sustainability

Article

A Catalogue Supporting Software Sustainability Design

Shola Oyedeji [1,*], Ahmed Seffah [1] and Birgit Penzenstadler [2]

[1] LUT School of Engineering Science (LENS), Lappeenranta University of Technology, 53850 Lappeenranta, Finland; ahmed.seffah@lut.fi
[2] Department of Computer Engineering and Computer Science, California State University Long Beach, Long Beach, CA 90840, USA; birgit.penzenstadler@csulb.edu
* Correspondence: shola.oyedeji@lut.fi; Tel.: +358-4-1369-7708

Received: 30 May 2018; Accepted: 27 June 2018; Published: 3 July 2018

Abstract: Like other communities, sustainability in and for software design is a grand research and development challenge. Current research focuses on eliciting the meanings of sustainability and on building approaches for its engineering and integration into the mainstream software development lifecycle. However, few concrete guidelines that software designers can apply effectively are available. A guideline aims to streamline the design processes according to a set of well-known research routines or sound industry practices. Such guidelines can help software developers in the elicitation of sustainability requirements and testing software against these requirements. This paper introduces a sustainability design catalogue (SSDC) comprising a series of guidelines. It aims to assist software developers and managers in eliciting sustainability requirements, and then in measuring and testing software sustainability. The catalogue is based on reviews of the current and past research on sustainability in software engineering, which are the grounds for the development of the catalogue. Four different case studies were analyzed using the Karlskrona manifesto principles on sustainability design. A pilot framework is also proposed that includes a set of sustainability goals, concepts and methods. It exemplifies how to apply and quantify sustainability.

Keywords: sustainability; software sustainability; information and communication technology; software design; sustainability requirement; software sustainability analysis; software sustainability guidelines; Karlskrona manifesto

1. Introduction

Software sustainability and software engineering for sustainability are now recognized as timely important concerns not only for researchers, but also for the entire software industry and standardization bodies. A Microsoft report as well as an IBM global chief executive officer (CEO) study on sustainability showed an increasing growth in the percentage of organizations redesigning their entire business models to incorporate sustainability [1–3]. The Sustainability and Innovation Global Executive Study indicates that 48% of respondents out of the 4000 executives and managers interviewed worldwide agree that sustainability urged them to modify their business models [4,5]. Sustainable development is also driving software innovations for creating new opportunities to cut costs, adding value and gaining competitive advantage [6]. As software is the catalyst for economic and social changes today [7] and the pillar for all industries, there is a huge pressure from regulators and civil society to develop more green software that uses less energy [8].

As a matter of fact, the Ericsson sustainability report shows that information and communications technology (ICT) could help reduce global greenhouse gas (GHG) emissions by up to 15%. It forecasts that by 2021, 28 billion devices will be connected to each other [9] which will increase energy consumption drastically. Another energy and carbon report from Ericsson forecasts 90% of the world's population to have mobile coverage, and 60% will have the ability to access high-speed long-term evolution (LTE) data networks [10]. Such reports are clear indicators of the huge sustainability impact of ICT. Overall, the ICT sector contributes around 2% of the global CO_2 emissions. It is also accountable for approximately 8% of the European Union's (EU) electricity consumption and 2% of the carbon emissions from ICT devices and services [6].

It is therefore important to look how to reduce the impact of ICT on the environment and how sustainability can be incorporated better into the software development lifecycle. However, current software development practices do not provide sufficient support to all sustainability concerns. This should not be limited to energy consumption, but it should include also all the other aspects of sustainability.

For example, [6] provides a more broader perception of sustainability in software engineering. Sustainability may or should refer also to, for example, electronic waste management and the ecological impacts of recycling the drastically increasing amount of computing gear. Moreover, it is worth mentioning that the impact of the cryptocurrency market on energy consumption is also a very serious problem from a sustainability perspective [11]. The energy consumption of blockchain technologies has raised environmental concerns. The resulting energy consumption per year is estimated at 12.76 TWh [12]. The digital infrastructure supporting the wide diversity of interactive devices and services available on the cloud accounts for up to 85% of total environmental impact [13].

Therefore, the meanings and integration of sustainability should cover the five dimensions of software sustainability [14–16]: economic, social, individual, environmental and technical [17]. The economic, technical and business dimensions are now core aspects of fundamental values in companies embracing sustainable development [18]. An area that has received less attention is the social dimension. It entails the well-being of the software users community and developers [14], and is about changing the human mindset and designing their perceptions and experiences of sustainability.

Practices and processes that are widely used in an industry setting such as agile methodologies and model-driven approaches lack aspects addressing sustainability challenges [19]. Practitioners are not prepared for integrating sustainability efficiently and effectively. Where should sustainability ingredients be considered? Indeed, the different sustainability dimensions have no reference framework that can assist software developers. Researchers also highlighted the vital need to define measures of sustainability and search for avenues for their integration in the wider engineering processes [6,20].

Our research focuses mainly on the integration of sustainability during the software design stage and into the design practices. Design is a key milestone where supporters and pioneers largely recognize the importance of sustainability. Varying perspectives have been discussed such as the design of sustainability and sustainability by design [21]. Sustainability by design is one way to achieve sustainability and for integrating sustainability perceptions in software engineering [22].

The research discussed in this paper is twofold. First, based on the analysis of the literature and built on the Sustainability Design Manifesto principles, the paper introduces an original sustainability design catalogue (SSDC). Then, it describes a pilot framework that exemplifies how the SSDC can be applied and how the underlying sustainability design principles can be incorporated into the design practices for all software development life-cycle (SDLC) phases. The SSDC is a set of practical concrete guidelines and indicators supporting sustainability by design practices. The Karlskrona manifesto principles are viewed in this research as a set of high-level abstract principles and perceptions for sustainability design in and for software systems [23,24].

Overall our research addresses the following specific questions and provides the following contributions:

1. How do the principles detailed in the Karlskrona manifesto relate to the software development life-cycle phases (SDLC) in general and the design stage especially? SSDC suggests concrete guidelines to apply the high levels and abstracts principles of sustainability.

2. How do the SSDC and the underlying perceptions of sustainability relate to the first-, second- and third-order impacts of software sustainability as well as the five dimensions of sustainability? SSDC tried to bridge the current gaps between the principles and the indicators of sustainability.
3. How can these principles be applied while ensuring that sustainability is achieved during design? The SSDC can be viewed as a tool supporting the sustainability by design approach.
4. How should these principles be applied for the wide diversity of software systems that exists today and those in the future that should consider sustainability as a quality in the same way we engineer the other quality attributes today of such security and usability? A pilot framework is described portraying the applicability of the SSDC for diverse software systems.

The remainder of this paper is as follows. Section 2 covers the background and related research work on sustainability in software design. Section 3 presents the foundation of the Karlskrona manifesto principles and how these relate to the software development life-cycle phases. Section 4 details the structure and components of the Software Sustainability Design Catalogue (SSDC). Section 5 discusses how the SSDC was derived and can be applied. Section 6 provided a practical example for the usage of the SSDC and the pilot framework. The benefits of the SSDC and pilot framework are summarized in Section 7. The conclusion summarizes with comments and future research work.

2. Sustainability in Software Design: Background and Related Research

Sustainability is one of the grand challenges of our civilization because of its pervasiveness. The way we design, and consequently use, software-intensive systems has a significant impact and can influence human perceptions of sustainability greatly [25]. Although design is a central phase of any software development process [22], there has been limited research work on software sustainability design. The most relevant related works are listed and described in the following.

Currently there is no single point of reference for researchers or practitioners where the sustainability measures are gathered and exemplified [26]. The perception of professionals about sustainability affects the way sustainability has been applied in software development [27]. However, the pathway to a sustainable society is unclear since sustainability means different things to different people [28]. People's different lifestyles, values and practices also affect how sustainability is treated [29]. Furthermore, one of the major problems for software designers is that even with a systems approach, there are few existing tools that wrap core principles of sustainability together. Instead, designers must learn to patch together a series of disparate sustainability understandings, and frameworks in order to address the different dimensions of sustainability [30]. An alternative design solution is based on the sustainable design practices that use the least energy over ICT's life cycle [31]. The global Sustainable Development Goals formally adopted by the United Nations (UN) in 2015 can serve as an inspiration. They have the potential to guide software practitioners, especially human–computer interaction (HCI) specialists [32]. In the area of cloud computing, there is not enough awareness about the value benefits of sustainability especially when selecting and deploying cloud computing software among organizations [33].

Software design as a key factor can help reduce energy consumption by 30% to 90% because software provides the real energy saving that tells hardware what to do and how to function [34]. A catalogue of sustainability guidelines has been proposed in [35]. It incorporates all phases of the system development life cycle while providing specific support to project managers, software architects, and developers during the entire system design, development, operation, and maintenance. However, it is not as detailed as the SSDC proposed in this article as it does not cover all the different sustainability dimensions (economic, environmental, social, individual, and technical), nor the first, second and third order of impacts of software systems and metrics/indicators to evaluate the effectiveness of guidelines in the catalogue.

The International Organization for Standardization (ISO) and Institute of Electrical and Electronics Engineers (IEEE) series of software engineering standards provide little guidance on sustainability. While there has been some increase in literature about the environmental and social dimension of sustainability for software systems, there is less attention on sustainability in software development and use [36]. The effects of software systems are getting less attention and work to formalize sustainability as part of software engineering process is still not considered in the official standards and models of software systems [37]. The concept of sustainability for software design, and its integration into the existing catalogue of design quality attributes is needed to achieve sustainable software [38] and sustainability should also be considered as a quality of software systems like security and usability [39]. Sustainability in software design requires a multidimensional and interdisciplinary approach [40–43]. Kern et al. [44] developed a model for green software based on sustainability criteria, although there are still several software design quality models that include attributes like flexibility and reusability, with no attribute that captures how cost-efficient are the set of design decisions over time.

A life-cycle model that uses a cradle-to-cradle approach to analyse impacts of each software product life-cycle phase can help to develop green and sustainable software products [45]. Another proposed generic model for sustainability was proposed with instances for companies and projects based on different cases studies. The proposed process helps requirements engineers to properly analyse projects during software design based on different sustainability dimensions [17]. An experience report from a case of applying standard requirements engineering methods to analyse sustainability aspects shows how requirements can impact software design [46]. Another generic model for improving the general software development process for sustainable software product design is the process enhancement model, which includes activities and artefacts such as sustainability reviews and previews, ongoing process assessments, a sustainability retrospective, and a sustainability journal. Although the model does not currently cover sustainability benchmarks, it provides a sound basis for future integration [47].

A description of how to support different aspects of sustainability in software development processes, software system analysis for production, and usage phases of the life cycle can also provide an understanding of what sustainability means in software engineering [19]. In the same vein, a study of the life-cycle activities of software development with a focus on environmental protection provided a guide through a formula to calculate software waste to encourage the development of green software [48]. The author highlights key activities during software design and development with key factors at each stage of software design and how each of these factors relates to a green aspect in software development. Also highlighted is the fact that thoroughly designed and implemented software uses energy efficiently through computational and data efficiency [48].

Researchers from different disciplines tried to tackle the issue of sustainability through collaborative work via organizing interdisciplinary conferences and workshops [24]. One common focus is sustainability in requirements. Sustainability requirements were treated as first-class quality requirements, and as such systematically elicited, analysed and documented with the goal of showing that small and easy steps during requirement can lead to the design and development of more sustainable systems [46]. This is corroborated by another research work [49] stating the need to characterize software sustainability as a quality factor in requirements elicitation. In addition, a sustainability requirements checklist and guide approach demonstrate how to include the objective of environmental sustainability from the very early steps of software development [50]. It also shows how green requirements engineering may be conducted within the scope of general purpose requirements engineering and accommodate the new objective of improving environmental sustainability [50].

The use of sustainability requirement patterns (SRPs) is another approach that provides software engineers with guidance on how to write specific types of sustainability requirements. The aim is to overcome the barriers of incorporating environmental sustainability into the requirements engineering process [51]. Sustainability requirements can also be a non-functional requirement (NFR) using an NFR framework informed by sustainability models and how it can be used to correctly obtain and describe sustainability related requirements of the software system to be developed.

The impact of ICT on the world's CO_2 emissions can be reduced through improved software-energy efficiency on multi-core systems [52] although there are few studies and suggestions about 'what' aspects of sustainability to measure and 'how' to do it with regards to ICT [53].

Sustainability should be considered in software quality models, although there has been less research channelled towards it [54]. Planning and management of software sustainability as a quality attribute is impaired by a lack of consistently applied, practical measures [55]. Without these measures, it is impossible to determine the effect of efforts to improve sustainment practices.

The following are the main conclusions from the background and related work:

1. There is no single reference point where measures of software sustainability are gathered and exemplified.
2. Design is key to achieve software sustainability, thus the need to show how software designers can incorporate sustainability during software design to improve ICT energy usage and CO_2 emission.
3. The need for a framework or model to assist and guide developers during software design to incorporate sustainability requirements.

These conclusions are the reasons for initiating the creation of a SSDC that can be used by researchers and developers to create new frameworks, tools, guidelines and practices for software design and development. An example of such framework is the proposed pilot framework in section five of this article as guide for both experienced and infant software designers during software design and development.

One last clarification that needs to be made here is the fact that the concepts of sustainable and green are often used interchangeably, in many communities including software engineering. This article considers that "green software" and "sustainable software" is not the same. Green is usually defined as "products, systems and services that have limited negative impact on human health and environment".

As defined in the article at hand, sustainability includes green and it goes beyond green. It is represented by five pillars for environmental, social, economic, human and technical sustainability.

3. The Foundations of Sustainability by Design: The Karlskrona Manifesto

The Karlskrona Manifesto for Sustainability Design (KMSD) has its roots in the Third International Workshop on Requirements Engineering for Sustainable Systems (RE4SuSy) [56], held at RE'14 in Karlskrona, Sweden. Christoph Becker's paper [57] about the relationship between the concerns of sustainability and longevity provided one of the motives for the creation of the manifesto.

The key goal was to blend the diverse aspects of sustainability to clarify its scope, objectives and challenges of the perceptions of sustainability leading to an interdisciplinary platform for researching sustainability [57]. The manifesto brings together input from researchers of various disciplines in the field of software engineering with sustainability research interests as the creators of the design manifesto [23,58].

The Karlskrona Manifesto for Sustainability Design includes nine principles of sustainability design [23]. Those principles provide the basis for creating a reference point that can be applied during software design by different stakeholders (Table 1). The manifesto is accessible via the web [58], where those interested in supporting the manifesto can sign it.

Table 1. Description of the Karlskrona manifesto principles, adapted from [23].

Principle Number	Principle	Description
P1	Sustainability is systemic	Sustainability is never an isolated property. It requires transdisciplinary common ground of sustainability as well as a global picture of sustainability within other properties.
P2	Sustainability has multiple dimensions.	We have to include those dimensions into our analysis if we are to understand the nature of sustainability in any given situation.
P3	Sustainability transcends multiple disciplines.	Working in sustainability means working with people from across many disciplines, addressing the challenges from multiple perspectives.
P4	Sustainability is a concern independent of the purpose of the system.	Sustainability has to be considered even if the primary focus of the system under design is not sustainability.
P5	Sustainability applies to both a system and its wider contexts.	There are at least two spheres to consider in system design: the sustainability of the system itself and how it affects the sustainability of the wider system of which it will be part.
P6	System visibility is a necessary precondition and enabler for sustainability design.	Strive to make the status of the system and its context visible at different levels of abstraction and perspectives to enable participation and informed responsible choice.
P7	Sustainability requires action on multiple levels.	Seek interventions that have the most leverage on a system and consider the opportunity costs: whenever you are taking action towards sustainability, consider whether this is the most effective way of intervening in comparison to alternative actions (leverage points).
P8	Sustainability requires meeting the needs of future generations without compromising the prosperity of the current generation	Innovation in sustainability can play out as decoupling present and future needs. By moving away from the language of conflict and the trade-off mindset, we can identify and enact choices that benefit both present and future.
P9	Sustainability requires long-term thinking.	Multiple timescales, including longer-term indicators in assessment and decisions, should be considered.

The Karlskrona manifesto principles aim to be a practical guide to the entire community like the Agile manifesto [59], the Business Rules manifesto [60], the Service-oriented architecture (SOA) manifesto [61,62], and the Recomputation manifesto [63]. It supports stakeholders in industry and academia (companies, standardization organization, software practitioners, researchers and students) for promoting and developing sustainability design and practices in software development [23,57]. The Karlskrona manifesto also serves as a facilitator for thinking about the broad effects of software on society and the need to embody longer-term thinking, ethical responsibility, and an understanding of how to integrate sustainability into the design of software systems [24].

Table 2 shows how these Karlskrona principles can be related to software development phases [64]. Relating these principles to the software development phases will provide an avenue for using these principles especially for different software systems.

Table 2. Karlskrona manifesto principles in relation to software development life-cycle (SDLC) phases.

SDLC Phases	Karlskrona Manifesto Principles
Phase 1. Project Definition	**P1-** This ensures that the project initiation considers sustainability in the overall project definition from the beginning. **P2-** Software sustainability has different dimensions that have to be considered from the beginning for better project management with different stakeholders. **P3-** Software project usually involves stakeholders from different domains, incorporating their sustainability concerns provides better management of those concerns from multiple perspectives which can help the incorporation of sustainability for the software.
Phase 2. User Requirements Definition	**P2-** It is important to take note of user requirements in relation to each of the sustainability dimensions in order to have better sustainability analysis during the analysis and design phase
Phase 3. System Requirements Definition	**P4-** During elicitation of system requirements, requirement engineers should consider sustainability concerns for the system during the requirements definition even when it is not a core part of the user requirements. **P5-** Cross evaluate the consequential impacts of the system sustainability requirements and the environment in which the system will function.
Phase 4. Analysis and Design	**P2-** Applying this principle provides a blueprint for system evaluation from all sustainability dimensions (economic, environment, social, individual and technical). **P4-** At this phase, this principle helps to encourage analysis of system design based on sustainability in order to facilitate better sustainable system. **P6-** Application of this principle enables better visual and visible overview of the system from different levels of abstraction. **P8-** This will provide better understanding during analysis to make better choices that will help the potential users of the system in present and in future when the system evolves.
Phase 5. Development	**P2-** This will encourage developers during this phase to consider different sustainability dimensions, especially technical, social and individual dimensions. **P4-** Encourage the search for better avenues to make the system sustainable from the development perspective (developers) and also the functions of the system to aid longevity.
Phase 6. Integration and Testing	**P2-** Provides integration and for test team to have a sustainability template that can be used to test the system for all sustainability dimensions based on the sustainability requirement output from phases 2, 3 and 4. **P4-** Application of this principle will aid consideration of sustainability in this phase even if the primary focus of system is not about sustainability.
Phase 7. Implementation	**P5-** Provides beforehand reasoning for the development team to consider the sustainability of the system, its production environment and when pushing it live for use. **P7-** Based on principle 5 (P5), this principle will aid consideration of seeking the involvement of different stakeholders to make the actualization of the system sustainability possible in the production environment and when pushed live.
Phase 8. Sustainment/Maintenance	**P9-** This principle at this stage help to create the conscious awareness so that when the system is in a live environment, there will be continuous evaluation to assess the system sustainability and think of ways for optimizing and improving the sustainability of the system from the different dimensions.

Table 2 highlights some avenues for putting the Karlskrona manifesto principles into practices. Relating these principles to the software development phases will provide an avenue for better understanding of how these principles relates to software development. However, the Karlskrona manifesto focused on high-level principles, not techniques [24], which means there is a need to exemplify the principles to show their practical usage with techniques. The following are the limitations of the manifesto that motivate the development of the SSDC:

1. The principles are abstract and generic to serve all the possible stakeholders interested in sustainability in all the stages of the software development and management phases.

2. The principles are at a high level of abstraction, missing many details for their practical usage.
3. The principles are closely related, making a trade-off among them difficult, especially for a novice in the field of sustainability.
4. The principles are not connected to tangible measures but serve as a guide to create measures.

4. Structure of the Proposed Software Sustainability Design Catalogue (SSDC)

The SSDC serves as a tool that can facilitate the integration of sustainability into design practices as well as lead to a better understanding of sustainability by practitioners. The Software Sustainability Design Catalogue (SSDC) is a set of criteria derived from the nine Karlskrona manifesto principles based on cross analysis of different systems. For each criteria, indicators of sustainability are also derived. The structure of the software sustainability design catalogue is detailed in Figure 1.

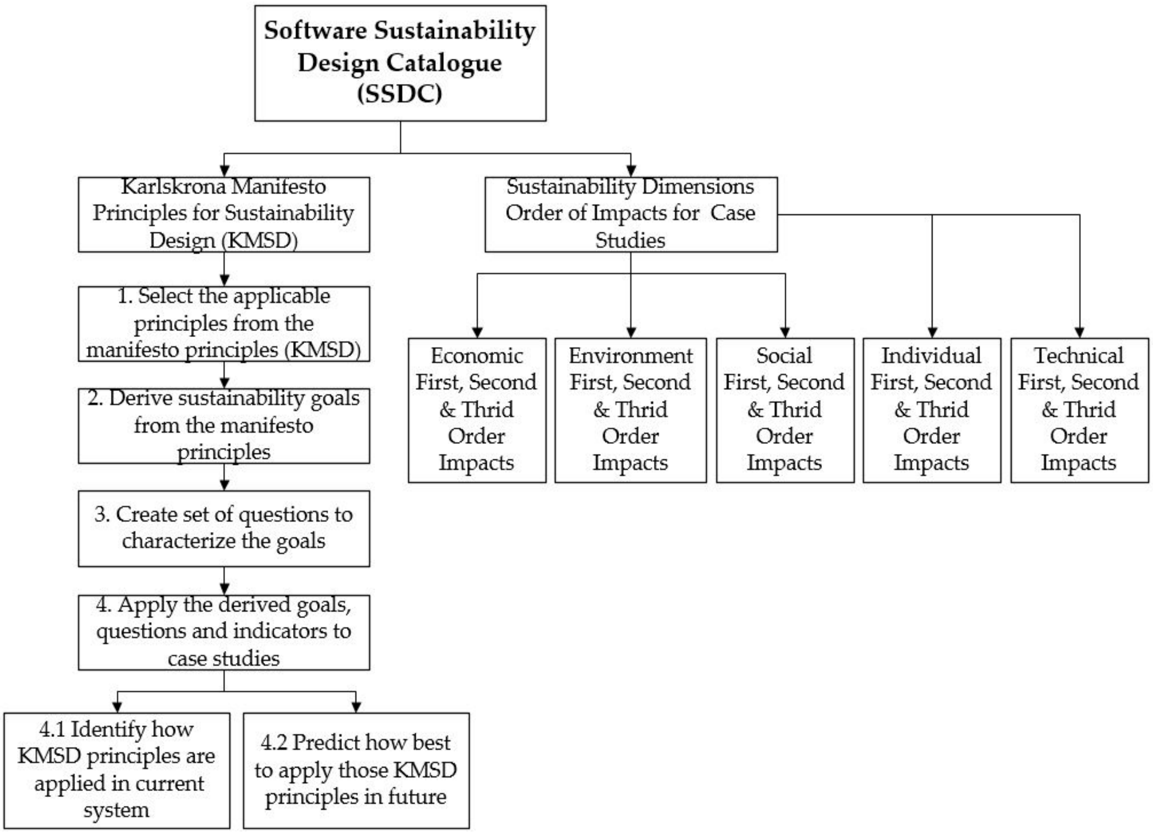
Software Sustainability Design Catalogue (SSDC)
Karlskrona Manifesto Principles for Sustainability Design (KMSD)
1. Select the applicable principles from the manifesto principles (KMSD)
2. Derive sustainability goals from the manifesto principles
3. Create set of questions to characterize the goals
4. Apply the derived goals, questions and indicators to case studies
4.1 Identify how KMSD principles are applied in current system
4.2 Predict how best to apply those KMSD principles in future
Sustainability Dimensions Order of Impacts for Case Studies
Economic First, Second & Thrid Order Impacts
Environment First, Second & Thrid Order Impacts
Social First, Second & Thrid Order Impacts
Individual First, Second & Thrid Order Impacts
Technical First, Second & Thrid Order Impacts

Figure 1. Flow of the derivation of Software Sustainability Design Catalogue (SSDC).

The SSDC distinguishes mainly two components. The first is the sustainability analysis of systems, which is mainly a set of criteria for evaluating the sustainability of software systems. Each criterion is characterized by the following core elements used for evaluating software systems:

1. One or more Karlskrona principle (the 9 principles in Table 1) is used in the evaluation of each system category. Not all principles necessarily can be applied to all systems. The principles are identified using a tag of **P1** to **P9** (Principle 1 to 9).
2. Goal/requirement: this highlights the desired end result for each system category based on sustainability consideration.
3. Stakeholders: those responsible for implementing the goals/requirement.
4. Questions characterizing each goal. From each goal, a set of questions are derived that will determine if each goal is being met.

5. Indicators are used to answer the questions as a way to evaluate if the goals were achieved.
6. Current principle usage in software: this covers the current application of the principle in existing system design and development, even if it is not explicitly stated in current system documentation.
7. Future principle usage in software: based on the evaluation of the current principle application in existing system design and development, a potential usage of the principle in future system enhancement and design is suggested.

The second component are the indicators of sustainability associated with each criteria. These indicators are related to the sustainability dimensions and their order of impacts.

The orders of impact [65,66], cover all the positive and negative effects of software on the environment which are decomposed into three orders of magnitude. The first order impacts (Immediate effects) are about the direct effects of the development and use of software system. The second order impacts (enabling effects) are about the indirect impacts related to the effects of using the software system in its application domain. The third order impacts (structural effects) are the cumulative long-term effects resulting from accumulating first and second order impacts over time.

The sustainability dimensions include [24]:

- The **individual dimension** covers individual freedom and agency (the ability to act in an environment), human dignity, and fulfilment. It includes individuals' ability to thrive, exercise their rights, and develop freely.
- The **social dimension** covers relationships between individuals and groups. For example, it covers the structures of mutual trust and communication in a social system and the balance between conflicting interests.
- The **economic dimension** covers financial aspects and business value. It includes capital growth and liquidity, investment questions, and financial operations.
- The **technical dimension** covers the ability to maintain and evolve artificial systems (such as software) over time. It refers to maintenance and evolution, resilience, and the ease of system transitions.
- The **environmental dimension** covers the use and stewardship of natural resources. It includes questions ranging from immediate waste production and energy consumption to the balance of local ecosystems and climate change concerns.

5. How the SSDC Have Been Derived and Can Be Used

The SSDC was developed using four case studies (see Table 3). Data were gathered for analysing the four different case studies using also the Karlskrona manifesto principles and the orders of impacts. The second and third authors then cross-validated the data collected. Based on the aggregated data, a first draft of catalogue was developed. Then, the proposed software sustainability design catalogue was refined using other types of systems.

Table 3. System categories and types used in case studies.

System Category	System Type
Cyber physical system	Smart home system
Embedded system	Washing machine
Gaming	Angry bird
Desktop application	Microsoft office

The types of systems used in the case studies are summarized in eight tables, Tables 4 and 5 (Cyber physical systems—Smart Home) and Tables A1–A6 detailed in the Appendix cover the following types of systems:

1. Embedded systems that are composed of electrical and mechanical components completely encapsulated by the device they control. The sample case study used in this category is a "Washing Machine" [67–72] (see Tables A1 and A2 in Appendix).
2. Mobile games as an application design that runs on mobile devices. The game case study used in this category is "Angry Bird Game" [73–77] (see Tables A3 and A4 in Appendix).
3. Desktop applications that run on standalone computers. The sample application used in this category is "Microsoft Office" [78–81] (see Tables A5 and A6 in Appendix).

To illustrate how the SSDC works and what the guidelines look like, Tables 4 and 5 present the guidelines for cyber physical system (CPS). CPS are defined here as the integrations of computation, networking, and physical processes that are tightly connected with its users. The sample system used in the catalogue is "Smart Home". The presented guidelines can assist companies and software developers identify key areas that relate to sustainability and recognize strategic avenues on how current and future smart home solutions should be designed in a more sustainable manner. This enables them to make good sustainability decisions during and after the design of smart home solutions.

Table 4 highlights one important issue that standardization authorities in this domain can work on, which is the cross-platform compatibility for smart home devices. Smart home appliances should be compatible with other devices from different manufacturers based on standards to avoid increase in energy usage. Smart home solutions should provide meaningful graphical information that can educate users, thereby encouraging users to behave more sustainably.

Table 5 of the SSDC for cyber physical system (smart home) provides different insights on the direct, indirect and structural impact of home automation design and deployment from the different dimension of sustainability. From Table 3, companies and stakeholders will be able to incorporate the following sustainability goals for the design and development of home automation solutions:

1. **Environment**: reduce household energy consumption.
2. **Economic**: reduce household cost on energy.
3. **Individual**: provide user friendly solution for home users with easy to use user interface and information to induce sustainable behaviour among users.
4. **Technical**: provide good security for user personal data and avoid technical glitch that could lead high energy usage.
5. **Social**: encourage users to form communities to share data as a way of encouraging each other to be energy conscious and environmentally aware of the consequences of their actions and inaction while using smart home solutions.

The application of these principles from the catalogue offers explicit goals and opportunities for sustainability integration in system design through multiple perspectives for systems with sustainability as their core goal and those system without sustainability as their main goal. The below are detailed descriptions of the principles used in providing information on how best to engineer and think of sustainability for smart home solution (see Tables 4 and 5) from the catalogue.

Table 4. Sustainability analysis of cyber physical system (smart home) based on Karlskrona principles [82–85].

Karlskrona Principle and Goal	Principle Usage	Stakeholders	Question	Indicator
(P2) Cross platform compatibility	**Current**: Smart home appliances are compatible with only few other manufacturers' devices in the market. **Future**: Smart home appliances should be compatible with other devices from different manufacturers based on standard interface to avoid increases in energy usage. This can be achieved by enforcing device standards that can be used for cross-platform compatibility. Home automation appliances should be economic and at same time environmental friendly.	Business analyst	Can device function with other device from different manufacturer?	Device cross-platform compatibility
(P4) Educate Users	**Current:** Smart home solutions provides graphical information about energy usage but not necessarily educate users on how to be energy conscious based on their daily habit over a period of time. **Future**: User data from smart home solution should be used for educating users thereby encouraging users to behave more sustainably by presenting energy usage in an informative and educative manner (for example, relate energy usage to the amount of killed trees). User data could also be used for prediction aimed at optimizing the use resources such as water and energy within a household or company. The solutions could help interconnect other systems that can help save resources like water and electricity in a household or company.	Software developers Business analyst	Are users aware of their actions relating to electrical appliances in the house or company?	Usage data over time to detect changes in user habits
(P9) Reduce production and solution cost	**Current**: There are currently few cost-effective solutions that will encourage users to adopt home automation solutions in the long term. **Future:** Use cheap, environmentally friendly resources in the production of home automation device (hardware) that can reduce production cost. If the cost of production reduces, the overall cost of smart home solution will also reduce which will increase its affordability among users. There can also be low-cost solutions for poor countries to assist in the use of water and energy judiciously (reduce waste).	Business Analyst	Did we manage to reduce costs compared to previous years (before solution was a smart home)?	Net cost of smart home solution

Table 5. Sustainability dimensions order of impacts for cyber physical system (smart home) [82,86–89].

Order of Impacts	Environment	Economic	Technical	Social	Individual
1st	Increase in the use of natural resources in the production of hardware for smart home devices and pollution of the environment from toxic material used in production.	Creates new business opportunities for those in this sector (setup and installation of devices at home).	Pave way for improving existing technologies and development of new tools to meet new market demands for sustainable usage of these technologies.	Breeds new communities of users and suppliers.	Users rely on devices to control some aspects of their lives at home and in offices.
2nd	Reduce household energy consumption.	Reduce household bill for energy consumption.	High demand for security of user personal data (privacy).	Increase comfort, safety, flexibility, and security of user.	Demand for sustainable user-friendly solution for home users.
3rd	Increase in the use of toxic material for production of hardware. Less energy consumption over a long period of time.	Decrease cost of energy through optimized solution over time.	Efficient provision of sound technical solutions to avoid technical glitch that could lead to high energy usage. Encourage innovation on how to create cost-effective technologies and devices to reduce household and company energy usage	Encourage users to form communities to share data as a way of encouraging each other to be energy conscious	Induces sustainable behaviour among users.

Sustainability has multiple dimensions (P2): the application of this principle provides an overview of the fundamental issues and positive opportunities that could encourage stakeholders in the smart home domain to cross reference in the design and development of smart home solutions, especially during solution requirements from users and choosing appropriate boundaries.

Smart home design and deployment in this domain requires getting inputs for the effect of design solutions on the environment from natural resources used in building hardware devices, energy consumption of the devices, social behaviour and interaction between people in a family (household), company and other places where these solutions will be deployed. This means all sustainability dimensions (environment, economic, social, individual, technical) will be analysed for better design output.

Sustainability is a concern independent of the purpose of the system (P4): the goal of most smart home solutions is to provide comfort and reduce energy consumption for its users, but it is important to consider an encompassing view of sustainability. This is to be able to get even more benefits such as reducing pollution through the use of environmental friendly materials in producing hardware devices used for smart home solutions. The smart home solution can be used to educate and inform users about the negative consequential effect of their behaviour and habits. This can help induce sustainable behaviour among users. For a smart home solution design to be effective and meet user needs, it will require the expertise of a psychologist or at least an adequately educated interaction designer to help provide information according to the level of comfort and technical expertise of those in manufacturing, transportation, electrical, business and ICT discipline.

Sustainability requires long-term thinking (P9): it is important to think of how the smart home solution provided today will evolve to meet the requirements of current users and be adaptive enough to satisfy future user needs. This will require looking at measures to capture user behaviours over time through computational intelligence to predict future actions of users through data generated from time to time.

Based on the SSDC, a pilot framework to guide stakeholders involved in the design and development of a software system is proposed. Figure 2 provides a detailed flow of the pilot framework.

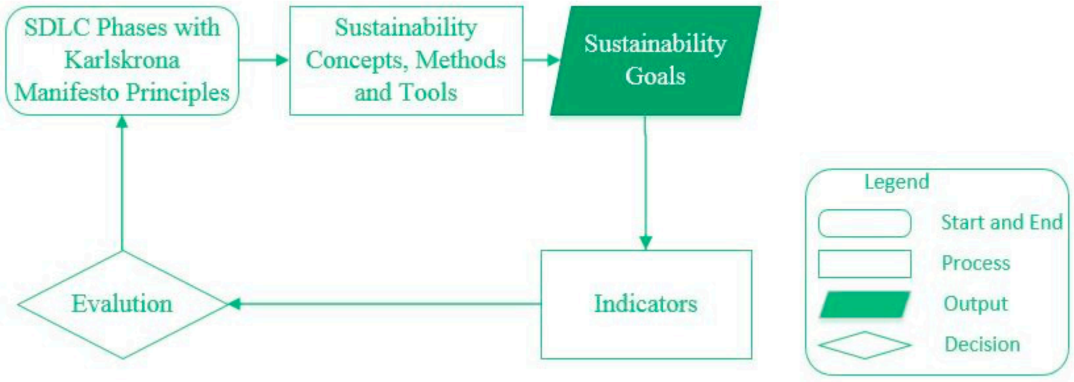
SDLC Phases with Karlskrona Manifesto Principles
Sustainability Concepts, Methods and Tools
Sustainability Goals
Indicators
Evalution
Legend
Start and End
Process
Output
Decision

Figure 2. Pilot framework for sustainability of software system design based on SSDC.

The pilot framework is the first derivative from the SSDC to assist developers incorporate sustainability during system design and development covering the software development life-cycle (SDLC) phases. For a better understanding, the pilot framework is presented below in a tabular form to show contents that are involved in the framework. Table 6 contains all contents of the framework. It is important to highlight that the indicators used in the framework (Table 6) are influenced by the nine Karlskrona manifesto principles mapped to each of the software development life-cycle phases (see Table 2) and the work of Kem-Laurin Kramer [90].

Table 6. Contents of pilot framework for sustainability of software system design.

SDLC Phases and Karlskrona Manifesto Principles	Sustainability Goals	Sustainability Concepts, Methods and Tools	Indicators
Phase 1. Project Definition, P1, P2 and P3	Transmaterialization, design for sustainable efficiency, reusability.	Cradle to cradle, biomimicry, sustainable business canvas.	Carbon footprint, material footprint, end of life footprint.
Phase 2. User Requirements Definition, P2	Increase sustainability awareness among users.	Helix of sustainability.	Total number of sustainability requirements, priority assign to sustainability requirements.
Phase 3. System Requirements Definition, P4, and P5	Design for efficiency, sustainability awareness and interoperability.	Biomimicry, cradle to cradle, goal model.	Total number of system goals relating to sustainability dimensions.
Phase 4. Analysis and Design, P2, P4, P6 and P8	Design for reuse and efficiency, localization, interoperability	Biomimicry, helix of sustainability, Life-cycle sustainability assessment, social return on investment, sustainability analysis radar chart.	Number of first-, second- and third-order impacts of system identified.
Phase 5. Development, P2 and P4	Design for reuse, design for module replicability, design for efficiency, design for sustainability awareness, design for efficiency, design for easy service and maintenance.	Biomimicry, cradle to cradle.	Number of coding choices influenced by sustainability, number of features (functions) added to systems to inform users about sustainability through functions like eco feedback.
Phase 6. Integration and Testing, P2 and P4	Design for easy assembly and disassembly, design for durability,	Cradle to cradle, sustainability analysis radar chart, life-cycle sustainability assessment.	How much information from sustainability analysis chart was used during integration and testing such as the number of systems functions tested against sustainability concerns such as the first-order (immediate) impact and possible second-order (enabling) impacts of the system.
Phase 7. Implementation, P5 and P7	Design for easy use, design to induce conscious sustainability awareness, design to educate users about sustainability, design for easy recycle.	Biomimicry, cradle to cradle.	The priority assign to sustainability by developers and the system owners/users during after implementation
Phase 8. Sustainment/Maintenance, P9	Proper design for serviceability, design for easy replacement of code modules, design for continuous user engagement through sustainability awareness.	Life-cycle sustainability assessment, sustainability analysis radar chart, cradle to cradle.	Number of improvements to system based on sustainability requirements either from users' feedback or developers.

6. Application of the SSDC and the Pilot Framework

In order to exemplify the application of SSDC and the pilot framework, an excerpt from a cyber physical system (smart home) is used here. We consider the following scenario:

> "*A software engineer called Henry has the task of eliciting and documenting the requirements for a new smart home system. Being aware of his responsibility for the software system sustainability he creates, and its impacts, he takes the template of the sustainability analysis of the five dimensions and the three orders of effects from the design catalogue with him to the customer during their first meeting. The customer is curious about these additional analysis ideas, and Henry explains to his client what they mean and gives his client a couple of examples. Then, together with the customer, he fills out the template applying the concepts from the design catalogue (SSDC) to find out what those dimensions and orders of impact mean for the smart home system the customer wants for his house. The information from the activity goes into the requirements analysis that is subsequently conducted and used as a measurement yardstick during the smart home system development and deployment*."

To showcase the use of the framework in the above scenario, the following explanation breakdown how the pilot framework for software sustainability design was used in creating the smart home system from the planning to requirement phase and finally delivery of system.

Phase 1 (project definition) with Karlskrona principles 1, 2 and 3: Henry uses the sustainable business canvas to show value that can be generated through sustainability consideration and how it can help improve the product. Henry was able to pinpoint two sustainability goals from this phase, which is design for sustainable efficiency and to create sustainability awareness through the smart home system by facilitating a community of users willing to share their energy usage to motivate each other.

Phase 2 (user requirements definition) with Karlskrona principle 2: from the information gathered in phase one and a discussion with the client, Henry was able to identify the goal of increasing the sustainability awareness among users of the system once it is created based on the sustainability helix concept. These were the indicators from this phase: percentage of reduced energy usage of the household, amount of feedback on the environmental impact of energy used (CO_2) by the family through eco feedback, number of suggestions provided on how to improve household energy usage based on usage patterns.

Phase 3 (system requirements definition) with Karlskrona principles 4 and 5: the goal in the phase of system requirements is to design for efficiency and sustainability awareness based on the overall system goal from phase 1. He uses the goal model to showcase how the system goals were broken into smaller piece based on the system requirements in order to identify requirement conflicts that might occur. Some of the smaller goals based on the overall goal in this phase include: reduce energy consumption, reduce CO_2 emissions, establish community of users sharing energy usage data, ensure high availability of system, and provide eco-feedback.

Phase 4 (analysis and design) with Karlskrona principles 2, 4, 6 and 8: in this phase, the main goals are design for easy usage, efficiency and sustainability awareness. Using the sustainability analysis diagram according to [24], Figure 3 portrays the sustainability analysis of the smart home system design for the first, second and third (immediate, enabling, and structural) impacts of smart home solutions from the different sustainability dimensions. Information from the analysis provides avenue for evaluating while guiding different stakeholders (managers, developers and users) on the benefits to aspire for sustainability in smart home solutions.

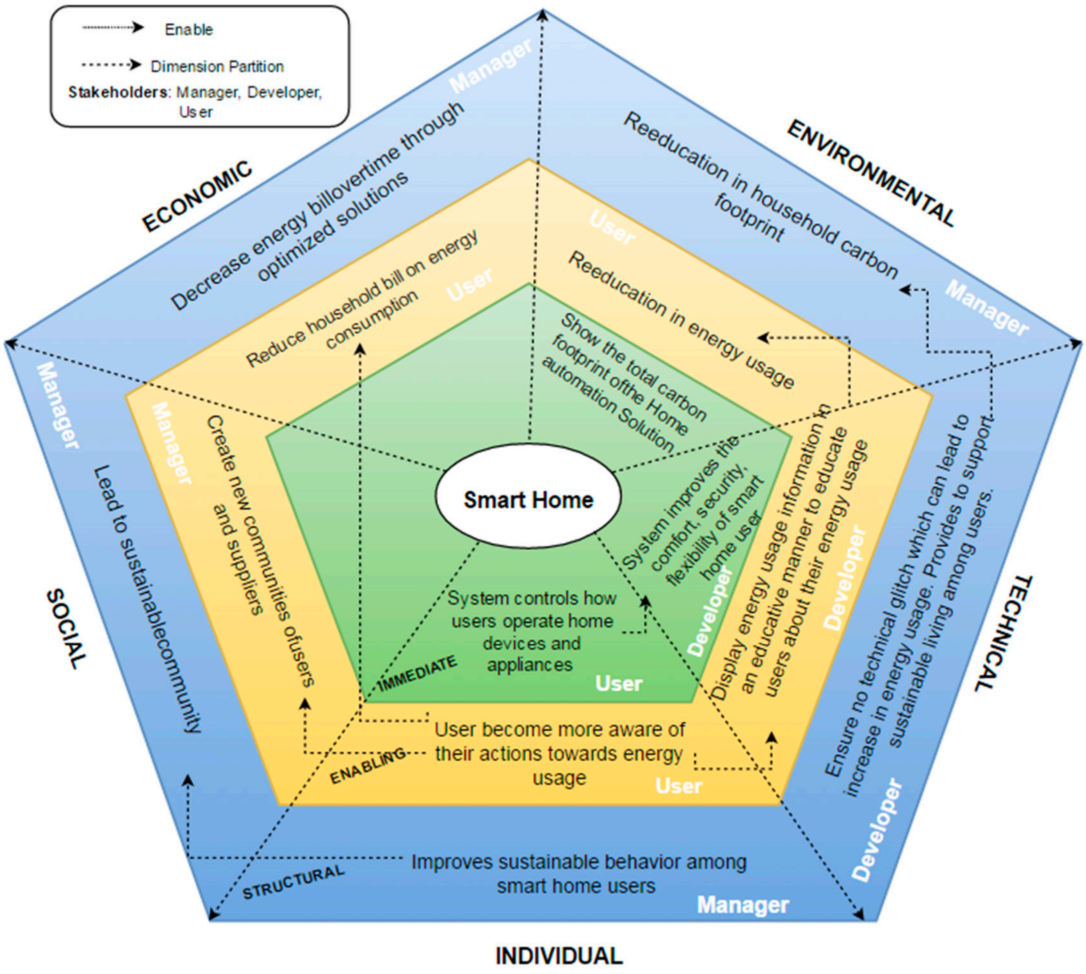
Enable
Dimension Partition
Stakeholders: Manager, Developer, User
ECONOMIC
ENVIRONMENTAL
TECHNICAL
INDIVIDUAL
SOCIAL
Manager
Decrease energy billovertime through optimized solutions
Reeducation in household carbon footprint
User
Reduce household bill on energy consumption
Reeducation in energy usage
Show the total carbon footprint ofthe Home automation Solution
Smart Home
System improves the comfort, security, flexibility of smart home user
Display energy usage information to an educative manner to educate users about their energy usage
Developer
Ensure no technical glitch which can lead to increase in energy usage. Provides to support sustainable living among users.
System controls how users operate home devices and appliances
IMMEDIATE
ENABLING
STRUCTURAL
User become more aware of their actions towards energy usage
Improves sustainable behavior among smart home users
Create new communities ofusers and suppliers
Lead to sustainablecommunity

Figure 3. Immediate, enabling and structural effects of smart home solution in sustainability dimension.

The impacts described in Figure 3 are then taken into account during the requirements refinement phase in such a way that they can be implemented. The indicators from this phase are the immediate, enabling and structural impacts of the system identified in the sustainability analysis diagram.

Phase 5 (development) using Karlskrona principles 2 and 4). The goal of this phase is to implement a smart home system that induces sustainability awareness among users. The sustainability sub-attribute that influences developers during this stage is biomimicry. This concept encourage developers to rethink how to create functionality of the system that can reduce energy consumption, while providing eco-feedback that improves users' sustainability awareness and as an enabler for reducing household energy consumption. The biomimicry is made visible using an energy user interface (UI) dashboard for the smart home system where a fully grown tree is used to mimic energy consumption. As the energy consumption increases, the tree leaf starts to change colour to brown (indicating that is dead), when energy consumption decreases, the tree becomes greener (indicating the tree is back to life).

The following are functions added to the system which are used as indicators in this phase: percentage of energy saved during system usage, the accuracy in the eco-feedback based on the energy usage pattern of users, total number of times users shared their energy usage percentage with friends on social media, the level of user comprehension and understanding of the displayed information on CO_2 emission and indicator for amount of saved trees based on reduce energy usage over time.

Phase 6 (integration and testing) with Karlskrona principles 2 and 4: at this stage, the goal is to assembly and disassembly using the sustainability analysis chart (see Figure 3) including the indicators from the other phases as guide for integrating and testing the smart home system.

Phase 7 (implementation) with Karlskrona principles 5 and 7: the goals of this phase are design for easy use, induce conscious sustainability awareness and educate users about sustainability, which is influenced by the sustainability concept of biomimicry. These are used as indicators: the change in developers coding practice based on the effectiveness of functions added due to sustainability requirements such as percentage of energy saved during system usage, accuracy in the eco-feedback based on the energy usage pattern of users, understandability of presented information on CO_2 emission as well as amount of saved trees based on reduced energy usage over time.

Phase 8 (sustainment/maintenance) with Karlskrona principle 9): this phase covers the long-term goal of the smart home system such as serviceability and continuous user engagement. The indicators used in this phase are the efficiency of data generated in optimizing the smart home system, the effectiveness of the eco-feedback to improve user behaviour, as well as the total percentage of energy saved over time and the backlog management index (BMI).

One important question here is what the effects of the principles detailed in the SSDC for the smart home solution (see Tables 4 and 5) mean for a process engineer. For improving the software development process, the principles considered relevant for the example at hand were principles 2 (multi-dimension), 4 (independent of purpose), and 9 (long-term thinking). Principles 2, 4 and 9 are further explained here as a way to show how the Karlskrona manifesto principles influence decisions made during the development of the smart home example.

Principle 2 from the SSDC highlights the need for cross platform compatibility, which should be considered during the project definition of the software development life cycle (SDLC). It is also relevant during the user requirements elicitation phase when collecting their perceptions of sustainability.

Principle 4 emphasizes the need to educate smart home users about how their actions and reactions affect the environment when using smart home devices. Even if the users are not interested in sustainability, as software developers, designers and engineers it is their responsibility to inform users about the benefits of sustainability during the requirement gathering [91]. Efforts to educate users are also addressed during the documentation processes. The developer needs to create user documentation that include information about energy usage, as well as ways of saving other natural resources such as waste paper when printing.

Principle 9, which is about reducing production and solution costs, plays a vital role. It is important to identify choices that benefit both the current and future users, as well as how the solution can be cost beneficial when encouraging a wider population of users. The business analyst takes charge of this before moving on to the user requirements stage and this issue is monitored throughout the whole project development.

To illustrate how these principles are used, we consider the following second scenario:

> *A company named Energy Life, based on the SSDC analysis, provides a game like menu to control smart home devices for a family named Miralles. The family only wants to reduce household energy cost. Mark, the deployment manager, after reading the SSDC for smart homes (see Tables 4 and 5), pilot framework (see Figure 2), Table 6 (framework description) and the sustainability analysis for smart home solution (see Figure 3), realizes that the best way to reduce the energy cost of the Miralles is to implement a game-like menu for the Miralles to control their smart home devices. This provides information about the energy consumption of each home device. Within a few days the Miralles were able to see the flow of their energy usage and how their daily habits impact the unnecessarily high energy use within their household with impacts on the environment. The game also provides the family with tips on how they can save energy and the amount of* CO_2 *emissions. Members of the family are able to identify the amount of energy consumed by their washing machine. They decided not to run a half empty washing machine again as they can see that this happens almost every day,*

> *and even with the half-load mode it would make more sense to run a full one every two days instead. The Miralles also started a new habit of reminding each other to switch off the computer, TV and other household appliances when they are not in use.*
>
> *Later on, when the company wants to update their system, developers revisit those orders of effect and the different dimensions. Developers also thought about what may and may not have changed in order to improve their system to be more efficient and effectively with regards to sustainability.*

Using the definition of a smart home proposed by Nicholl et al. [92] "dwellings that use integrated communication systems to monitor and manage the performance of the home, and to support the lifestyle choices of the occupants", this scenario illustrates the following features to support sustainability in a smart home solution:

1. Automatic analysis of users' data to educate them and induce sustainable behaviour among users.
2. Alerting users through notification when electrical devices or appliances are running without being used.
3. Automatic scheduling of task such as washing cloths and dishes when energy rate is low during the day.
4. Planning when to turn on/off heating and lighting based on season and user behaviour (prediction).

7. Discussion

The SSDC provides a comprehensive overview of sustainability design considerations and requirements for systems and applications in different domains. SSDC guidelines are based on the analysis of information around the impact of different kinds of system on its application for sustainability dimensions-environment, society, individual, economy, technical and an overview of potential long-term consequences as seen in Tables 4 and 5, Tables A1 and A6 in the Appendix.

Table 4 for sustainability analysis for a cyber physical system (smart home) indicates areas where sustainability improvement can applied in smart home system design such as cross-platform compatibility and design for user awareness about sustainability. Indicators to evaluate these changes are also provided for stakeholders in this domain. Table 5, which presents all the sustainability dimensions order of impacts for cyber physical system (smart home), gives insight into the holistic overview of sustainability effects for smart home systems design and development. Tables A1–A6 in the Appendix includes other system types in the SSDC with information for how to better design those systems and evaluate their sustainability impacts. The SSDC as a catalogue that can inspire the development of tools and framework as shown in this article encourages the development of the pilot framework.

The pilot framework for sustainability of software system design exemplifies the use of the SSDC. The example for smart home in Section 6 shows how each of the development phases mapped with the Karlskrona manifesto and the sustainability goal inspired by different sustainability concepts such as biomimicry, sustainability helix provides better understanding of how sustainability can be centre of software design and development. The indicators from each phase of the SDLC while applying the pilot framework provide a way to evaluate the process and derivatives from each of the SDLC phase influenced by different sustainability concepts. The application of sustainability methods and tools used illustratively, such as sustainability business canvass, goal model and sustainability analysis diagram, provides software developers and requirements engineers with a way to structurally elicit and manage sustainability requirements and monitor system impacts (immediate, enabling and structural).

Specifically, in Section 6 we use the template for sustainability analysis by [24], where Figure 3 depicts an instance of such a sustainability analysis diagram for the smart home solution. During the analysis phase of SDLC, it provides a variety of information for different stakeholders for the direct, indirect and structural effects of sustainability in smart home design and deployment. This information can then be used to create or enhance processes, methods and tools that can automate the incorporation of sustainability into the design of smart home solutions.

The software sustainability design catalogue and the underlying pilot framework can be beneficial for the following stakeholders interested in sustainability, its engineering and its integration in/for software systems design and development:

1. For companies and software developers, it serves as guide on how sustainability can be incorporated into software design and development. It can also enable them to identify the effects of their project on technical, economic, social, individual, and environmental sustainability. Furthermore, we support the current revision of the Association for Computing Machinery (ACM) code of ethics and propose to incorporate sustainability principles and explicitly acknowledge the need to consider sustainability as part of professional practice [23].
2. Standardization organizations can benefit from it to create future standards for software and organizational sustainability. SSDC shows areas where software applications can impact the environment and humans, and this information can help create standards that would encourage companies and stakeholders to improve existing and new applications and policies to promote sustainability.
3. Public authorities will be able to use the information from the catalogue to enact new laws persuading industry practitioners to design software systems, applications and devices in a more sustainable manner.
4. Academic institutions can identify avenues to advance research on sustainability by design, sustainability design patterns and tools to support, among others.

8. Conclusions

Effective sustainability engineering and integration requires clarifying the current perceptions of sustainability and defining a concrete framework for its engineering and measurement. As a first milestone, this paper presented a catalogue that quantifies sustainability via a series of guidelines that can be used for incorporating sustainability into the design loop.

By analysing how the principles defined in the Karlskrona Manifesto for Sustainability Design can be applied for some specific systems, we were able to identify a series of guidelines and develop the foundations for a "sustainability by design" approach. First, we reviewed the current perceptions of sustainability for various types of systems. Furthermore, based on how sustainability has been perceived in different software engineering disciplines, the SSDC has been defined. Each guideline is defined as a set of principles, dimensions of sustainability, orders of impact, and indicators. The usage and applicability of the catalogue have been demonstrated for four types of systems.

Future research includes examining other types of systems and the application of the guidelines in an industry setting. This will give better insights for the development of the guidelines for different types of systems and its usage by diverse stakeholders in the software development life cycle. An important aspect of its validity is that the catalogue was created based on the expertise of the wide set of researchers involved in the sustainability design manifesto. Considering the fact that sustainability in software engineering is still evolving, the SSDC provides common ground for further research.

An important limitation of the SDLC is its validity in industry. Consequently, the theoretical validation of the methodological aspects underlying the proposed guidelines will be considered beyond the industry evaluation to be conducted in future. The Appraisal of Guidelines for Research and Evaluation (AGREE) instrument [93], which is the appropriate tool that assesses the methodological rigor and transparency in which the guidelines is being developed, will used for this purpose.

The SSDC also has automation potential in the future. The design catalogue can become the basis for a recommender system. This would help developers to identify and apply effectively the sustainability guidelines. However, this requires more case studies for building a knowledge base required by a recommender system. The automation will provide a practical guide to enable developers during each stage of design and development to understand and incorporate the Karlskrona manifesto principles, sustainability goals, concepts, tools and methods with indicators that can help in the evaluation of a software system.

Finally, this paper provides a foundation (via the SSDC and pilot framework) for the software engineering community to design and engineer sustainability into their systems.

Author Contributions: S.O. conceived the idea of the paper; S.O. and B.P. designed the research methodology; investigation and data collection: S.O.; data validation: B.P. and A.S.; S.O. wrote the paper and all authors contributed to reviewing all sections; review and editing supervision: B.P. and A.S.

Funding: This research received no external funding.

Acknowledgments: The authors are grateful to the researchers who initiated the Karlskrona manifesto principles which were used in this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Tables A1 and A2 provide information on sustainability analysis of an embedded system (washing machine) using the Karlskrona principles and sustainability dimensions based on usage of order of impacts. Based on output from Tables A1 and A2, water and energy efficiency are key objectives of sustainability for a washing machine.

Tables A3 and A4 covers information about sustainability analysis for a mobile game (Angry Bird) centred on the Karlskrona principles and sustainability dimensions based on order of impacts. Table A3 provides points of energy efficiency, reduction of wear and tear of hardware and creating a sense of belong to community among users as key goals of sustainability for stakeholders. In addition, details from Table A4 prompt the need to aspire for these goals from all sustainability dimensions:

- **Environment**: optimize energy and computing resource consumption during game development and when users are playing game.
- **Economic**: provide continuous innovation on the game features to encourage current users to keep playing the game and attract new users and ensure game is maintainable (longevity).
- **Technical**: ensure that game does not encourage quick hardware wear and tear and at same time has the ability to evolve with new demands of the market.
- **Individual**: good user experience while interacting with the game and serves as a medium of inducing sustainable behaviour.
- **Social**: create good community sense among angry bird users and educate them about sustainability.

Tables A5 and A6 presents information on sustainability analysis of desktop application (Microsoft Office) using the Karlskrona principles and sustainability dimensions based on order of impacts.

Table A1. Sustainability analysis of embedded system (washing machine) based on Karlskrona principles.

Karlskrona Principles and Goal	Current Principle Usage	Future Principle Usage	Stakeholders	Question	Indicator
(P 9) Efficient water usage	Current washing machine has some design features to help reduce water wastage during washing circle.	Good mechanism within the washing machine to aid efficient use of water during washing cycle and also display the amount of water saved to users. This will serve as means of educating users about water wastage. This will aid positive impact on the amount of water usage in a household.	Software developer	Does the washing machine reduce water usage?	Washing cycle/total amount of water used.
(P 6, 8) Energy Efficiency	Some current sets of washing machine have eco-friendly features to reduce energy usage.	One good feature to reduce energy usage of washing is to turn off or hibernate automatically after washing cycle if idle for 2 min. It will help reduce energy cost (P 6) and also reduce resource usage in the long term (P 8). This will help reduce energy consumption when machine is idle. Incorporate the use of scheduler to the washing machine as a way to time when the washing cycle should start during the period of the day when energy cost is less.	Software developer	Does the machine use too much energy for a single washing cycle?	Energy efficiency (washing cycle/total amount of energy used).
(P 8, 9) Water efficiency	Allows collection of grey water from washing machine.	Encourage reuse of greywater (with biodegradable laundry detergent) in garden watering.	Business analyst	How much of the greywater gets reused afterwards?	Percentage (%) of reused water.

Table A2. Sustainability dimensions order of impacts for embedded system (washing machine).

Order of Impacts	Environment	Economic	Technical	Social	Individual
1st	Increase in the use of natural resources in the production of washing machine components such as iron, copper, medium-density fibreboard, ethylene propylene diene monomer (EPDM) rubber.	High demand for natural resources boosts the economies of countries with those natural resources.	Increase demand for new technologies, tools and equipment for extracting raw materials.	New job opportunities for people.	Increase the risk of having skin diseases due to toxic material exposure.
2nd	Increase in water and energy usage when using a washing machine as opposed to manual washing.	High energy cost for household as a result of increase in energy usage due to ease of washing cloths (convenience factor and little manual labour).	Demand for energy and water saving mechanism in washing machine.	More job opportunities for technicians with knowledge of washing machine technologies.	Increase the ease of washing for users.
3rd	Over a long time it lead to increase water usage and a culture of washing a lot (e.g., California). In turn, that leads to a higher wear and tear of the hardware.	Increase in profit for washing machine production companies with high demand for washing machine from users.	Over time there will be pressure to build a more energy and water efficient washing machine from manufacturers to have a competitive edge over other competitors.	Increase job creation over time both from industry for skilled workers and in household for technicians to fix minor issues of washing machine.	Over a long time it leads to increased water usage and a culture of washing a lot (e.g., California). In turn, that leads to a higher wear and tear of the hardware

Table A3. Sustainability analysis of mobile games (Angry Bird) based on Karlskrona principles.

Karlskrona Principles and Goal	Current Principle Usage	Future Principle Usage	Stakeholders	Question	Indicator
(P 1) Energy Efficient	Currently game development focus more on usability and fun factor aspect than sustainability aspect.	Create the mobile game architecture with sustainability consideration. Since sustainability is systemic, it should be core of the application structure. Consider energy efficiency during game development. Incorporate green patterns to game application development.	Software developer	Is the mobile energy Efficient?	(Energy efficiency) useful-work done/energy used.
(P 1) Reduce wear and tear of hardware	Although sustainability is not the core of current game development practices, although game developer tries to ensure optimal use of hardware resources to make game run faster and better on hardware (phone, computer, tablets).	Ensure that the mobile game during operation uses hardware resources (memory, CPU etc.) in efficient and sustainable manner to reduce wear and tear.	Software developer	What is the impact of the game on hardware components like CPU and RAM?	Does the game use too much hardware resources?
(P 2) Sense of belonging to a community and connectedness to other people	Current game development provide sort of community for it users in form of forums and groups online.	There is a digital community surrounding most games and the question is how it compares to face to face communities for gamers. Create a community that make gamers feel connected to both the digital and real worlds. Make gamers feel like they are part of something (people always want to belong to a tribe).	Business analysts, software developer	Is there a sense of community amongst players?	Connectedness in community? Number of 'friends' in a particular gaming community?
(P9) Good user experience	Current game development incorporates user experience into their production to gain more user base and profit.	Game should use reasonable lighting effect for the game display which can help reduce energy usage and also incorporate sustainability concept in the total overall design of the game. It can also add features that will educate users about sustainability.	User experience (UX) engineer and software developer	Can users complete their task easily?	Gateway metrics.

Table A4. Sustainability dimensions order of impacts for mobile games (Angry Bird).

Order of Impacts	Environment	Economic	Technical	Social	Individual
1st	Increase in energy usage from computers and mobile devices used for developing and testing game application.	Cost of production for game development companies.	Increase in demand for sophisticated hardware and software for game development.	Open job opportunities for game developers.	Provides avenues for leisure activities for users.
2nd	Increase in energy usage because users are using mobile phone, laptops, iPad to play game. There will also be need for charging of these devices coupled with the energy consumption when playing the game.	Company make profit from game purchase. Increase demand and user cost for hardware (computers, phones and tablets).	Demand for better graphics and quick game response from users.	Create community sense among Angry Bird users.	Demand for good user experience while interacting with the game.
3rd	Over time leads to hardware wear and tear because of continuous game time from user side on computer/phones and continuous game development from game production company side.	Increase in profit from huge user and fan base for the game company.	Demand for newer features and innovation from customers	Increase the sense of belonging in form of community among users.	Lead to game addiction and lack of social interaction with outside world.

Table A5. Sustainability analysis of desktop application (Microsoft Office) based on Karlskrona principles.

Karlskrona Principles and Goal	Current Principle Usage	Future Principle Usage	Stakeholders	Question	Indicator
(P1) Incorporate sustainability into development process.	Current development framework allow the use of scrum and version control.	Provide a development framework that support sustainability focusing on the software and those developing the software itself.	(Development) process engineer	Are guidelines available?	Boolean (Yes or No).
(P8) Offer reasonable amount of features.	Currently all office application comes with all the features which sometimes are rarely used by users.	Provide all basic features for office application and allow users to add other features when needed.	Business analyst and software developer	What is the number of changes to be made to add new features?	Rework metric.
(P9) Add green print (and let user know how many pages they save over time).	Users can print any document as their need requires.	Incorporate green print to office application that inform users whenever they want to print a document for the second time that they can skip few pages because changes were not made on those pages.	Software developer	Do people print less by using this green print button?	Number of pages printed compared to if there was no green button.

Table A6. Sustainability dimensions order of impacts for desktop application (Microsoft Office).

Order of Impacts	Environment	Economic	Technical	Social	Individual
1st	Reduce the amount of manual writing on papers which in turn reduces the amount of paper used by people for writing.	Open new potential market for company to explore.	Ensure the ability of Word application to meet new market demands from the technical implementation aspect.	Creates new job opportunities for those that are expert with most of the office applications.	Provide efficient way of doing daily task such as documentation, project management and design.
2nd	Reduce paper resource wastage.	Increase in profit for Microsoft and other partner companies through sales of office application.	Increase in use of computers for development of add-ons and plugins for office applications.	Provides a community feeling among users.	Guarantee that user will easily finish their task while using the application for their day-to-day work.
3rd	It will increase energy usage over time due to ease of doing daily task on computers, phones and tablets.	Rise in company's profit through product innovation as demands change from generation to generation.	Demand for new features to meet new demands.	Improves connectedness among users through collaboration online.	Improves the overall ability of users completing their task (documentation, project management and design).

References

1. Microsoft. Microsoft 2015 Citizenship Report Letter from Our CEO. 2015. Available online: http://download.microsoft.com/download/7/3/6/736CED21-9D8B-4CBB-98E8-DCBAE7026251/Microsoft%202015%20Citizenship%20Report.pdf (accessed on 20 March 2018).
2. IBM. Global CEO Study: The Enterprise of the Future. Available online: https://www-935.ibm.com/services/uk/gbs/pdf/ibm_ceo_study_2008.pdf (accessed on 11 October 2017).
3. Nidumolu, R.; Prahalad, C.K.; Rangaswami, M.R. Why sustainability is now the key driver of innovation. *IEEE Eng. Manag. Rev.* **2013**, *41*, 30–37. [CrossRef]
4. Firstcarbon Solutions. Constructing Sustainable Business Models. Available online: http://www.firstcarbonsolutions.com/resources/newsletters/january-2014-constructing-sustainable-business-models/constructing-sustainable-business-models/ (accessed on 18 June 2018).
5. Knut, H.; Martin, R.; Ingridvon, S.; Michael, A.; David, K.; Nina, K. Sustainability Nears a Tipping Point Sustainability Nears a Tipping Point. *MIT Sloan Manag. Rev.* **2012**, *53*, 69–74.
6. Calero, C.; Piattini, M. Introduction to Green in software engineering. *Green Softw. Eng.* **2015**, 3–27. [CrossRef]
7. Becker, C.; Betz, S.; Chitchyan, R.; Duboc, L.; Easterbrook, S.M.; Penzenstadler, B.; Ssyff, N.; Venters, C.C. Requirements: The key to sustainability. *IEEE Softw.* **2016**, *33*, 56–65. [CrossRef]
8. Koçak, S.A.; Alptekin, G.I.; Bener, A.B. Evaluation of software product quality attributes and environmental attributes using ANP decision framework. *CEUR Workshop Proc.* **2014**, *1216*, 37–44.
9. Ericsson. Technology for Good. Available online: https://www.ericsson.com/assets/local/about-ericsson/sustainability-and-corporate-responsibility/documents/2015-corporate-responsibility-and-sustainability-report.pdf (accessed on 30 November 2014).
10. Ericsson. Energy and Carbon Report. 2013. Available online: http://www.ericsson.com/res/docs/2013/ericsson-energy-and-carbon-report.pdf (accessed on 11 November 2017).
11. Digiconomist. Bitcoin Energy Consumption Index. Available online: https://digiconomist.net/bitcoin-energy-consumption (accessed on 18 June 2018).
12. Elbahrawy, A.; Alessandretti, L.; Kandler, A.; Pastor-satorras, R. Evolutionary dynamics of the cryptocurrency market. *R. Soc. Open Sci.* **2017**, 1–16. [CrossRef] [PubMed]
13. Chris, P.; Dan, S.; Eli, B. Understanding and Mitigating the Effects of Device and Cloud Service Design Decisions on the Environmental Footprint of Digital Infrastructure. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI'16), San Jose, CA, USA, 7–12 May 2016; ACM/IEEE: New York, NY, USA, 2016; pp. 1324–1337.
14. Al Hinai, M.; Chitchyan, R. Engineering Requirements for Social Sustainability. 2016. Available online: http://www.cs.le.ac.uk/people/rc256/ict4s2016_hinai_chitchyan.pdf (accessed on 20 June 2018).
15. Penzenstadler, B. Software Engineering for Sustainability. Available online: https://www.ics.uci.edu/~djr/DebraJRichardson/SE4S.html (accessed on 20 June 2018).
16. Blevis, E.; Preist, C.; Schien, D.; Ho, P. Further Connecting Sustainable Interaction Design with Sustainable Digital Infrastructure Design. In Proceedings of the 2017 Workshop on Computing Within Limits (LIMITS'17), Santa Barbara, CA, USA, 22–24 June 2017; ACM/IEEE: New York, NY, USA, 2017; pp. 71–83.
17. Penzenstadler, B.; Femmer, H. A Generic Model for Sustainability with Process-and Product-Specific Instances. In Proceedings of the 2013 workshop on Green in/by software engineering (GIBSE'13), Fukuoka, Japan, 26 March 2013; ACM: New York, NY, USA, 2013; pp. 3–7.
18. Jeanette Schwarz, E.B.; Beloff, B. Use Sustainability Metrics to Guide Decision Making. *CEP* **2002**, *98*, 58–63.
19. Penzenstadler, B. What does Sustainability mean in and for Software Engineering? In Proceedings of the 1st International Conference on ICT for Sustainability (ICT4S), Zurich, Switzerland, 14–16 February 2013.
20. Venters, C.C.; Jay, C.; Lau, L.M.S.; Griffiths, M.K.; Holmes, V.; Ward, R.R.; Austin, J.; Dibsdale, C.E.; Xu, J. Software sustainability: The modern tower of babel. In Proceedings of the Third International Workshop on Requirements Engineering for Sustainable Systems Co-Located with 22nd International Conference on Requirements Engineering (RE 2014), Karlskrona, Sweden, 26 August 2014.
21. Ehrenfeld, J.R. *Sustainability by Design: A Subversive Strategy for Transforming Our Consumer Culture*; Yale University Press: New Haven, Ct, USA; London, UK, 2008.
22. Freeman, P. The Central Role of Design in Software Engineering: Implications for Research. *Softw. Eng. Res. Dir.* **1980**, 121–132. [CrossRef]

23. Becker, C.; Chitchyan, R.; Duboc, L.; Easterbrook, S.; Mahaux, M.; Penzenstadler, B.; Rodriguez-Navas, G.; Salinesi, C.; Seyff, N.; Venters, C.; et al. The Karlskrona manifesto for sustainability design. *Softw. Eng.* **2014**, *20*, 2014.
24. Becker, C.; Chitchyan, R.; Duboc, L.; Easterbrook, S.; Penzenstadler, B.; Seyff, N.; Venters, C.C. Sustainability Design and Software: The Karlskrona Manifesto. In Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, Italy, 16–24 May 2015.
25. Saval, G.; Mahaux, M.; Heymans, P. Discovering Sustainability Requirements: An Experience Report. In Proceedings of the 20th International Working Conference on Requirements Engineering: Foundation for Software Quality, Essen, Germany, 7–10 April 2014.
26. Chitchyan, R.; Betz, S.; Duboc, L.; Penzenstadler, B.; Ponsard, C.; Venters, C.C. Evidencing Sustainability Design Through Examples. 2015. Available online: http://eprints.hud.ac.uk/id/eprint/25699/1/Session2Paper3.pdf (accessed on 28 June 2018).
27. Groher, I.; Weinreich, R. An Interview Study on Sustainability Concerns in Software Development Projects. In Proceedings of the 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Vienna, Austria, 30 August–1 September 2017.
28. Pargman, D.; Eriksson, E.; Höjer, M.; Östling, U.G.; Borges, L.A. The (Un)sustainability of Imagined Future Information Societies. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17), Denver, CO, USA, 6–11 May 2017; ACM: New York, NY, USA, 2017; pp. 773–785.
29. Ilstedt, S.; Eriksson, E.; Hesselgren, M.I.A. Sustainable Lifestyles–How Values Affect Sustainable Practises. 2017. Available online: http://www.nordes.org/nordes2017/assets/full_papers/nordes17a-sub1039-cam-i26_ILSTEDT_v2.pdf (accessed on 28 June 2018).
30. Shedroff, N. Design is the Problem: The Future of Design Must be Sustainable. 2009, p. 582. Available online: https://designethosandaction.files.wordpress.com/2015/01/design-is-the-problem.pdf (accessed on 28 June 2018).
31. Bibri, M. Sustaining ICT for Sustainability. 2009. Available online: https://www.diva-portal.org/smash/get/diva2:833352/FULLTEXT01.pdf (accessed on 28 June 2018).
32. Eriksson, E.; Pargman, D.; Bates, O.; Normark, M.; Gulliksen, J.; Anneroth, M.; Berndtsson, J. HCI and UN's Sustainable Development Goals: Responsibilities, Barriers and Opportunities. In Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI'16), Gothenburg, Sweden, 23–27 October 2016; ACM: New York, NY, USA, 2016; pp. 140:1–140:2.
33. Vister, K.K.; Evans, R.D. Identifying Contributing Factors to Sustainability Awareness in the Norwegian Software Industry. In Proceedings of the 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 22 December 2017.
34. Musthaler, L. Energy-Aware Software Design Can Reduce Energy Consumption by 30% to 90%. Network World. 2014. Available online: https://www.networkworld.com/article/2861005/green-it/energy-aware-software-design-can-reduce-energy-consumption-by-30-to-90.html (accessed on 28 June 2018).
35. Durdik, Z.; Klatt, B.; Koziolek, H.; Krogmann, K.; Stammel, J.; Weiss, R. Sustainability Guidelines for Long-Living Software Systems. In Proceedings of the 2012 28th IEEE International Conference on Software Maintenance (ICSM), Trento, Italy, 23–28 September 2012; pp. 517–526.
36. Malik, M.N.; Khan, H.H. Investigating Software Standards: A Lens of Sustainability for Software Crowdsourcing. *IEEE Access* **2018**, *6*, 5139–5150. [CrossRef]
37. Nina, W.; Patricia, L.; Francesco, O. Sustainability in Software Engineering. 2017. Available online: http://dl.ifip.org/db/conf/ifip6-3/sustainit2017/08_P08_S21_SustainIT2017.pdf (accessed on 28 June 2018).
38. Robillard, M.P. Sustainable Software Design. In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Seattle, WA, USA, 13–18 November 2016; ACM: New York, NY, USA; pp. 920–923.
39. Lago, P.; Koçak, S.A.; Crnkovic, I.; Penzenstadler, B. Framing Sustainability as a Property of Software Quality. *Commun. ACM* **2012**, *55*, 56–64. [CrossRef]
40. Chitchyan, R.; Duboc, L.; Becker, C.; Betz, S.; Penzenstadler, B.; Venters, C.C. Sustainability Design in Requirements Engineering: State of Practice. In Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C), Austin, TX, USA, 14–22 May 2016.

41. Calienes, G.G. Requirements Prioritization Framework for developing Green and Sustainable Software Using ANP—Based Decision Making. 2013. Available online: http://enviroinfo.eu/sites/default/files/pdfs/vol7995/0327.pdf (accessed on 28 June 2018).
42. Penzenstadler, B. RE4ES: Support Environmental Sustainability by Requirements Engineering. 2012. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.382.9172&rep=rep1&type=pdf (accessed on 28 June 2018).
43. Bozzelli, P.; Gu, Q.; Lago, P. A Systematic Literature Review on Green Software Metrics. 2013. Available online: http://dare.ubvu.vu.nl/bitstream/handle/1871/52417/SLR?sequence=1 (accessed on 28 June 2018).
44. Kern, E.; Dick, M.; Naumann, S.; Guldner, A.; Johann, T. Green Software and Green Software Engineering—Definitions, Measurements, and Quality Aspects. In Proceedings of the First International Conference on Information and Communication for Sustainability, Zurich, France, 14–16 February 2013; pp. 87–94.
45. Johann, T.; Dick, M.; Kern, E.; Naumann, S. Sustainable Development, Sustainable Software, and Sustainable Software Engineering: An Integrated Approach. In Proceedings of the 2011 International Symposium on Humanities, Science and Engineering Research, Kuala Lumpur, Malaysia, 6–7 June 2011; pp. 34–39.
46. Sawyer, P.; Paech, B.; Heymans, P. (Eds.) Requirements Engineering: Foundation for Software Quality. In Proceedings of the 13th International Working Conference (REFSQ 2007), Trondheim, Norway, 11–12 June 2007; pp. 247–261.
47. Dick, M.; Naumann, S. Enhancing Software Engineering Processes Towards Sustainable Software Product Design. 2010. Available online: https://pdfs.semanticscholar.org/98d0/4de0318b252f273e3cf36cc385253542b924.pdf (accessed on 28 June 2018).
48. Erdélyi, K. Special Factors of Development of Green Software Supporting Eco Sustainability. In Proceedings of the 2013 IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 26–28 September 2013.
49. Oyedeji, S.; Seffah, A.; Penzenstadler, B. Sustainability Quantification in Requirements Informing Design. 2017. Available online: http://ceur-ws.org/Vol-1944/paper6.pdf (accessed on 28 June 2018).
50. Penzenstadler, B. Infusing Green: Requirements Engineering for Green in and through Software Systems. 2014. Available online: http://ceur-ws.org/Vol-1216/paper8.pdf (accessed on 28 June 2018).
51. Roher, K.; Richardson, D. Sustainability Requirement Patterns. In Proceedings of the 2013 3rd International Workshop on Requirements Patterns (RePa), Rio de Janeiro, Brazil, 15–19 July 2013.
52. Colmant, M.; Rouvoy, R.; Seinturier, L. Improving the Energy Efficiency of Software Systems for Multi-Core Architectures. In Proceedings of the 11th Middleware Doctoral Symposium, Bordeaux, France, 8–9 December 2014.
53. Lami, G.; Buglione, L. Measuring Software Sustainability from a Process-Centric Perspective. In Proceedings of the 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, Assisi, Italy, 17–19 October 2012.
54. Calero, C.; Bertoa, M.F.; Moraga, M.A. A Systematic Literature Review for Software Sustainability Measures. In Proceedings of the 2013 2nd International Workshop on Green and Sustainable Software (GREENS), San Francisco, CA, USA, 20 May 2013.
55. Seacord, R.; Elm, J.; Goethert, W.; Lewis, G.A.; Plakosh, D.; Robert, J.; Wrage, L.; Lindvall, M. Measuring Software Sustainability. In Proceedings of the International Conference on Software Maintenance, Amsterdam, The Netherlands, 22–26 September 2003.
56. Penzenstadler, B.; Martin, M.; Camille, S. RE4SuSy: Requirements Engineering for Sustainable Systems. *J. Syst. Softw.* **2013**, *995*. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.415.8601&rep=rep1&type=pdf (accessed on 28 June 2018).
57. Christoph, B. Sustainability and longevity: Two sides of the same quality? *CEUR Workshop Proc.* **2014**, *1216*, 1–6.
58. Becker, C. Website for The Karlskrona Manifesto for Sustainability Design. 2014. Available online: http://sustainabilitydesign.org/karlskrona-manifesto/ (accessed on 10 October 2017).
59. Fowler, M.; Highsmith, J. The agile manifesto. *Softw. Dev.* **2001**, *9*, 28–35.
60. Bussiness Rules Group. The Business Rules Manifesto. Available online: http://www.businessrulesgroup.org/brmanifesto.htm (accessed on 12 November 2017).

61. Arsanjani, A.; Booch, G.; Boubez, T.; Brown, P.; Chappell, D.; de Vadoss, J.; Loesgen, B. The SOA Manifesto. *SOA Manifesto* **2009**, *35*, 82–88.
62. Felikson, L. The SOA Manifesto: Establishing a Common Understanding about SOA and Service-Orientation History of The SOA Manifesto. 2010. Available online: http://2010.secrus.org/wp-content/uploads/download/Felikson.pdf (accessed on 28 June 2018).
63. Gent, I. The Recomputation Manifesto. Available online: https://www.software.ac.uk/blog/2016-10-05-recomputation-manifesto (accessed on 12 November 2016).
64. Langer, A.M. Guide to Software Development—Designing and Managing the Life Cycle. 2016, p. 402. Available online: https://www.springer.com/la/book/9781447167976 (accessed on 28 June 2018).
65. Berkhout, F.; Hertin, J. Impacts of Information and Communication Technologies on Environmental Sustainability: Speculations and Evidence. Available online: http://www.oecd.org/sti/inno/1897156.pdf (accessed on 28 June 2018).
66. Hilty, L.M.; Aebischer, B. ICT for Sustainability: An Emerging Research Field. In *ICT Innovations for Sustainability*; Advances in Intelligent Systems and Computing; Hilty, L., Aebischer, B., Eds.; Springer: Cham, Switzerland, 2015; Volume 310.
67. ETH Sustainability Summer School 2011. Washing Machine. 2011. Available online: http://webarchiv.ethz.ch/sustainability-v2/lehre/Sommerakademien/so2011/washies_report.pdf (accessed on 10 December 2017).
68. Consumer Reports. Washing Machines That Save Water and Money. 2017. Available online: https://www.consumerreports.org/cro/news/2015/04/washing-machines-that-save-water-and-money/index.htm (accessed on 28 June 2018).
69. National Geographic. Washing Machines Buying Guide. 2017. Available online: http://environment.nationalgeographic.com/environment/green?guide/washing?machine?buying?guide/environmental?impact/ (accessed on 5 May 2017).
70. Jack, T. The dirt on clothes: Why washing less is more sustainable. *J. Home Econ. Inst. Aust.* **2013**, *20*, 44.
71. Ross, C. The Damage I Cause When I Wash My Clothes. 2015. Available online: https://theswatchbook.offsetwarehouse.com/2015/07/16/environmental-impact-of-the-washing-machine/ (accessed on 28 June 2018).
72. Markham, D. How to Reuse Grey Water in the Home and Yard. 2017. Available online: https://www.treehugger.com/green-home/how-reuse-grey-water-home-and-yard.html (accessed on 28 June 2018).
73. R Entertainment. UN Honours 'Angry Birds, Happy Planet' Campaign. 2017. Available online: http://www.goodnewsfinland.com/un-honours-angry-birds-happy-planet-campaign/ (accessed on 28 June 2018).
74. Deloittte, G. 2016 Mobile Industry Impact Report: Sustainable Development Goals. 2016. Available online: https://www.gsma.com/betterfuture/wp-content/uploads/2016/09/_UN_SDG_Report_FULL_R1_WEB_Singles_LOW.pdf (accessed on 28 June 2018).
75. Arndt, H.; Dziubaczyk, B.; Mokosch, M. *Information Technology in Environmental Engineering*; Springer International Publishing: Cham, Switzerland, 2014; pp. 13–24.
76. Sony Pictures. Make the Angry Birds Happy! Take Climate Actio Send a Tweet, Record a Vine, Take a Photo That Shows Raising Your Voice for Good Cause. Available online: http://www.angrybirdshappyplanet.net/ (accessed on 13 October 2017).
77. Sony Pictures. The Angry Birds Movie Themed Angry Birds for a Happy Planet. Available online: http://www.sonypicturesgreenerworld.com/articles/angry-birds-for-a-happy-planet (accessed on 14 October 2017).
78. Accenture. Microsoft, Accenture and WSP Environment & Energy Study Shows Significant Energy and Carbon Emissions Reduction Potential from Cloud Computing. Available online: https://newsroom.accenture.com/article_display.cfm?article_id=5089 (accessed on 20 November 2017).
79. Microsoft. Microsoft Products Our Products are Empowering People and Organizations to Achieve More While Improving Efficiency and Reducing Carbon Emissions. Cloud Services. Available online: https://www.microsoft.com/about/csr/environment/solutions/cloud/ (accessed on 11 November 2017).
80. Ahmad, I.; Ranka, S. (Eds.) *Handbook of Energy-Aware and Green Computing–Two Volume Set*; Chapman and Hall/CRC: New York, NY, USA, 2012.
81. Accenture. Cloud Computing and Sustainability: The Environmental Benefits of Moving to the Cloud. Available online: http//download.microsoft.com/download/A/F/F/AFFEB671-FA27-45CF-9373-0655247751CF/CloudComput.Sustain.-Whitepaper-Nov2010.pdf (accessed on 22 August 2017).

82. Queensland Department of Public Works. Smart and Sustainable Homes Design Objectives. 2016. Available online: http://www.hpw.qld.gov.au/construction/sustainability/smartsustainablehomes/Pages/Default.aspx (accessed on 28 June 2018).
83. Wilson, C.; Hargreaves, T.; Hauxwell-Baldwin, R. Benefits and risks of smart home technologies. *Energy Policy* **2017**, *103*, 72–83. [CrossRef]
84. Bhati, A.; Hansen, M.; Chan, C.M. Energy conservation through smart homes in a smart city: A lesson for Singapore households. *Energy Policy* **2017**, *104*, 230–239. [CrossRef]
85. Monaghan, A. Seven Things You Need to Know about the UK Economy. Available online: https://www.theguardian.com/sustainable-business/2015/apr/17/things-need-know-sustainable-smart-technology (accessed on 18 December 2017).
86. Solanki, V.K.; Muthusamy, V.; Katiyar, S. Think Home: A Smart Home as Digital Ecosystem. *Circuits Syst.* **2016**, *7*. [CrossRef]
87. Li, R.Y.M. The Usage of Automation System in Smart Home to Provide a Sus-Tainable Indoor Environment: A Content Analysis in Web 1.0. *Int. J. Smart Home* **2013**, *7*, 47–60.
88. Barker, S.; Mishra, A.; Irwin, D.; Cecchet, E.; Shenoy, P.; Albrecht, J. Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Home. 2012. Available online: http://www.ecs.umass.edu/~deirwin/sustkdd12.pdf (accessed on 28 June 2018).
89. Makonin, S.; Bartram, L.; Popowich, F. A Smarter Smart Home: Case Studies of Ambient Intelligence. *IEEE Pervasive Comput.* **2013**, *12*, 58–66. [CrossRef]
90. Kramer, K.-L. *User Experience in the Age of Sustainability*; Morgan Kaufmann: Burlington, MA, USA, 2012.
91. Spinellis, D. The Social Responsibility of Software Development. *IEEE Softw.* **2017**, *34*, 4–6. [CrossRef]
92. Nicholl, A.; Perry, M. *Smart Home Systems and the Code for Sustainable Homes: A BRE Guide*; IHS BRE Press: Bracknell, UK, 2009; ISBN 9781848061125.
93. Brouwers, M.C.; Kho, M.E.; Browman, G.P.; Burgers, J.S.; Cluzeau, F.; Feder, G.; Fervers, B.; Graham, I.D.; Grimshaw, J.; Hanna, S.; et al. AGREE II: Advancing guideline development, reporting and evaluation in healthcare. *Can. Med. Assoc. J.* **2010**, *182*, E839–E842. [CrossRef] [PubMed]