1167-1183.

[47] International Organization for Standardization (2010). ISO 26000. Guidance on Social Responsibility, Geneva.

[48] Stavrou, P. (2018) Addiction to Video Games: A Case Study on the Effectiveness of Psychodynamic Psychotherapy on a Teenage Addict Struggling with Low Self-Esteem and Aggression Issues. *Psychology*, 9, 2436-2456. doi: 10.4236/psych.2018.910140.

[49] Austin, (2019). Smart devices may be hazardous to your health Contributor, THE TRUTH WILL OUT. CIO. JULY 2019, <https://www.cio.com/article/3411951/smart-devices-may-be-hazardous-to-your-health.html>. visited: 16 may 2020.

[50] Christoph, B. Sustainability and longevity: Two sides of the same quality? *CEUR Workshop Proc.* 2014, 1216, 1–6.

[51] Becker, C.; Chitchyan, R.; Duboc, L.; Easterbrook, S.; Penzenstadler, B.; Seyff, N.; Venters, C.C. Sustainability Design and Software: The Karlskrona Manifesto. In Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, Italy, 16–24 May 2015.

[52] Oyediji S, Seffah A, Penzenstadler B. A Catalogue Supporting Software Sustainability Design. *Sustainability*. 2018; 10(7):2296.

Appendix-1

This is no model that emphasis on sustainability, re-usability of project resources, and power-saving features	
Quality models	Quality attributes
McCall's quality mode	Product version: Maintainability (Simplicity, Conciseness, Self-descriptiveness, Modularity); Flexibility (Self-descriptiveness, Expandability, Generality); Testability (Simplicity, {instrumentation, Self-descriptiveness, Modularity} Product operations: Correctness (Traceability, Completeness, Consistency); Efficiency (Execution, efficiency, Storage, efficiency); Reliability (Consistency, Accuracy, Error, tolerance); Integrity {Access control, Access audit); Usability (Operability, Training, Communicativeness}
Boehm's quality	Utility: Reliability (Self Containedness, Accuracy, Completeness, Robustness/Integrity, Consistency); Efficiency (Accountability, Device Efficiency, Accessibility); Human Engineering (Robustness/Integrity, Accessibility, Communicativeness) Portability: Device independence; Self Containedness Maintainability: Testability{ Accountability, Communicativeness, Self Descriptiveness, Structuredness}; Understandability (Consistency, Structuredness, Conciseness, Legibility); Modifiability (Structuredness, Augment ability}
Dromey's Quality model	Correctness: Functionality, Reliability Internal, Maintainability, Efficiency, Reliability Contextual: Maintainability, Reusability, Portability, Reliability Descriptive: Maintainability, Efficiency, Reliability, Usability
FURPS Quality Model	Functionality: feature sets, capabilities, security Usability. human factors, aesthetics, consistency, Correctness, Reliability: severity of failure, recoverability, predictability, accuracy Performance: speed, efficiency, availability, accuracy, throughput, response time, recovery time, and resource usage Supportability: testability, extensibility, adaptability, maintainability, compatibility, serviceability, installability, and localizability.
ISO 9126 quality model	Functionality: Suitability, Accuracy, Interoperability, Security, Functionality Compliance Reliability, Maturity, Fault Tolerance, Recoverability, Reliability Compliance Usability: Understandability, Learnability, Operability, Attractiveness, Usability Compliance Efficiency: Time Behavior Resource Utilization, Efficiency Compliance Maintainability: Analyzability, Changeability, Stability, Testability, Maintainability Compliance Portability: Adaptability, Installability, Co-existence, Replaceability, Portability Compliance