How to cite this paper:

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

AN EMPIRICAL STUDY ON PREDICTORS OF GREEN SUSTAINABLE SOFTWARE PRACTICES IN MALAYSIAN ELECTRONIC INDUSTRIES

Bokolo Anthony Jnr., Mazlina Abdul Majid & Awanis Romli
Faculty of Computer Systems and Software Engineering
Universiti Malaysia Pahang, Malaysia

bkanjr@gmail.com; mazlina@ump.edu.my;
awanis@ ump.edu.my

ABSTRACT

Currently, sustainability is a pertinent issue that should be considered in the software development process; hence it is imperative to recognize how environmental-friendly practices can be applied in the electronic industries that develop and deploy software products. However, sustainability is not fully considered when electronic industries implement modern software systems. Additionally, software developers in electronic industries believe that software is environmental friendly mainly because it is virtual. Conversely, the life cycle process and approaches applied to implement, deploy and maintain software do possess social and environmental impacts that are usually not accounted for by electronic industries. Therefore this study identified the predictors that determine sustainable software practice applications in electronics industries by presenting a model to facilitate sustainable software products development. The identified predictors influence sustainable software practices applications which correlate to environmental, technical, economic, social and individual dimensions of sustainability in electronics industries. Based on the identified predicators,

Received: 27 June 2017 **Accepted**: 21 December 2017

this research developed a set of indicators for survey questions and collected data from 133 respondents from Information Technology (IT), software, environmental and electronic-based industries. The survey data aimed to verify each of the identified predictors that influence sustainable software practice applications. Descriptive and inferential statistical results from the survey data show that each of the predictors is significant and do influence sustainable software development. The finding from this study provides insights to electronic industries in implementing sustainable software practice applications.

Keywords: Green software development, sustainable software development dimensions, software practice application, software process life cycle, predictors.

INTRODUCTION

Computer systems mainly consist of hardware which includes physical devices such as memory, CPU, input, output circuits, etc. and installed software programs that instruct the hardware to execute specified operations. Software does not utilize power by itself, but energy is consumed by the hardware when powering the motherboard circuit. Software does control the deployment flow in hardware and intrinsically impacts the energy proficiency of the hardware. With the emerging issue of global warming and increasing energy-related costs, reducing energy associated with computer utilization has become an important issue (Moshnyaga, 2013). But as the years go by, sustainable software research is gaining momentum based on the critical need for Green development as well as the effect of Information Technology (IT) on our society (Dustdar et al., 2013; Anthony & Majid, 2016b). Although IT plays an essential role in resolving sustainability issues, IT can be utilized in electronic industries to facilitate Green software engineering by deploying ecologically-friendly operations that consume less resources such as using e-mail instead of postal mail or deploying virtual meetings and teleconferences instead of travelling to attend software development team meetings (Jnr et al., 2017). IT possesses the capability to synthesise knowledge towards enhancing resource-intensive processes; for example, informatics for water consumption and smart energy grids for power utilization. Alternatively the impacts generated by the development of IT-related products are rarely accounted for across industries; for instance, it is projected that one computer becomes outdated for every new computer put in the shop.

At the moment, old computer hardware are discarded even when they are still usable due to newer software versions that mostly render the hardware unusable. But if software developers acknowledge and take this fact into deliberation, novel software products and services can be developed to run on older hardware platforms (Albertao et al., 2010). But since software-executed applications and systems are more prevalent in industrial activities and society at large, the environmental impact of software-deployed products has indeed become a global issue. IT infrastructures utilized in electronic industries contribute to about 2% of global carbon dioxide (CO2) emission, an amount equal to the aviation-based industry. IT can be deployed in electronic industries to achieve software system efficiency in terms of energy consumption, deployment of architectural optimization and practice of effective software engineering management practices (Lami & Buglione, 2012; Anthony & Majid, 2016a).

Over the years, due to the utilization of computing applications, software is integrated with the life of the society and subsequently software development is becoming increasingly related to sustainability (Amri & Saoud, 2014). Green sustainable software is an extension of Green IT which over the years has concentrated on hardware optimization towards waste minimization, energy reduction and CO2 emission reduction. Green IT practice aims to decrease energy-related costs incurred in industries and organizations, but software runs on hardware and the software facilitates the functionality of hardware, and without the application layer, IT-integrated hardware systems cannot be deployed to work. Consequently academicians have been paying much consideration to the effect of software within Green IT. This propagated the birth of Green sustainable software which is an application or program that produces as little waste as possible throughout software development and usage (Erdelyi, 2013).

Green sustainable software produces less IT-related waste than the old traditional software, but developing Green software entails certain operations to be considered during the software development process. Although software development methodologies transform continuously, the key operations such as requirements specification, system analysis and design, implementation, testing and deployment, maintenance and modification, etc. remain unchanged. In electronic industries, approaches such as agile methodology are deployed based on different traditional activities that consume more energy, generate e-waste, utilize natural resources, emit CO2 and at times cause pollution of the environment. Due to the effects, Green software engineering was suggested to develop software that facilitates environmental consciousness and also generates less waste throughout the development. However over the years, sustainability issues in the software development process have been addressed by a set of defining sustainability specific procedures as suggested by ISO/IEC IS 12207 and ISO/IEC IS 15504 (ISO/IEC, 2008; ISO/IEC, 2011) standards which provided sets of guidelines to facilitate sustainability management, sustainability engineering and sustainability qualification in the software process. Nevertheless, researchers such as Lami & Buglione (2012) mentioned that ISO/IEC IS 12207 and ISO/IEC IS 15504 only provided mere definitions of the Green sustainable processes and as such were not sufficient to provide software practitioners with an operative means to address sustainability of the software processes since electronic industries utilize software systems by means of software programs or applications; for instance, software is utilized to enhance the design, analysis, production, maintenance and disposal of software products and the services being developed. It is consequently obvious that software is infused in the software development process (Penzenstadler, 2014). Although academicians in the Information Systems (IS) domain have recently been trying to find competent solutions for environmental issues tagged as “Green IT” and “Green IS”, it is not yet confirmed whether natural resource and energy savings by software will surpass its resource utilization. Over the years there has been a range of scientific contributions towards Green IT and Green IS; while most of the work has mostly focused on environmental sustainability in correlation to computer hardware, only a few studies have concentrated to address issues related to Green sustainable software practice in achieving sustainable development in the electronic industries domain towards CO2 reduction, cost decrease, waste minimization, decreased natural resources utilization and lesser energy utilization.

Therefore this research aimed to identify the predictors that influence sustainable software practice application mainly in the electronics industries. Furthermore, this study also considered not only the environmental dimensions as explored by previous researchers but also considered the social, economic, people and technical dimensions of sustainability in relation to sustainable software practice applications. Findings from this study provided empirical evidence on the predictors of sustainable software practice applications in the electronic industries. Furthermore, this study indicated the significance of the predictors that influence green sustainable software practice applications. The remainder of this article is structured as follows. The next section presents the related works; as the third section presents the methods. Then the results of the survey are provided. Next, discussions from the survey are outlined, after which the practical and research implications are revealed. The article concludes with the conclusion, limitation and future work section.

RELATED WORKS

This section reviews existing scientific studies that have been carried out regarding sustainable software development. Since this study presents a research model presenting the predictors that influence sustainable software practice, only past studies that presented models or frameworks for sustainable software practice were reviewed. Among the studies, Kern et al. (2013) investigated the energy saving ability of software programs by exploring Green software engineering. The authors described a reference model for sustainable and Green software to evaluate energy proficiency of software in addition to its engineering, and lastly they provided some definitions related to sustainable Green software development. The reference model predictors comprised of the software product life cycle, sustainability criteria, model procedure and lastly recommendations and tools. The limitation of this study was the authors only assessed the energy efficacy of software consumption.

Scanniello et al. (2013) developed an approach aimed at facilitating migration strategy to provide a current software system which was ecologically sustainable throughout the development lifecycle. Particularly, the authors presented a strategy and procedure for migrating software system based on a graphics-processing unit architecture. The developed approach predictors comprised of reverse engineering, reengineering and integration and testing, although their approach was limited to lowering energy consumption resulting in a Greener and more eco-sustainable system. Kocak (2013) researched on Green software development for ecological sustainability and offered a framework based on the Analytical Network Process (ANP) to facilitate decision-making. Their approach involved two main levels; the first level aimed to develop Green sustainable software, whereas the second level outlined the criteria to be considered for developing Green sustainable software. The researchers adopted the quantitative research methodology integrated with a case study approach. The predictors or criteria in their studies included functionality, reliability, usability, efficiency, energy consumption, CO2 emission, Green energy usage and return of Green investment. However, their study only addressed power consumption analysis on database-deployed software. Steigerwald & Agrawal (2011) described the features of Green software design methodologies and considerations to enhance software energy efficiency. The authors believed that software plays an imperative role in decreasing power utilized on mobile platforms. Hence their research aimed to improve the power usage issue in mobile systems that used software. The researchers explored computational efficiency, data efficiency, context awareness of humans and idle efficiency as predictors in their research. The limitation of their study was that the researchers only improved software energy efficiency in mobile-based devices that utilized software for longer battery life in mobile devices.

Abdullah et al. (2014) proposed a model that integrated the web-based knowledge management system to control and disseminate Green software development knowledge among software development team members. The researchers aimed to fill the gap in knowledge and address how to infuse the knowledge management approach to administer Green software development knowledge. The model comprised of global environmental alertness, competitive awareness and industry initiatives, web-based knowledge management system, Green software development process and software development community. However, the model could only be applied on a web-based knowledge management system to share and manage Green knowledge of software development. Amri & Saoud (2014) developed a generic sustainable software star model that created the basis for achieving a comprehensive view of sustainable software. The model also aimed to provide a roadmap for sustainability which still remained an intangible concept for software developers. The model predictors encompassed technical, environmental, economic, individual and social dimensions of sustainability; however, the model could not be adopted to manage software sustainability characteristics during software life cycle.

Shenoy & Eeratta (2011) proposed a Green software development model that provided a method for sustainable software development. The model addressed the alterations in the traditional software development life cycle and recommended suitable steps and activities that could lead to reduced carbon emissions, less power consumption and limited paper use, thereby supporting software enterprises to achieve Greener software development. The proposed Green software development model predictors comprised of requirements, design, implementation, testing, deployment, maintenance, retirement alongside supporting process. Although the model was concerned with environmental issues, economic and societal dimensions of sustainability were not fully addressed in the model.

Dustdaretal.(2013)examinedGreensoftwareservicesinrelationtostakeholders' requirements and presented a business model to resolve Green software from a business standpoint. The model was based on three main predictors of Green software services stakeholders, stakeholders' requirements and business models. The limitation of the model was that the authors addressed Green software issues from the business perception trying to ascertain stakeholders' benefits only; the environmental dimension of sustainability was not fully explored, only the people and economic dimensions were inculcated in their study. Dick et al. (2010) presented some findings that formed the foundation for sustainable software attainment and designed a software process life cycle model for Green sustainable software engineering. The process and life cycle model helped to achieve energy savings through Information Communication Technology (ICT) by overbalancing the energy consumption of ICT. The model predictors comprised of guidelines and checklists, process and life cycle model (which included development, acquisition and distribution, deployment, usage and maintenance, and deactivation and disposal) and lastly the developers, administrators and users. Although the researchers considered the 3 dimensions of sustainability (society, economy and environment) they did not provide solutions for resolving energy efficiency related issues. Johann et al. (2011) explored software usage, software development process and proposed a life cycle model to support Green software development and sustainable software systems. Furthermore, the researchers presented tangible comprehension to support software professionals involved in the software development life cycle process. The proposed life cycle model predictors included metrics for tools, models and software systems for carrying out measurement as well as comparability in relation to sustainability. The authors failed to present how they could resolve societal, economic and environmental issues in the software development process. These pillars of sustainability were isolated in their study.

Thiry et al. (2014) designed a GreenRM reference model for sustainable software development to assist in decreasing the effect caused by Greenhouse gas emissions, energy utilization and e-waste generation. The GreenRM model predictors were based on the ISO/IEC 14001 environmental management requirements. Hence, the model infused the Green IT concept into software development alongside ISO/IEC 14001 environmental management requirements to the organizational process. Thus, the GreenRM reference model could be utilized as a guide for environmental endorsement as well as for the implementation of Green IT practices. The authors evaluated the GreenRM reference model in three Brazilian-based software organizations to test the financial and technical feasibility of the model. The model was grounded only on the ISO/IEC 14001 environmental requirements. Due to this, the author did not consider the economic and societal effects of the software development process.

Mahmoud and Ahmad (2013) proposed a model to facilitate the Green and sustainable software engineering process and product. The model comprised of a two-stage Green software model that addressed the sustainable life cycle of software tools and software products supporting environmentally sustainable software practice. The model predictors covered the first and second levels. The first level suggested the sustainable software engineering process that comprised of a hybrid iterative, agile development and sequential processes aimed at producing environmentally sustainable software. The second level described how the software itself can be utilized as a tool to support Green practice by monitoring natural resources utilization in an energy proficient manner. The model was criticized for addressing only the software product life cycle towards promoting environmentally sustainable software. The technical and individual concerns were slightly addressed.

The finding from this section presents a review of studies similar to this research. Although all the 12 reviewed studies aimed to achieve Green sustainable software development, none of the studies attempted to identify the predictors that may influence Green sustainable software development in the electronic industries domain. The studies were mostly concerned with the life cycle process and dimensions of sustainability in the software development domain. The predictors that influenced the Green sustainable software process in relation to the life cycle process and dimensions of sustainability were not fully explored by the researchers. Therefore, there is a need for a study to identify the predictors that influence Green sustainable software development in relation to the attainment of the dimensions of sustainability in electronic industries.

Industries Involved in Sustainable Software Practice Application

This section presents a comparison of the types of industries involved in the sustainable software practices application.

Information Technology-based Industries

IT-based industries such as IBM deployed a Tele-work software application in 2005. The system achieved a cost-saving of fuel, thereby decreasing CO2 emissions. IBM's Tele-work software system reduced pollution and traffic congestion. IBM also applied a cloud computing technology called virtualization in achieving energy savings. Virtualization deploys fewer servers to control several services in an industry. Hence, in virtualization, a limited number of servers are used which means enhanced manageability, lower cooling costs, less headcount and reduced CO2 emission (Harmon & Auseklis, 2009).

Software-based Industries

Software-based industries such as Sun Microsystems reduce their transport cost and CO2 emission generated when industrial staffs come to work by applying the open-work software system, which provides a solution suite of policy products and support software tools that allow Sun employees to work efficiently in the office or at a remote location (Boudreau et al., 2008). Other software-based industries such as Google, Microsoft and Yahoo have re-located a few of their industrial data centers to the Pacific Northwest, close to cheap hydroelectric energy sources. Google also deployed solar power facilities in few of their offices (Harmon & Auseklis, 2009).

Manufacturing and Engineering-based Industries

Manufacturing and engineering industries such as Intel which develop processors, chips, motherboards chipsets, integrated circuits and network interface controllers currently provide resources for applying the sustainable information system. The industry applies Green software practices by deploying energy competent data centers, virtualization, server operation analyzer, energy effective services through Green procuring, Green manufacturing and solar panel installations (Grant & Marshburn, 2014).

Supply Chain Management-based Industries

Supply chain management industries such as Wal-Mart presently apply information software systems to manage their supply chain transportation and distribution operations. Wal-Mart presently uses ecological friendly pack among their wholesalers. In the context of integrating sustainable software, the industry is usually imperiled to pressures from its supply chain contacts that have currently or previously applied Green practices. Wal-Mart uses sustainable software to monitor and measure enterprise costs, carbon emissions and e-waste generated in each phase of the service product packaging (Boudreau et al., 2008).

Automotive-based Industries

Over the years, automotive-based industries such as Ford have been utilizing information systems software to administer their vehicle sales and services to their customers and suppliers. Ford also applies the ISO 14001 Environmental Management System (EMS) aimed at caring for the environment when the industry disposes of by-products generated from motor vehicle manufacturing. Additionally, Toyota Corporation deployed the built-in information system software to manage hybrid engines and features to facilitate ecological-friendly driving, with diverse driving positions to reduce cost-expenditure through fuel effectiveness (Simmonds & Bhattacherjee, 2014). Volvo also applies a viable information software system aimed at lessening energy utilization in their logistics division. The software management system collects real-time data used to enhance and optimize truck logistics, thereby decreasing CO2 emission from the industry’s vehicles during transportation operations.

Government-based Industries

Improving social-environmental performance and natural resource consumption is an essential part of sustainability. Thus, government-based industries are currently aligning environmental, economic and societal goals concurrently rather than addressing them separately. But at the moment they are faced with challenges. Among these challenges, there is the reduction of IT-associated energy usage, waste and emissions. Opportunities exist because government-based industries are applying information systems software to lessen material utilization, reduce CO2 emissions, and minimize energy consumption. Therefore, government-based industries are beginning to consider the environment by adopting Green software initiatives. The application of such Green software initiatives is mostly induced by economic forces that result in decreasing energy costs, and adhering to environmental protection regulations enacted by non-governments or inter-governmental associations (Harmon & Auseklis, 2009).

Institutions of Higher Learning-based Industries

Institutions of higher learning such as university campuses are similar to small cities in terms of urban characteristics and population size and several diverse activities take place across the campuses, which possess direct or indirect impacts on the natural environment. University campuses involve several operations and activities each with implications to the eco-system that directly or indirectly impacts the environment but over the years these campus operations have been generally overlooked in terms of environmental and social responsibility. As such, only economic-related operations have been fully addressed; hence, to address the environmental and social dimensions university campus activities and operations apply software systems that provide information for monitoring significant environmental and social impacts (Nifa et al., 2015).

Electronic-based Industries

Electronic-based industries are mainly computer software and hardware-based enterprises. These industries such as Dell, Apple, Toshiba, etc. apply Green practices in their enterprise towards promoting.

Zero Carbon strategy aimed at decreasing hardware infrastructure energy consumption of the industries’ products, thereby lessening CO2 emission. These industries also allow their end users to recycle their earlier equipment if they procure new equipment. Hence, electronic-based industries contribute to recycling by providing suitable procedures facilitated by software systems to track and monitor the movement of hardware products to be recycled (Anthony, 2016).

The review of existing industries that have applied sustainable software to facilitate their industrial process are discussed in this section. The findings show that each of the industries aimed to address environmental-related issues as seen in the electronic industries. However, none of the reviewed industries has fully and concurrently addressed all dimensions of sustainability (economic, social and environmental, technical and individual) when applying sustainable software systems or applications. Hence, there is a need for an approach to support a sustainable software practice application that considers the economic, social and environmental, technical and individual dimensions of sustainability in the electronic industries.

METHODOLOGY

This study aims to identify the predictors that influence sustainable software development in electronic industries. Figure 1 is followed to accomplish the aims of this study and also verify the identified predictors that influence sustainable software development practice in electronic industries.

Figure 1 shows the methods carried out in this study. As seen in Figure 1 the methodology comprises of four main steps. Step 1 is mainly the literature review that discusses the existing models or frameworks developed to support the sustainable software practice application, the dimensions to be considered for sustainable software practice application in the electronic industries, the predictors that influence sustainable software practice application in the electronic industries and lastly the life cycle process to be applied for achieving sustainable software practice in the electronic industries. Step 2 is the generation of indicators to measure and verify each of the predictors that influence sustainable software practice application, reliability and validity test for each indicator and lastly choosing purposive sampling to collect data from 133 respondents, where the sample population is from IT, software, environmental and electronic-based industries. Next is step 3 which is data collection which uses online survey questionnaires, and lastly step 4 is the analysis of the collected data and the presentation of the results using descriptive statistics (via frequency, mean and standard deviation, maximum, minimum and median value) and inferential statistics (using regression analysis).

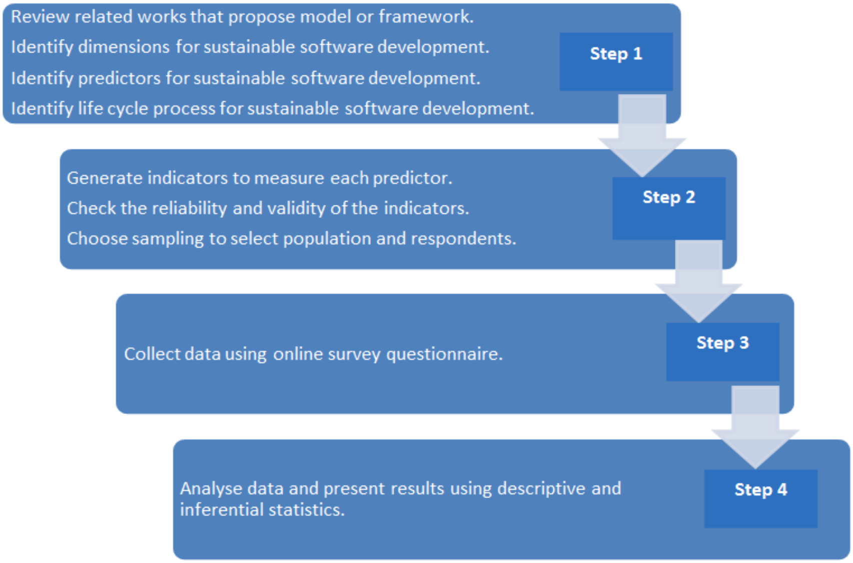
Review related works that propose model or framework.
Identify dimensions for sustainable software development.
Identify predictors for sustainable software development.
Identify life cycle process for sustainable software development.
Step 1
Generate indicators to measure each predictor.
Check the reliability and validity of the indicators.
Choose sampling to select population and respondents.
Step 2
Collect data using online survey questionnaire.
Step 3
Analyse data and present results using descriptive and inferential statistics.
Step 4

Figure 1. Methodology for this research paper.

Dimensions for Sustainable Software Development

This section presents the dimensions to be considered for sustainable software practice application in electronic industries. The dimensions to be attained for sustainable software development in electronic industries are shown in Figure 2.

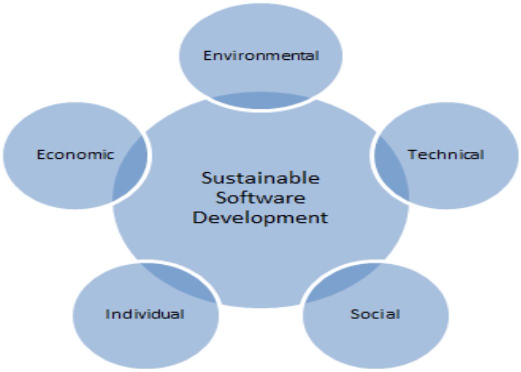

Figure 2. Sustainable software development dimensions.

Figure 2 shows the dimensions to be considered for sustainable software development in electronic industries. Each of the dimensions are discussed below.

Environmental

The environmental dimension emphasises how software can be developed, utilized, maintained and disposed-off with negligible impact on the natural environment. In the electronic industries, environmental dimension is assessed based on two main aspects which involve resources consumption and energy consumption. The consumed resources include software products, software applications, hardware and materials such as printing paper, storage media, etc. The consumed energy can be managed by deploying energy efficient practices (Amri & Saoud, 2014). According to Penzenstadler (2014) environmental dimension is mostly concerned with waste management and natural resource usage which can be assessed using life cycle evaluation. Moreover the environmental dimension in electronic industries can also be explored based on the ecological impact assessment. Thus, the environmental dimension reflects the impacts of software system deployment on the atmosphere (Anthony & Majid, 2016a).

Technical

The technical dimension comprises software quality system requirements such as reliability, supportability, portability and maintainability which all results in the durability of software systems infrastructures in the electronic industries. The technical dimension also entails energy efficiency of hardware (Penzenstadler, 2014; Anthony & Majid, 2016a). Technical dimension also addresses how software is developed so that it can be easy to adapt to imminent change. Additionally, technical dimension also relates to long-time utilization of software systems. The technical dimension comprises the functional and the operational aspects that influence software system survivability. Functional software involves alterations due to changes in requirement whereas technical is normally due to continuous technology changes (Amri & Saoud, 2014).

Social

The social dimension includes computer-sustained collaboration in the industry which involves communication among software developers, software service end-users, software decision-makers and software development team members, through software application for personal, organizational and industrial usage (Penzenstadler, 2014). The social dimension also focuses on how to develop software that can also improve social capital value. Therefore, this dimension emphasises on software communal added values. The social dimension in relation to software systems is divided into two main categories; the end-users and the technical community (Amri & Saoud, 2014). The end-users are the society that utilizes the software services whereas the technical communities are the community of software developers.

Individual

The individual dimension involves the welfare of software practitioners in relation to their health, safety, privacy and security as well as their personal well-being. The individual dimension aims to address the welfare of the software practitioners working in industries (Penzenstadler, 2014; Anthony & Majid, 2016a). The individual dimension also addresses how software can be developed and maintained in a manner that facilitates software developers to be contented with their profession for a long period of time in correlation to the software development process being applied in the industry. Furthermore, the individual dimension also addresses the comfort of software developers in relation to working conditions, number of working hours, salary payment, knowledge and skills upgrading of software developers (Amri & Saoud, 2014).

Economic

The economic dimension addresses financial constraints and monetary expenditure incurred by the industry in applying sustainable software development (Penzenstadler, 2014; Anthony & Majid, 2016a). The economic dimension also takes into consideration how software systems can be developed so that the stakeholders' investments are as safe as possible from economic-related risks. For any electronic industry to be economically sustainable, developed software services and systems should possess a reduced cost process, a long-term profit, and the operations should support the industrial capital in assisting software managers make decisions based on the assessed economic paybacks before executing any project (Amri & Saoud, 2014).

Predictors for Sustainable Software Practice Application

Recently a few researches have been published on developing and using sustainable software. Some studies focused on developing sustainable software, while others proposed software methods to support all software professionals in developing sustainable software systems and products (Mahmoud & Ahmad, 2013). Others paid attention to developing software tools that quantify the impact of software on the natural environment and energy efficiency (Erdelyi, 2013; Mahmoud & Ahmad, 2013). However, no studies have investigated sustainable software in the electronic industries. The existing work are mostly concerned with only software industries, hence there is a need to identify the predictors that influence sustainable software practice application in electronic industries. The predictors discussed below were selected for this study based on the fact that these predictors were suggested in previous studies related to Green sustainable IT and Green sustainable IS in IS and environmental related research. Hence, we were motivated to explore these predictors for Green sustainable software in the electronic industry domain.

Software Practitioners

This predictor comprises the software experts, professionals, developers and software team members that possess the skills and knowledge to develop sustainable software. This predictor comprises the staffs involved in industrial operation. In the industrial context "software practitioners" refers to the number of people in a particular electronic industry, hence industries with more practitioners are more likely to apply sustainable software development practices. Also software practitioners' attitudes towards the environment will affect the outcome of sustainable software development. The electronic industries should train their staffs on sustainable software development. Thus electronic industries should not only see software developer experts, professionals, software team members and software support staffs as a means to attaining profit, but need also emphasise on the welfare of the software practitioners (Mishra et al., 2014; Akman & Mishra, 2014; Deng & Ji, 2015; Lami & Buglione, 2012).

Software Governance

Software governance comprises the administrative rules and regulations that oversee the industry's daily operations. Software governance refers to policies that support industries in decision-making. These polices are guidelines that direct sustainable software development practices aimed at influencing sustainability attainment, hence software governance policies increase the industry's awareness on issues pertaining to sustainability governance at the management level and also provides an agenda for software practitioners in the industry to achieve sustainability. Software governance polices also ensure that the materials to be procured are ecologically friendly and will cause little or no harm to the natural environment. This predictor incorporates the commitment and support of the management towards the industry applying sustainable practices for sustainability attainment, where the management support is an important indicator for any industrial success (Ali et al., 2016). Thus, software governance policies comprise the agenda put toward by the management to support software developers apply ecological-friendly practices in the industry's daily activities (Penzenstadler, 2014; McGibbon & Van Belle, 2013; Deng & Ji, 2015; Uddin et al., 2015).

Technologies and Systems

Technologies and systems consist of both IT infrastructures such as servers, networks and software, and hardware utilized by software developers to deliver the intended objectives of the industry (Surendro et al., 2016). Hence the industry acquiring, deploying eco-friendly technologies and systems can facilitate the attainment of sustainability. These technologies may include server virtualization and server consolidation. Technologies and systems predictors also explore the technical perspective that influences the application of sustainable software development. These technologies and systems enable sustainable related practices in industries as they aim to decrease energy depletion of running facilities. They can be-utilized to reduce power consumed in the cooling of IT infrastructures by enhancing energy competence of IT infrastructure (Luan et al., 2015), thereby lessening Greenhouse gas emissions (Negulescu & Doval, 2014). Renewable power technologies generated from solar or wind can be used as a substitute to replace coal-fired energy stations that deliver electricity needs, since coal emits carbon emissions which add to global warming. Information systems can be deployed to digitize industrial documents and e-filing cabinet systems by automating industrial daily activities, thus reducing office space, minimizing costs and energy required for the book-keeping process (Surendro et al., 2016). Technologies such as Radio Frequency Identification (RFID) which uses the electromagnetic field to automatically identify, track the gathering and the handling of data could help generate sustainable practice which can be used to improve the industrial pollution-prevention policy agenda (Karanasios et al., 2010; McGibbon & Van Belle, 2013; Deng & Ji, 2015; Mishra et al., 2014; Akman & Mishra, 2014; Lami & Buglione, 2012).

Pressure

This predictor involves the rules and regulations initiated by governmental and non-governmental bodies to protect the natural environment, hence pressure is a predictor that influences the industries' decision to apply sustainable software development in attaining sustainability. These pressure results from rising energy costs of energy utilization in the industry, thus resulting in the need for the industry to lessen energy consumption. Furthermore, electronic industries are presently being pressured by regulators to practice ecological friendly software development. Other pressures such as social pressure also influence the industry's mission to apply sustainable software development practices. This is induced by the increased community demand for ecological-friendly services and the positive public understanding of sustainable software (Howard & Lubbe, 2012; Jenkin et al., 2011; Karanasios et al., 2010; Ainin et al., 2016; Krishnadas and Radhakrishna, 2014).

Software Strategy

This predictor comprises the activities and procedures carried out in electronic industries. The strategy is an important predictor that influences industrial growth and also promotes the industry's bids to practice sustainable software development in achieving environmental, social and economic advantages in the long term. The strategies infused may include supporting software developers reduce operational cost and minimizing carbon emissions, thereby changing the direction towards realizing the goal of sustainable development. Thus, electronic industries should possess strategies with goals aimed at attaining a neutral carbon operation. This predictor also involves the description of the industry's scope and operations carried out for sustainable software development. Software strategy therefore, aims to support the industry's reduced operating costs in development, hence strategy deployed in accomplishing the industry's objectives is significant in sustainable software development (Deng & Ji, 2015; McGibbon & Van Belle, 2013; Krishnadas and Radhakrishna, 2014; Savita et al., 2014; Mangla et al., 2015).

Knowledge Accessibility

One of the assets in the electronic industry is the knowledge held by software developers and practitioners involved in the development of software products. Furthermore, knowledge of environmental sustainability is becoming a valuable and intangible asset that can be used to facilitate Green competitive advantage in the software development process (Abdullah et al., 2015). Hence, one of the main assets in the electronic industry is the knowledge held by software developers and practitioners, where software development can be referred to as a knowledge-intensive practice and it is imperative to disseminate the knowledge efficiently so that electronic industries can decrease time and cost, thereby improving the quality of software products (Abdullah et al., 2015; Ali et al., 2016). Furthermore, the knowledge accessibility predictor signifies activities and practices that facilitate the process of creating, capturing, disseminating and sharing knowledge to provide experience that can be used to provide sustainable suggestions and improvement to novel software developers (Koçak et al., 2014).

Sustainable Software Practice Application

In software engineering sustainability is still an evolving field, where a sustainable software is a software whose direct and indirect effects on the environment, economy, human and society result from its usage, whose development and deployment are minimal, and/or also have a positive influence on sustainable development (Johann et al., 2011; Dustdar et al., 2013). According to Johann et al. (2011) sustainable software relates to software whose direct and indirect utilization of natural resources arise based on the deployment and consumption operations that are continuously monitored, measured, assessed and improved in the development life cycle to cyclically minimize the software process for direct and indirect usage of energy and natural resources. Lami & Buglione (2012) added that Green software engineering should focus on the software development life cycle that adopts techniques and principles aimed at improving sustainability attainment.

Sustainable Software Development Lifecycle

This section presents the life cycle process to be applied for achieving sustainable software practice in the electronic industries. Lami & Buglione (2012) suggested that in order to address sustainability issues in the software development process, there is a need to apply a minimum set of sustainability-specific life cycle processes. These life cycle processes should be defined based on eco-friendly activities to be practised in order to integrate and introduce Greenness culture in the electronic industries. The life cycle process includes development, distribution, acquisition, deployment, usage and maintenance, deactivation and lastly disposal as shown in Figure 3.

Figure 3 shows the sustainable software practice application life cycle process to be implemented in the electronic industries. The first life cycle process is the development phase, which is the main focus of this study "Sustainable Software Practice Application". In this process several well-structured tools, techniques and methods are applied throughout the software development process, hence participating software practitioners are able to evaluate sustainability impacts that arise from the overall software development life cycle. Furthermore, this phase allows software developers to take action in enhancing software products in order to improve environmental impacts, thereby designing a more sustainable software product (Dick and Stefan, 2010; Dick et al., 2010). Hence in the development phase, ecological impacts that result directly from the software development operations as well as the effects of industrial software design operations are considered. These range from energy that is needed to power software developers' workstations, for

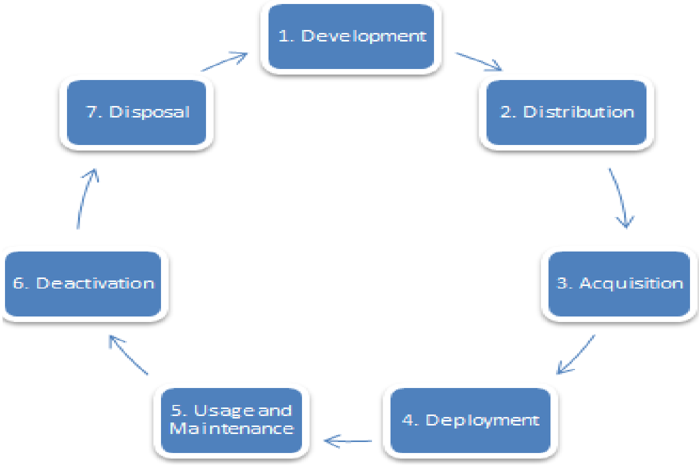

Figure 3. Sustainable software practice application life cycle process.

power needed to operate IT infrastructures such as networking-related devices such as enterprise servers, for energy needed for departmental office lighting and energy needed for air conditioning, ventilating, heating industrial offices and cooling data centers. Additional impacts include energy consumed daily when software practitioners' commute to work, for transportation during team meetings with software development team members or end-users (customers) (Johann et al., 2011).

Next is the distribution phase which is pertinent for both custom and standard software. This phase aims to resolve sustainability impacts that result from the manufacture of data medium, such as the packaging or the transfer of software packages (Dick and Stefan, 2010; Dick et al., 2010). The distribution phase also considers effects on sustainable development that arise from the delivery of software products. This phase also comprises the ecological impacts of printed manuals which are paper derived from the exploitation of natural forest wood, selected means of conveyance, design and type of merchandising and transport wrapping (such as plastic, cardboard, wooden transport pallets and polyurethane foam), data medium (such as CD/DVD, Universal Serial Bus (USB) memory stick and download) in addition to the download size if the software is accessible as a download which also utilizes network bandwidth resulting in energy usage (Johann et al., 2011). Next is the acquisition phase where software practitioners' evaluate a few standard software products and select the standard that best fits the current development needs and procure hardware components that execute the software from Green software accredited retailers such as Dell, HP, etc. Throughout the software selection or acquisition phase, software practitioners' may also consider the functional, technical and licensing criteria in relation to the sustainability criteria (Dick and Stefan, 2010; Dick et al., 2010). Another phase is the deployment lifecycle process which considers features that are applicable for software administrators during the running, deployment and execution of the software systems (Dick and Stefan, 2010; Dick et al., 2010).

The usage and maintenance phase mostly address the indirect and direct sustainability impacts which develop from the use of software-related products (Dick and Stefan, 2010; Dick et al., 2010). The usage and maintenance process results from using, deploying and maintaining software products. In practice sustainable software developing maintaining does not relate to the traditional software maintenance which involves not only resolving and addressing bugs but also involves software administrators taking care of installed software for end-users. Thus, maintenance may involve the installation of software updates or patches, re-configuration of software systems and the proper training of novice software practitioners and staffs on appropriate software usage, etc. Such sustainable practice training can support industrial staffs to tum off office lighting and their computers when they leave their offices, thereby resulting in less energy depletion (Johann et al., 2011; Mahmoud & Ahmad, 2013).

The deactivation process addresses aspects which become significant if software systems are decommissioned out of service (Dick and Stefan, 2010; Dick et al., 2010); any software product decommissioned is often replaced with a new software system. Hence, it is essential to transform the existing data to the new software material format or to make it available for software practitioners. This might have an economic impact on the industry (Johann et al., 2011). The disposal process considers the impacts on sustainability in relation to the disposal of data package and medium (Dick and Stefan, 2010; Dick et al., 2010). The disposal phase also addresses the impacts on the natural environment that result from recycling and disposing the aforementioned user manuals, data mediums and packages (Johann et al., 2011). This phase is responsible to address the replacement of hardware that are outdated or obsolete due to technology change. Hence, the disposal phase covers software recycling in relation to the reuse of the software code for future software projects, thus reducing in-house software development costs. The hardware recycling involves the reuse and recycling of hardware equipment instead of disposing the facilities and materials that can be re-used repeatedly.

Research Model

Based on the finding from the dimensions to be considered for sustainable software practice application, predictors influence sustainable software practice application and the process life cycles to be applied for achieving sustainable software practice in electronic industries. The research model is developed as shown in Figure 4 which shows the developed research model for this research. The model presents the identified predictors (software practitioners, software governance, technologies and systems, pressure, software strategy and knowledge accessibility) which influence the dependent variable "Sustainable Software Practice Application" in the electronic industry resulting in the environmental, technical, social, individual and economic dimensions of sustainability.

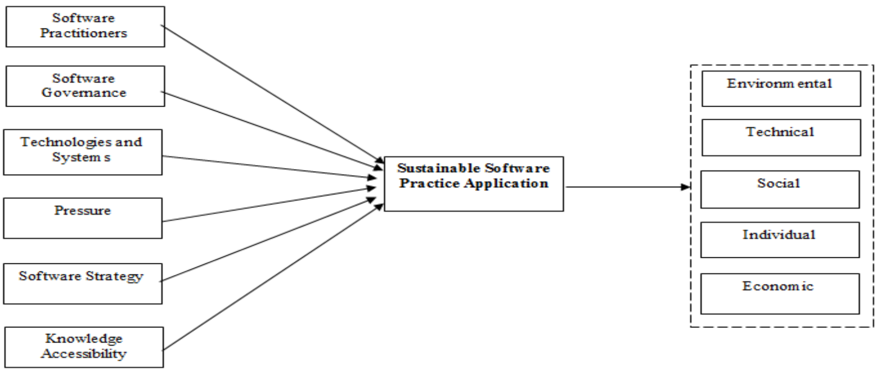

Figure 4. Research model.

Indicators Generation

Table 1

Operationalization of Predictors and Indicators

Predictors	Code	Indicators
Software practitioners	SP1	Positive attitude of software practitioners.
	SP2	Ethical consideration of software practitioners.
	SP3	Social-culture of software practitioners.
	SP4	General capabilities of software practitioners.
	SP5	Beliefs of software practitioners in relation to climate and environment.
	SP6	Knowledge of software practitioners in relation to climate and environment.
	SP7	Experience of software practitioners in the industry.
	SP8	Software practitioners' commitment.

(continued)

Predictors	Code	Indicators
Software governance	SG1	Formal industrial structures.
	SG2	Industrial management playing leading role.
	SG 3	Industrial management support.
	SG4	Industrial management investigation on ways to reduce software's power consumption.
	SG5	Industrial management advocates the use of equipment by potential software suppliers.
	SG6	Industrial management policy for the use of software to reduce overall wastes.
	SG7	Industrial management policy on staff's use of software in an energy-efficient manner.
	SG8	Allocated budgets and other resources by industrial management.
Technologies and systems	TS1	Transforming its industrial process to be paperless.
	TS2	Server/Storage virtualization and consolidation to reduce energy usage.
	TS3	Use of teleconferencing for industrial meetings.
	TS4	Use of video conferencing for daily operations.
	TS5	Use of telecommuting by software developers transporting around the organization.
	TS6	Use of on-line collaboration tools for industrial day-to-day software operations.
	TS7	Installation of software to reduce overall emissions and wastes.
	TS8	Installation of software to reduce overall use of hazardous and toxic materials.
Pressure	PS1	The pressure from government and non-governmental bodies.
	PS2	Management involvement influences sustainable software development.
	PS3	Provision of government incentives and other resources.
	PS4	The actions of other industrial competitors.
	PS5	Pressure from software clients, software consumers and software vendors.
	PS6	Encouragement from industrial associations.
	PS7	Future consequences of industrial actions
Software strategy	SS1	Tackling the carbon foot print of software-based systems.
	SS2	Own industrial strategy.
	SS3	Financial returns (cost saving) on investment.
	SS4	Plan initiatives on how to achieve environmental goals.
	SS5	Effective routines to facilitate the combination of newly acquired knowledge.
	SS6	Refine procedures to facilitate the combination of newly acquired knowledge.
	SS7	Develop business opportunities based on sustainability perspective.

(continued)

Predictors	Code	Indicators
Knowledge accessibility	KA1	Providing latest data relating to the environment and climate across the industry.
	KA2	Usage of data to communicate and have access to information unconstrained.
	KA3	Providing precise and unique data within the industry.
	KA4	Providing same and consistent data across the industry.

To measure the predictors that influence sustainable software development in the electronic industry, indicators are derived from existing literature to measure each predictor. Each of the predictors and associated indicators are shown in Table 1. Software practitioners and software governance are all measured with 8 different indicators with a 5-point Likert scale ranging from not important as "1" and very important as "5". Technologies and systems are measured with 8 different indicators with a 5-point Likert scale ranging from not relevant as "1" and very relevant as "5". Pressure is measured with 7 different indicators with a 5-point Likert scale ranging from not influential as "1" and very influential as "5". Software strategy is measured with 7 different indicators with a 5-point Likert scale ranging from not important as "1" and very important as "5". Lastly knowledge accessibility is measured with 4 indicators with a 5-point Likert scale ranging from not important as "1" and very important as "5".

Reliability and Validity of Indicators

Reliability measures the extent to which the questionnaire (instrument) gives the same result consistently. The value of alpha measures the internal consistency of a test and it is defined as a number ranging from 0-9 (Hair et al, 2010). Kumar (2005) provides the following rules of thumb: "> .9 – Excellent, > .8 – Good, > .7 – Acceptable, > .6 – Questionable, > .5 – Poor and < .5 – Unacceptable". Cronbach's alpha, is the most widely used objective measure of reliability and it was used to measure the reliability of the questionnaire adopted for this study. The closer Cronbach's alpha coefficient is to 1.0 the greater the reliability (internal consistency) of the items in the scale. The reliability test that was conducted on the data that was obtained from the SPSS version 22 is shown in Table 2.

Table 2 shows the Cronbach's alpha result of the questionnaire to be 0.986. This reveals that the instrument (questionnaire) used in this study has good reliability and is appropriate for the study.

Table 2

Item Reliability

Reliability Statistics		
Cronbach's alpha	Cronbach's alpha based on standardized items	No. of items
.986	.986	42

With regards to the validity of the instruments (questionnaire), validity refers to the degree to which a measurement (questionnaire survey) tool actually measures the construct that is used to measure (Hair et al., 2010). To check the validity of the indicators in the survey, face and content validity were carried out as suggested by Hair et al. (2010). Face validity is the easiest form of validation and indicates whether the survey indicators (items or questions) appear to be appropriate for the purpose of this study and the content area. Thus, face validity in this study evaluated the appearance of the survey questions in terms of feasibility, readability, consistency of style, formatting and clarity of the English language used, whereas content validity was concerned to what extent the indicators are relevant and represent the items to be measured as presented in Table 1. Furthermore, a few domain experts in Software Engineering were involved in assessing the content validity of each indicator to check if the indicators derived from the model predictors were easily understandable. Lastly content validity helped to confirm that the participants understood each of the questions in the survey.

Sampling Technique

The participants were chosen using the purposive sampling technique, where each participant was selected based on their current roles in their industry. To confirm that each participant was suitable to provide the data needed in verifying the identified predictors, each participant's profile and background were confirmed through their organizations' sustainability departmental website. Email request messages were sent to qualified participants to partake in the survey session at the convenience of the participants. The respondents were IT, software, environmental practitioners who had in-depth understanding of the issues surrounding the sustainability practice application. However it is to be noted that even though the sample was selected from industries in Malaysia, it did not represent all electronic industries in Malaysia.

Data Collection

Data was collected using an online survey questionnaire from 133 respondents from different industries based in Malaysia. The survey questionnaire was designed to verify the identified predictors that influence sustainable software development in electronic industries. The respondents were asked to select the importance of the different indicators used to measure each predictor with regards to sustainable software development in their respective industries. The survey questionnaire was reviewed and updated by experts (as previously stated) to further enhance the questions after which it was deployed online and the link to the survey was sent via e-mail alongside a cover letter to potential respondents that currently implemented Green sustainable practice in their industries. In the survey questionnaire the participants were given a short introduction which included the need for the research and a brief definition of a few key terminology. The respondents were also assured of their anonymity. The first part of the survey questionnaire contained the research overview. The second part carried the demographic characteristics of the respondents and their industries. The third part had questions that measured each of the predictors based on several questions. The six predictors and the related indicators are shown in Table 1. The participants were asked questions to measure the level of importance of each predictor's indicator. The Likert scale with five response categories (1-5) was used where "1" indicated not important and "5" represented very important. The higher the selected value, the more important the indicator is in relation to the measured predictor.

RESULTS

The collected data was analyzed using the descriptive statistic technique. The characteristics of the survey participants are shown in Figures 5 to 10.

Figure 5 shows that 58% of the respondents were male and the remaining 42% were female. With regards to the age of the respondents, 43% of the respondents were between 35-44 years old, 42% were 25-35 years old, 13% of the respondents were around 45-55 years old, 1% was less than 25 years old and another 1% was above 55 years old.

Considering the educational qualification of our respondents, Figure 6 shows that 32% were Bachelor's degree holders, 23% was Diploma holders, 30% possessed Master's degrees, 13 % were PhD holders and lastly only 2% were high school certificate holders.

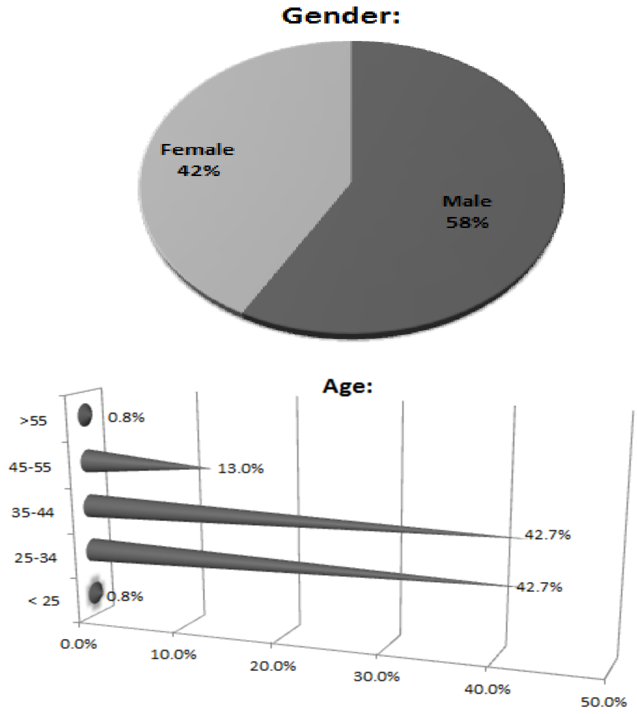

Figure 5. Demographic data for gender and age distribution.

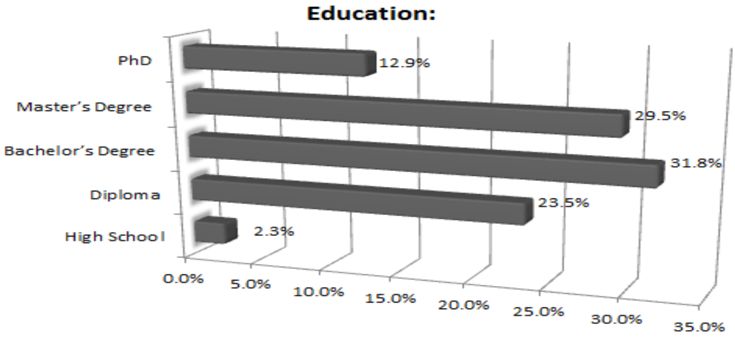

Figure 6. Demographic data for education and country distribution.

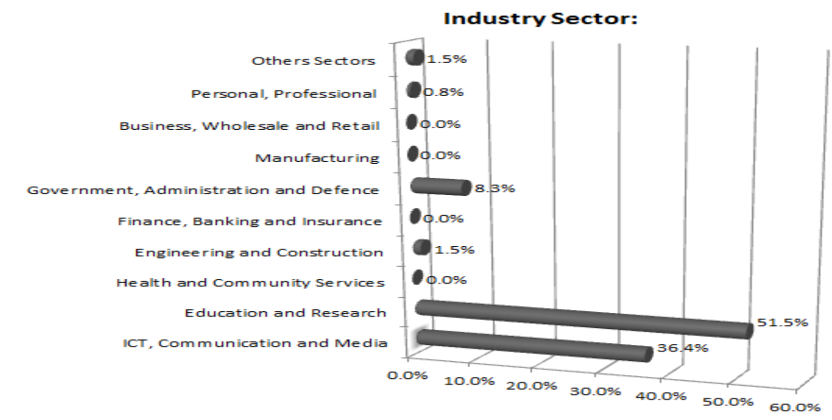

Figure 7. Demographic data for industry sector.

Figure 7 shows that 57% of the respondents (software practitioners, software professionals and software staffs) were from the education and research sector such as institutions that adopted sustainable practices, followed by respondents from the ICT, communication and media industries with 36 %. Next were the government, administration and defense-based industries with 8%. 2% of the respondents were from the engineering and construction-based industries and another 2% of the the respondents were from other services, and 1% of the respondents were from the personal and professional-based industry.

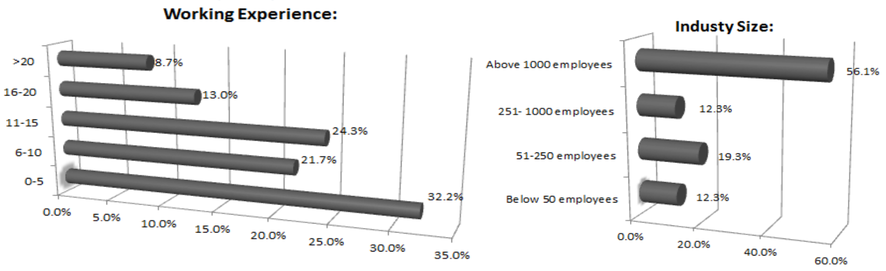

Figure 8. Demographic data distribution for working experience and industry size.

Figure 8 shows that 32% of the respondents possessed working experience of 0-5 years, 22% had experience of between 6-10 years, followed by 24% between 11-15 years, 13% had experience of 16-20 years and lastly 9% had experience of more than 20 years. In terms of industry size, Figure 8 shows that 56% had over 1000 employees, whereas 19% had 51-250 employees, 13% had below 50 employees and lastly 12% had 251- 1000 employees.

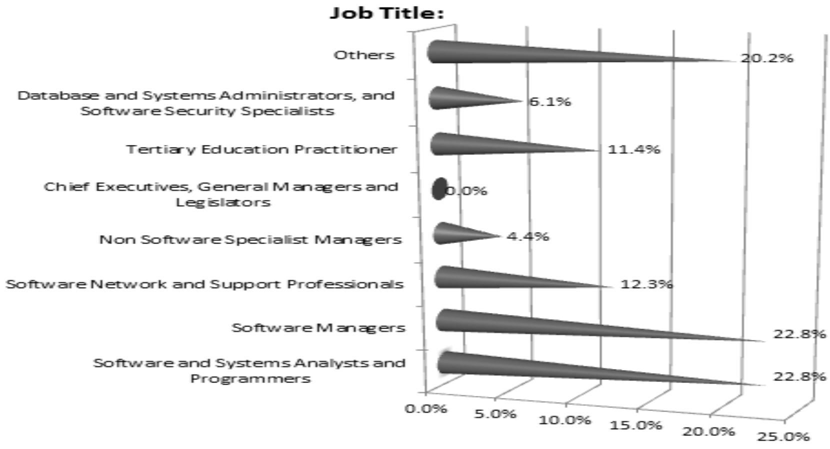

Figure 9. Demographic data for job title.

Figure 9 shows the job titles of the respondents. 23% were software and systems analysts and software programmers, 20% occupied other positions not listed in the option list, 23% were software managers, 12% were software network and support professionals, 12% were tertiary education practitioners. Besides these 6% were database and systems administrators, and software security specialists. 5% were non-software specialist managers. None of the respondents occupied the position of chief executive, general manager and legislator's positions.

Regarding the industry is founding date, Figure 10 shows that 32% of the respondents' industries were established between 1991-2000, whereas 32% of the respondents' industries were founded between 2001- 2010. 21% of the respondents' industries were founded between 1967-1980, another 7% were founded between 2011-till date (2017). Lastly 9% was established before 1967. Upon investigating the industry is annual revenue, Figure 10 shows that 36% claimed their organization had annual revenues ranging from RM 9,000,000 or above. 17% had annual revenues of RM 90,000 or below, whereas 13% had annual revenues of RM 90,000 to RM 900, 000 and another 11% had annual revenues of RM 900,000 to RM 2, 700,000. 14% of the respondents' industries had annual revenues of RM 2,700,000 to RM 4,500,000 and lastly 9% of the respondents, industries had annual revenues ranging from RM 4, 500,000 to RM 9, 000,000.

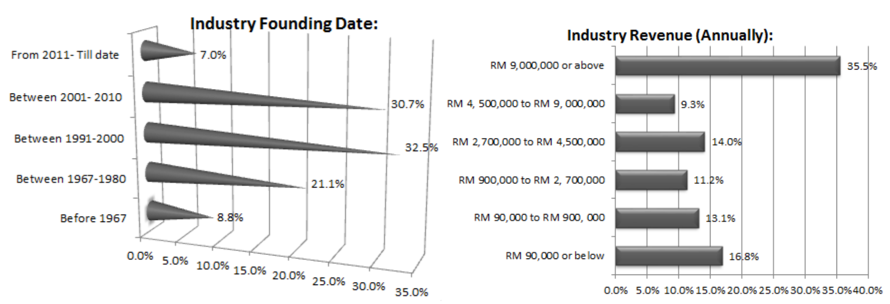

Figure 10. Demographic data distribution for industry founding date and industry yearly revenue

Descriptive Statistics

With regards to the predictors that influence sustainable software practice application, the indicators that measure the predictors as presented in Table 1 are empirically explored in this section.

Table 3

Frequency and Descriptive Statistic Results for Software Practitioner Predictor

How important are the following in influencing your organization sustainable software practice application? (Software practitioners)

Frequency response in percentage						***Descriptive statistics***				
Indicators	**Not important 1**	**2**	**3**	**4**	**Very important 5**	**Mean**	**Standard deviation**	**Min**	**Max**	**Median**
SP1	0.00%	0.00%	7.45%	41.49%	51.06%	4.44	0.63	3.00	5.00	5.00
SP2	0.00%	0.00%	7.45%	41.49%	51.06%	4.44	0.63	3.00	5.00	5.00
SP3	0.00%	0.00%	12.77%	53.19%	34.04%	4.21	0.65	3.00	5.00	4.00
SP4	0.00%	0.00%	15.22%	45.65%	39.13%	4.24	0.70	3.00	5.00	4.00
SP5	0.00%	1.06%	15.96%	46.81%	36.17%	4.18	0.73	3.00	5.00	4.00
SP6	0.00%	1.08%	13.98%	37.63%	47.31%	4.31	0.75	2.00	5.00	4.00
SP7	1.08%	1.08%	11.83%	43.01%	43.01%	4.26	0.79	2.00	5.00	4.00
SP8	0.00%	0.00%	9.68%	33.33%	56.99%	4.47	0.67	1.00	5.00	4.00

Table 3 shows the results to verify the software practitioner predictor. The findings confirm that all indicators are important; however, SP8 has the highest frequency where 56.99% believe this indicator is very important. The mean values for all indicators are > 4.00 and standard deviation < 1 and nearer to "0". Table 3 also reveals that the action, ethical beliefs, commitment and attitude of software practitioners influence sustainable software practice application in the electronic industries based on the higher mean being greater than 4. The lower standard deviation number and the median range from 4 to 5.

Table 4

Frequency and Descriptive Statistic Results for Software Governance Predictor

How important are the following in influencing your organization towards sustainable software practice application? (Software governance)										
Frequency response in percentage						***Descriptive statistics***				
Indicators	**Not important 1**	**2**	**3**	**4**	**Very important 5**	**Mean**	**Standard deviation**	**Min**	**Max**	**Median**
SG1	0.00%	0.00%	13.83%	44.68%	41.49%	4.28	0.69	3.00	5.00	4.00
SG2	0.00%	0.00%	7.45%	39.36%	53.19%	4.46	0.63	3.00	5.00	5.00
SG3	0.00%	0.00%	8.60%	35.48%	55.91%	4.47	0.65	3.00	4.00	5.00
SG4	0.00%	0.00%	8.51%	48.94%	42.55%	4.34	0.63	3.00	4.00	4.00
SG5	1.06%	2.13%	10.64%	44.68%	41.49%	4.23	0.80	1.00	4.00	4.00
SG6	0.00%	0.00%	5.26%	45.26%	49.47%	4.44	0.59	3.00	4.00	4.00
SG7	0.00%	0.00%	6.38%	47.87%	45.74%	4.39	0.61	3.00	4.00	4.00
SG8	0.00%	1.06%	8.51%	39.36%	51.06%	4.40	0.69	2.00	5.00	5.00

Table 4 shows the results to verify the software governance predictor. It can be seen that all indicators for these predictors are important based on the frequency findings where most of the respondents selected either 4 as important or 5 as very important. The mean value for all items are >=2.5 and the standard deviation values ranged from 0.59 for SG6 and 0.80 for SG5. Findings from the high frequency and mean values in relation to the low standard deviation show that the software governance policy initiated influence sustainable software practice application. Thus management support and incentives provided will aid Green in sustainable software development operations.

Table 5 shows the results to confirm the technologies and system predictor. The findings show that all the indicators are either relevant or very relevant, although the finding also reveals that TS4 and TS5 values are low compared to other indicators as seen by their mean values of 3.96 and 3.97 respectively.

Table 5

Frequency and Descriptive Statistic Results for Technologies and System Predictor

How relevant are the following in influencing your organization towards sustainable software practice application? (Technologies and systems)

	Frequency response in percentage					*Descriptive statistics*				
Indicators	Not relevant 1	2	3	4	Very relevant 5	Mean	Standard deviation	Min	Max	Median
TS1	1.06%	0.00%	10.64%	45.74%	42.55%	4.29	0.74	1.00	5.00	4.00
TS2	2.13%	1.06%	9.57%	40.43%	46.81%	4.29	0.85	1.00	5.00	4.00
TS3	0.00%	4.26%	8.51%	44.68%	42.55%	4.26	0.78	2.00	5.00	4.00
TS4	1.06%	6.38%	18.09%	44.68%	29.79%	3.96	0.91	1.00	5.00	4.00
TS5	1.06%	6.38%	17.02%	45.74%	29.79%	3.97	0.90	1.00	5.00	4.00
TS6	1.06%	5.32%	9.57%	44.68%	39.36%	4.16	0.88	1.00	5.00	4.00
TS7	1.06%	4.26%	15.96%	45.74%	32.98%	4.05	0.87	1.00	5.00	4.00
TS8	2.13%	4.26%	17.02%	42.55%	34.04%	4.02	0.93	1.00	5.00	4.00

However, the standard deviation values for all indicators are acceptable since all values are below 1 showing that the participants' responses are similar and do not deviate much from each other. The result confirms that the technologies and system deployed in the electronic industry do influence sustainable software practice application, hence an industry that practices paperless process, deploys virtualization and uses teleconferencing will reduce energy usage and lessen the cost incurred and also decrease CO2 emission, thus caring for the natural environment.

Table 6 shows the results for the pressure predictor. All the indicators are influential to sustainable software practice application with PS1 and PS7 each having a value of 54.65%. The mean value of all the indicators were greater than 2.5 thresholds. With regards to the standard deviation values for all the indicators the values are less than "1" apart from PS5 which has a value of 1. The findings from Table 4 disclose that pressure from government, non-

Table 6

Frequency and Descriptive Statistic Results for Pressure Predictor

How does the following influence your organization to sustainable software practice application? (Pressure)										
Frequency response in percentage						*Descriptive statistics*				
Indicators	**Not influential 1**	**2**	**3**	**4**	**Very influential 5**	**Mean**	**Standard deviation**	**Min**	**Max**	**Median**
PS1	0.00%	5.81%	15.12%	54.65%	24.42%	3.98	0.79	2.00	5.00	4.00
PS2	0.00%	1.16%	11.63%	47.67%	39.53%	4.26	0.70	2.00	5.00	4.00
PS3	0.00%	5.81%	13.95%	52.33%	27.91%	4.02	0.81	2.00	5.00	4.00
PS4	1.16%	11.63%	16.28%	53.49%	17.44%	3.74	0.92	1.00	5.00	4.00
PS5	3.49%	9.30%	20.93%	46.51%	19.77%	3.70	1.00	1.00	5.00	4.00
PS6	2.33%	9.30%	17.44%	46.51%	24.42%	3.81	0.98	1.00	5.00	4.00
PS7	2.33%	3.49%	16.28%	54.65%	23.26%	3.93	0.86	1.00	5.00	4.00

governmental bodies, management and end users can influence electronic industries to apply sustainable software. Furthermore the provision of government incentives and other resources and the actions of other industrial competitors also have an effect on sustainable software practice application in the electronic industries grounded on the median value of 4 selected by the respondents in relation to the pressure predictor.

Table 7 shows the results to verify the software strategy predictor. The findings show that all the indicators are important, although the findings also disclose that SS5, SS6 and SS7 are more important in the respondents' respective industries. The mean value for all the indicators is extremely encouraging with all values higher than 4.0. The standard deviation values for all indicators are acceptable since all the values are below 1 showing that the participants' responses are similar and do not deviate from each other. Additionally, the results show that the tackling of CO2 emission of the software-based systems, putting in place initiatives on how to achieve environmental goals and designing enterprise opportunities based on sustainability dimensions to be attained influence how the industry will apply sustainable software as supported by the maximum value of 5 which was computed from all the respondents.

Table 7

Frequency and Descriptive Statistic Results for Software Strategy Predictor

How important are the following in influencing your organization to sustainable software practice application? (Software strategy)										
Frequency response in percentage						*Descriptive statistics*				
Indicators	**Not important 1**	**2**	**3**	**4**	**Very important 5**	**Mean**	**Standard Deviation**	**Min**	**Max**	**Median**
SS1	0.00%	3.45%	16.09%	44.83%	35.63%	4.13	0.80	2.00	5.00	4.00
SS2	0.00%	4.60%	13.79%	42.53%	39.08%	4.16	0.83	2.00	5.00	4.00
SS3	1.15%	2.30%	11.49%	43.68%	41.38%	4.22	0.82	1.00	5.00	4.00
SS4	0.00%	2.30%	10.34%	48.28%	39.08%	4.24	0.73	2.00	5.00	4.00
SS5	0.00%	2.30%	11.49%	57.47%	28.74%	4.13	0.69	2.00	5.00	4.00
SS6	0.00%	2.30%	10.34%	55.17%	32.18%	4.17	0.70	2.00	5.00	4.00
SS7	1.16%	1.16%	12.79%	55.81%	29.07%	4.10	0.75	1.00	5.00	4.00

Table 8

Frequency and Descriptive Statistic Results for Knowledge Accessibility Predictor

How important are the following in influencing your organization towards sustainable software practice application? (Knowledge accessibility)										
Frequency response in percentage						*Descriptive statistics*				
Indicators	**Not important 1**	**2**	**3**	**4**	**Very important 5**	**Mean**	**Standard deviation**	**Min**	**Max**	**Median**
KA1	0.00%	2.30%	13.79%	39.08%	44.83%	4.26	0.78	2.00	5.00	4.00
KA2	0.00%	1.15%	14.94%	44.83%	39.08%	4.22	0.73	2.00	5.00	4.00
KA3	0.00%	2.35%	14.12%	40.00%	43.53%	4.25	0.78	2.00	5.00	4.00
KA4	0.00%	3.45%	12.64%	44.83%	39.08%	4.20	0.79	2.00	5.00	4.00

Table 8 illustrates the results to verify the knowledge accessibility predictor. The result shows that all the four indicators are important. The mean values for all the indicators are > 4.0 which is > 2.5 thresholds. The standard deviation values for all the indicators are satisfactory since all the values are below 1 and closer to 0 point showing that the participants' responses are similar and do not diverge from each other. The result also shows that consistent precise unconstrained information relating to the environment and climate disseminated across the electronic industry influences sustainable software practice application in the enterprise. Hence, electronics industries that provide data to communicate and have access to unconstrained information are more plausible in applying sustainable practice. This is supported by the mean values from our study which confirmed that respondents believe that knowledge is a necessity in attaining sustainable software, thus knowledge accessibility is important and it influences Green sustainable software development in electronic industries.

Inferential Statistics of Predictors

This section aims to report on the significance of the predictors in influencing sustainable software practice. Hence, regression analysis was carried out in SPSS version 22 to run an inferential test and validate the correlation or association between the predictors and the dependent variable as seen in Table 9.

Table 9

Regression Results for Predictors

Dependent variable: **Sustainable Software practice application**	**Regression analysis**					
Independent variables	**F-tests**	**R2**	**Beta**	**Standard error**	**t-test**	**P-value (Sig.)**
Software practitioners	15.271	0.104	0.323	0.091	3.908	0.000
Software governance	26.315	0.167	0.409	0.083	5.130	0.000
Technologies and systems	46.213	0.261	0.511	0.079	6.798	0.000
Pressure	69.018	0.345	0.587	0.081	8.308	0.000
Software strategy	52.742	0.287	0.536	0.077	7.262	0.000
Knowledge accessibility	56.012	0.300	0.547	0.072	7.484	0.000

The result of the inferential test using regression analysis between predictors (software practitioners, software governance, technologies and systems, pressure, software strategy and knowledge accessibility) and the dependent variable (sustainable software practice application) is shown in Table 9 where the results outline the goodness of fit relationship test, namely F-test for all predictors given as 15.271, 26.315, 46.213, 69.018, 52.742 and 56.012 with p-value 0.000 outlining the test is highly significant for all predictors.

Since the p-value of the F-test is less than the significant level α=0.05, it was concluded there was a significant relationship between the predictors and the dependent variable (sustainable software practice application). The strength of the relationships between the predictors and the sustainable software practice application is measured by examining R2 of all the predictors, where R2= 0.104 for software practitioners shows that the software practitioners predictor has been interpreted at 10.4% of the variance in sustainable software practice application. Next is software governance with R2 =0.167 interpreting at 16.7% of the variance.

Technologies and systems have an R2 =0.261 interpreting at 26.1% of the variance. Next is the pressure predictor which has the highest R2 =0.345 interpreting at 34.5% of the variance, followed by the software strategy with R2 =0.287 interpreting at 28.7% of the variance and lastly is knowledge accessibility with R2 =0.300 interpreting at 30% of the variance in sustainable software practice application confirming that there exists a strong relationship associated with the predictors and sustainable software practice application. Additionally, all the predictors have a direct effect on the sustainable software practice application (as shown by the positive beta result (β = 0.323, 0.409, 0.511, 0.587, 0.536, 0.547)), which expresses the relative importance of the predictors and the collinearity statistics. In terms of improvement, an increase of 1 unit in all predictors, sustainable software practice application will increase by 0.32 (32%) unit for software practitioners, 0.409 (40.9%) unit for software governance, 0.511(51.1%) unit for technologies and systems, 0.587 (58.7%) unit for pressure, 0.536 (53.6%) unit for software strategy, and lastly 0.547 (54.7%) unit for knowledge accessibility.

Lastly considering that the t-test value of all the predictors (3.908, 5.130, 6.798, 8.308, 7.262, 7.484) were higher than the 1.96 benchmark as recommended by Hair et al. (2010), all the identified predictors in this study were very significant, with pressure predictor being the most important predictor at t = 8.308, p = 0.000 and software practitioners being the least important predictor at t = 3.908, p = 0.000. Therefore, it can be concluded that software practitioners, software governance, technologies and systems, pressure, software strategy and knowledge accessibility significantly influence sustainable software practice application in the electronic industries based in Malaysia.

DISCUSSION

In recent years, Green sustainable research in software process has been emerging because of the important need for sustainability in industries.

According to Amri & Saoud, (2014) sustainability attainment in the software engineering domain characterizes how software products are developed so that positive and negative effects induced from software operations are discovered, optimized and documented throughout software product's life cycle. In the electronic industry, sustainability can be attained with the utilization of programs and approaches that minimize energy usage and waste generation. Hence, for the software to be sustainable the software should have direct and indirect adverse impacts on the environment, society, human and the economy resulting from design, deployment, and utilization of the software should be negligible and/or has a positive influence on sustainable development. Past research in sustainable software development concentrated on the descriptions, metrics and technical solutions for environmental-friendly software, but few have addressed sustainable software development from the electronic industry's perspective. Hence, there is a need to explore the predictors that address the usage and development of sustainable software. Therefore, this study identified the predictors of sustainable software practices application in the electronic industries in relation to social, environmental, technical, economic and individual dimensions of sustainability.

Findings from this study show that the software practitioners are important and thus influence sustainable software practice application. Since these are the people that carry out software development process in their organizations, how they apply eco-friendly operations will determine the outcome of sustainability attainment. This is similar to the finding presented by Mishra et al. (2014). They found out that people's is norms, values and beliefs in the organization influence environmental practices for the application of Green IT. This also correlated with the finding presented by Akman & Mishra (2014) where the authors found out that Green IT practices influence IT professionals' actions. The analysis of our results also provides strong support for software governance which reveals that the management policies, rules and regulations influence sustainable software practice application in the electronic industries. Also the financial support provided by the management in electronic industries will determine if software developers will deploy ecological-friendly practices. This is consistent with the findings provided by Penzenstadler (2014) in her research on implementing Green requirements for Green through and in software products.

The finding of this research also supports the finding presented by Lami & Buglione (2012) on evaluating software sustainability where the authors mentioned that the Green information technologies and system applied during the software development process influence energy usage and CO2 emission. Furthermore, findings regarding pressure as one of the predictors that influences sustainable software practice application corroborates Jenkin et al. (2011) and Vykoukal et al. (2011). The authors stated that pressure within the organization by the management, pressure from governmental and non-governmental associations and pressure from end-users of services produced by the organization motivate the organization towards sustainable practice application. With regards to results for software strategy, the findings show that the strategy deployed to facilitate the day-to-day software development operation in the industry definitely influences sustainable software practice application. This is similar to the findings provided by Deng & Ji (2015) where the researchers provided evidence of Green IT adoption in organizations. The findings outlined that strategy based on the activities and procedures deployed in the industry achieve the aims and objectives of the organization. Results related to knowledge accessibility suggest that this predictor also influences sustainable software practice application. Hence best Green practice knowledge of sustainable software development can be created, stored, retrieved, transferred and applied by the electronic industry when developing software for sustainability attainment. This is analogous with the discussion put forward by Abdullah et al. (2015) on the importance of knowledge in sustainable software development. Finally, the results presented in this study confirm that all predictors (software practitioners, software governance, technologies and systems, pressure, software strategy and knowledge accessibility) presented in Figure 4 influence sustainable software practice application and no need predictors were derived from the survey. Therefore, the results presented in this study correlate with sustainable software practice application in terms of environmental, technical, social, individual and economic are made possible based on the indicators used to measure sustainable software practice application. Hence if all indicators are practiced in electronic industries, each of the dimensions (economic, social, environmental, technical and individual) will be adequately attained in the industry. Hence if a particular electronic industry applies all sustainable software practices presented as indicators in Table 1, the sustainability dimension (economic, social, environmental, technical and individual) can be achieved.

IMPLICATION OF THE STUDY

Based on issues such as climate change and global warming, sustainability has become an important domain, where the term sustainability was derived from the Latin word "sustenere" which means to hold up or show the competency of persistence over time (United Nations, 1987; Lami & Buglione, 2012). Sustainability which refers to addressing the needs of the present day without compromising the capability of future generations to address their own needs

(Brundtland, 1987) by considering the environmental, societal and economy dimensions as being interdependent instead of independent. Sustainability aims to decrease the environmental impact while supporting social justice and economic growth (Albertao et al., 2010). Environment conservation for sustainable development is the most significant agenda industries aim to accomplish in the next decade. Sustainability embodies one of the most feasible conceptions to be examined and understood by being beneficially applied in the software development process.

At present it seems to be typically related to environmental issues, while it should be addressed from a broader outlook, taking into account other dimensions in conserving the natural resources for the long term and not the short term. Sustainability in electronic industries can help preserve hardware resources in the software development process to produce Green sustainable software that is implemented on time, within budget and on quality (Calero et al., 2013). It is obvious that software deployed systems have led to intense advancement in human civilization, but simultaneously they have added considerably to the utilization of the Earth's natural resources. However, over the years the development of web-based cloud services utilized during software development has made software services more scalable. Moreover, these developments have huge implications to the Green software research domain since sustainability can be enhanced by applying these software services which offer novel paradigms to decrease energy consumption and also care for the natural environment (Dustdar et al., 2013).

Furthermore, software-based systems are mostly associated with the day-to-day operations in electronic industries and different aspects of life. As such, software provides a dominant leverage point for supporting sustainability attainment to be brought to different domains. However, a few software developers lack understanding of how to apply sustainability into their current software development processes (Roher & Richardson, 2013). Hence this research provides implication for software developers by presenting the predictors that influence sustainable software development, the dimensions to be considered for sustainable software development and lastly the software development life cycle to be applied in attaining sustainability. Therefore the practical implication of this study is that it enhances the ability of software developers in the electronic industries to develop software systems that possess less negative effects on the natural environment.

Furthermore, this study increases software developers' general knowledge of the sustainable software process and related software practice application life cycle. This study also contributes to Green software engineering by providing a comprehensive set of best practice for electronic industries (as seen in Table 1) to facilitate Green sustainable practice application in the electronic industry. The research implication of this study provides an opportunity for creating environmental awareness of the numerous sustainability issues existing in developing and utilizing software systems in electronic industries. Software developers and engineers are provided with the prospect to aid the United Nations development goals for sustainable development by resolving the needs of the present day without conceding the ability of the generations to come to meet their own needs, thus contributing to decrease the environmental issues that influence today's society.

CONCLUSION, LIMITATIONS AND FUTURE WORKS

The software engineering scientific community has comprehended the need to evolve toward sustainability as proven by available research papers published in journals, conference proceedings, book chapters, technical reports, etc. It is the right time to address the impact software systems possess on the natural environment. Although software developers are faced with complex intellectual barriers that arise based on the inter-disciplinary nature of sustainability, resulting in a gap of knowledge that address environmental issues in electronic industries. However, Green sustainable software development practice application may be more expensive in the beginning and this may discourage software developers to practice sustainable software development, but it is to be noted that the long-term benefits of sustainability such as helping to reduce the global climate is viable (Roher & Richardson, 2013). Hence, Green sustainable software development in electronic industries may consider first the runtime energy utilization of software as stated by Steigerwald and Agrawal (2011) and software development life cycle as suggested by Dick et al. (2010). Secondly, Green sustainable software development may explore the imperative role that software plays in all plausible aspects of the society, and also examine how software can be used to progress industries, social and individual activities towards sustainability.

This study mainly focused on the second direction to leverage software to solve sustainability problems in a broader scope in research and development. Although this study did not consider how to advance the runtime energy utilization of software, it identified and presented the software development life cycle, the predictors that influence sustainable software practice application in electronic industries, and lastly the dimension of sustainability to be considered for sustainable software development. Data was collected using a survey of 133 respondents who had experience in Green sustainable practice application. The survey data was used to verify each of the identified predictors that influence sustainable software practice application. Descriptive statistics was used to present the frequency, mean, standard deviation, median, and maximum and minimum results. The findings show that each of the predictors is relevant and influences sustainable software development. Inferential statistics was also carried out using regression analysis to test the significance of the identified predictors. The findings of the regression analysis show that all the predictors are highly significant.

The limitation of the research is that the research model investigated the influence of the predictors on the sustainable software practice application only. This research did not investigate the life cycle processes that are practically applied for sustainable software development in attaining the social, economic, environmental, technical and individual dimensions of sustainability. Furthermore, the data was collected from respondents in Malaysia only and cannot be generalized to other countries. Therefore, future work will examine the life cycle processes that are practically applied for Green sustainable software development in electronic industries and data will be collected from respondents in other regions or countries.

REFERENCES

Abdullah, R., Abdullah, S., & Tee, M. (2014). Web-based knowledge management model for managing and sharing green knowledge of software development in community of practice. *2014 8th Malaysian Software Engineering Conference (MySEC),* 210-215.

Abdullah, R., Abdullah, S., Din, J., & Tee, M. (2015). A systematic literature review of green software development in collaborative knowledge management environment. *International Journal of Advanced Computer Technology (IJACT)*, *9*, 136.

Ainin, S., Naqshbandi, M. M., & Dezdar, S. (2016). Impact of adoption of Green IT practices on organizational performance. *Quality & Quantity*, *50*, 1929-1948.

Akman, I., & Mishra, A. (2014). Green information technology practices among IT professionals: Theory of planned behavior perspective.

Albertao, F., Xiao, J., Tian, C., Lu, Y., Zhang, K. Q., & Liu, C. (2010). Measuring the sustainability performance of software projects. *2010*

IEEE 7th International Conference on e-Business Engineering (ICEBE), 369-373.

Ali, A., Nor, R.N.H., Abdullah, R., & Murad, M.A.A. (2016). Developing conceptual governance model for collaborative knowledge management system in public sector organisation. *Journal of Information & Communication Technology (JICT)*, *15*, 171-191.

Amri, R., & Saoud, N. B. B. (2014). Towards a generic sustainable software model. *Fourth International Conference on Advances in Computing and Communications,* 231-234.

Anthony, B. J. (2016). Green information systems integration in information technology-based organizations: An academic literature review. *Journal of Soft Computing and Decision Support Systems*, *3*, 45-66.

Anthony, B. J., & Majid, M. A. (2016a). Green IS for sustainable decision-making in software management. *Journal of Soft Computing and Decision Support Systems*, *3*, 20-34.

Anthony, B. J., & Majid, M. A. (2016b). Development of a Green ICT model for sustainable enterprise strategy. *Journal of Soft Computing and Decision Support Systems*, *3*, 1-12.

Boudreau, M. C., Chen, A., & Huber, M. (2008). Green IS: Building sustainable business practices. *Information systems: A global text*, 1-17.

Brundtland, G. H. (1987). World commission on environment and development. *Environmental Policy and Law*, *14*, 26-30.

Calero, C., Bertoa, M. F., & Moraga, M. Á. (2013). A systematic literature review for software sustainability measures. *Proceedings of the 2nd International Workshop on Green and Sustainable Software,* 46-53.

Deng, Q., & Ji, S. (2015). Organizational Green IT adoption: Concept and evidence. *Sustainability*, *7*, 16737-16755.

Dick, M., & Naumann, S. (2010). Enhancing software engineering processes towards sustainable software product design. In *EnviroInfo,* 706-715.

Dick, M., Naumann, S., & Kuhn, N. (2010). A model and selected instances of green and sustainable software. *What kind of information society?*

Governance, virtuality, surveillance, sustainability, resilience. Springer Berlin Heidelberg, 248-259.

Dustdar, S., Li, F., Truong, H. L., Sehic, S., Nastic, S., Qanbari, S., & Claeßens, M. (2013). Green software services: From requirements to business models. *Proceedings of the 2nd International Workshop on Green and Sustainable Software,* 1-7.

Erdelyi, K. (2013). Special factors of development of green software supporting eco-sustainability. *2013 IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY),* 337-340.

Grant, N., & Marshburn, D. (2014). Understanding the enablers and inhibitors of decision to implement green information systems: A theoretical triangulation approach. *Twentieth Americas Conference on Information Systems., Savannah*.

Hair, J. F., Black, W. C., Babin, B. J., and Anderson, R. E. (2010). Multivariate data analysis: A global perspective. Pearson Education Inc., Upper Saddle River, NJ, USA.

Harmon, R. R., & Auseklis, N. (2009). Sustainable IT services: Assessing the impact of green computing practices. *Portland International Conference on Management of Engineering & Technology*, 1707-1717.

Howard, G. R., & Lubbe, S. (2012). Synthesis of green is frameworks for achieving strong environmental sustainability in organisations. *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference,* 306-315.

ISO/IEC IS 12207. (2008). Systems and software engineering-Software life cycle processes, *International Organization for Standardization*.

ISO/IEC IS 15504. (2011). Information technology- Process assessment, Parts 1-10, *International Organization for Standardization.*

Jenkin, T. A., Webster, J., & McShane, L. (2011). An agenda for 'Green'information technology and systems research. *Information and Organization*, *21*, 17-40.

Jnr, B. A., Majid, M. A., & Romli, A. (2017). Application of intelligent agents and case-based reasoning techniques for green software development. *TTEM*, *12*, 30-43.

Johann, T., Dick, M., Kern, E., & Naumann, S. (2011). Sustainable development, sustainable software, and sustainable software engineering: An integrated approach. I*nternational Symposium on Humanities, Science & Engineering Research (SHUSER),* 34-39.

Karanasios, S., Cooper, V., Deng, H., Molla, A., & Pittayachawan, S. (2010). Antecedents to greening data centres: A conceptual framework and exploratory case study. *Australasian Conference Proceedings on Information System*, 1-3. Retrieved from http://aisel.aisnet.org/alis2010/54

Kern, E., Dick, M., Naumann, S., Guldner, A., & Johann, T. (2013). Green software and green software engineering–Definitions, measurements, and quality aspects, 87-94.

Kocak, S. A. (2013). Green software development and design for environmental sustainability. *11th International Doctoral Symposium an Empirical Software Engineering,* Maryland.

Koçak, S. A., Alptekin, G. I., & Bener, A. (2014). Evaluation of software product quality attributes and environmental attributes using ANP decision framework, 37-44.

Krishnadas, N., & Radhakrishna, R. (2014). Green information technology: Literature and research-domains. *Journal of Management Systems*, *24*, 57-79.

Kumar, R. (2005). *Research methodology: A step-by-step guide for beginners*. Sage.

Lami, G., & Buglione, L. (2012). Measuring software sustainability from a process-centric perspective. *2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA),* 53-59.

Luan, C. J., Tien, C., & Chen, W. L. (2016). Which “green” is better? An empirical study of the impact of green activities on firm performance. *Asia Pacific Management Review*, *21*, 102-110.

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

Mangla, S. K., Kumar, P., & Barua, M. K. (2015). Risk analysis in green supply chain using fuzzy AHP approach: A case study. *Resources, Conservation and Recycling*, *104*, 375-390.

McGibbon, C., & Van Belle, J. P. (2013). Integrating green information systems into the curriculum using a carbon footprinting case. *European Conference on Information Management and Evaluation* (p. 104). Academic Conferences International Limited.

Mishra, D., Akman, I., & Mishra, A. (2014). Theory of reasoned action application for green information technology acceptance. *Computers in Human Behavior*, *36*, 29-40.

Moshnyaga, V. G. (2013). An assessment of software lifecycle energy. *23rd International Workshop on Power and Timing Modeling, Optimization and Simulation,* 112-119.

Negulescu, O., & Doval, E. (2014). Managers' position against risk, uncertainty and efficiency within the Green industry: Ten questions. *Procedia-Social and Behavioral Sciences*, *124*, 339-343.

Nifa, F. A. A., Nawi, M. N. M., Nadzri, W., & Osman, S. A. R. (2015). Towards development of sustainable design in Malaysian university campus: A preliminary framework for Universiti Utara Malaysia, *Jurnal Teknologi*, *77*, 43–49.

Penzenstadler, B. (2012). Supporting sustainability aspects in software engineering. *3rd International Conference on Computational Sustainability (CompSust)*.

Penzenstadler, B. (2014). Infusing Green: Requirements engineering for Green in and through software systems. In *RE4SuSy@ RE*, 44-53.

Roher, K., & Richardson, D. (2013). A proposed recommender system for eliciting software sustainability requirements. *2nd International Workshop on User Evaluations for Software Engineering Researchers (USER),* 16-19.

Savita, K. S., Dominic, P. D. D., & Ramayah, T. (2014). The adoption of Green information technologies and systems as a driver within Green SCM. *International Conference on Computer and Information Sciences (ICCOINS)*, 1-6.

Scanniello, G., Erra, U., Caggianese, G., & Gravino, C. (2013). Greening an existing software system using the GPU. *Software-Practice & Experience*, 1-21.

Shenoy, S. S., & Eeratta, R. (2011). Green software development model: An approach towards sustainable software development. *Annual IEEE India Conference (INDICON)*, pp. 1-6.

Simmonds, D., & Bhattacherjee, A. (2014). Green IT adoption and sustainable value creation.

Steigerwald, B., & Agrawal, A. (2011). Developing green software. *Intel White Paper*, *9*.

Surendro, K., Supriana, A., & Supriana, I. (2016). Requirements engineering for cloud computing adaptive model. *Journal of Information & Communication Technology (JICT)*, *15*, 1-17.

Thiry, M., Frez, L., & Zoucas, A. (2014). GreenRM: Reference model for sustainable software development. *SEKE,* 39-42.

Uddin, M., Hindu, R. C., Alsaqour, R., Shah, A., Abubakar, A., & Saba, T. (2015). Knowledge management framework using green IT to implement sustainable entrepreneur ecosystem. *Applied Mathematics & Information Sciences*, *9*, 2703.

United Nations General Assembly. (1987). Report of the World Commission on environment and development: Our common future. Transmitted to the General Assembly as an Annex to document A/42/427 - Development and International Co-operation: Environment.

Vykoukal, J., Beck, R., & Wolf, M. (2011). Impact of pressure for environmental sustainability on grid assimilation–Empirical results from the financial services industry. *Australasian Journal of Information Systems*, *17*, 5-28.