

Chapter 1

Introduction to Green in Software Engineering

Coral Calero and Mario Piattini

1.1 Introduction

Sustainability is gaining importance worldwide, reinforced by several initiatives with wide media coverage such as the Earth hour¹; this is a worldwide grassroots movement uniting people to protect the planet, organised by the WWF (World Wide Fund for Nature). Other organisations such as the United Nations (UN) also highlight the importance of reducing energy consumption and our carbon footprint, including this issue in the Millennium Development Goals (MDGs²). In Rio+20, the United Nations Conference on Sustainable Development, the world leaders approved an agreement entitled ‘The Future We Want’, where it is stated that ‘We recognize the critical role of technology as well as the importance of promoting innovation, in particular in developing countries. We invite governments, as appropriate, to create enabling frameworks that foster environmentally sound technology, research and development, and innovation, including in support of green economy in the context of sustainable development and poverty eradication...’.

Clean and efficient energy as a societal challenge has also been included by the European Union in Horizon 2020,³ the biggest EU Research and Innovation programme with nearly €80 billion of funding available from 2014 to 2020. Other initiatives related to environmental sustainability can likewise be found in other countries.

¹ <http://www.earthhour.org/>

² <http://www.un.org/millenniumgoals/>

³ <http://ec.europa.eu/programmes/horizon2020/>

C. Calero (✉) • M. Piattini

Department of Information Technologies and Systems, University of Castilla-La Mancha, Ciudad Real, Spain

e-mail: Coral.Calero@uclm.es; Mario.Piattini@uclm.es

Although these initiatives point to ICTs (information and communication technologies) as a key to achieve these goals, we must be aware that ICTs also have a negative impact on the environment. In fact, as noted by [20], when pursuing strategic sustainability, the impact of technology is important from two different points of view at the same time. On the one hand, it helps organisations to tackle environmental issues (using video conferences, dematerialisation, more efficient processes, etc.); on the other hand, technology itself is often responsible for major environmental degradation (amounts of energy consumed by the engineering processes used to manufacture products). This dual aspect of technology means that organisations also face two challenges: they need to have more sustainable processes and they must produce products that contribute to a more sustainable society.

As far as the ICT sector is concerned, it contributes about 2 % of the global CO₂ emissions and is responsible for approximately 8 % of the EU’s electricity use, and some 2 % of its carbon emissions come from the ICT equipment and services and household electronic sector. The total electricity consumption of the ICT sector is forecast to increase by almost 60 % from 2007 to 2020 (see Fig. 1.1) due to the increasing number of devices as well as to network expansion [24].

In [38], the authors estimate that present-day systems for business email, productivity and CRM software in the United States require 268, 98 and 7 petajoules (PJ) of primary energy each year, respectively, when the direct energy use and embodied energy of all system components are considered. When combined, the present-day primary energy footprints of these three business software applications add up to as much as 373 PJ per year.

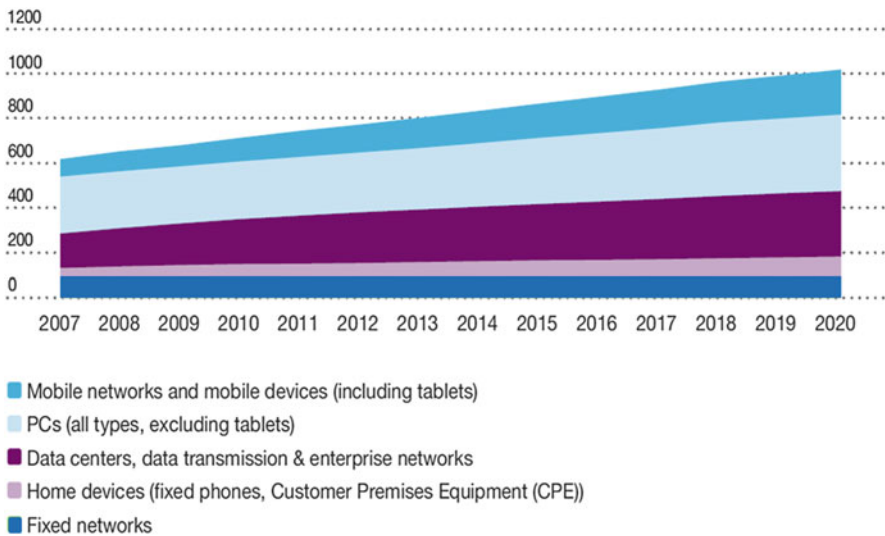


Fig. 1.1 ICT carbon footprint outlook (Mtonnes CO₂e) (from [2])

It is therefore essential to control the use of ICTs in order to reduce their impact on sustainability as much as possible. We will focus specifically on software technology, because software is more complex to sell, service and support than hardware, and dollar for dollar, software generates more downstream economic activity than hardware [31], but it has been disregarded in this area until now. Spending on software is growing faster than spending on IT overall—4.8 % a year between 2008 and 2013, compared to 3.3 % for all IT spending [31].

Sustainability has also become more and more important to business recently. A business that fails to have sustainable development as one of its top priorities could receive considerable public criticism and subsequently lose market legitimacy [20]. According to a global IBM survey in 2008, 47 % of organisations have begun to redesign their business models on the basis of sustainability, treating sustainable development as a new source of innovation, a new opportunity for cutting costs and a new mechanism for gaining competitive advantage. All of this can be summarised under the concept of ‘strategic sustainability’, introduced by [55]. Most people claim that they will pay more for a green product [13]. In early 2010, the ISO 26000 standard [33] for corporate social responsibility (CSR) was published, providing executives with the directions and measures for demonstrating social responsibilities. In this standard, businesses are required to take a precautionary approach to protecting the environment; the aim is to promote greater environmental responsibility through business practices and encourage the adoption of environment-friendly information technologies. CSR involves the voluntary integration by companies of social and environmental concerns in their business operations, as well as in relationships with their partners [26]. Expectations from corporations are higher than ever. Investors and other stakeholders consider companies in terms of the ‘triple bottom line’, reflecting financial performance, environmental practices and corporate social responsibility (CSR). The present-day dominant conception of CSR implies that firms voluntarily integrate social and environmental concerns in their operations and interactions with stakeholders [12]. All the CSR definitions consistently refer to five dimensions: voluntariness, stakeholders, social, environmental and economic [17].

In general, the initiatives that foster respect for the environment by means of ICT, IT, software, etc., are called Green or Greening ICT/IT/Software or sometimes sustainability in IT. The problem that arises is that, as in every new discipline, there is no clear map of concepts and definitions. As [8] points out, however, the fact is that Green IT is not only a trend; it is becoming a necessity as more and more organisations are implementing some form of sustainable solutions. These same authors comment that, according to Forrester Research, it is expected that the Green IT services market will grow from \$500 million to nearly \$5 billion in 2013.

In the next section, we will try to clarify the differences, similarities and relationships between all these concepts.

1.2 Sustainability

The aim of this section is to give a general definition of the word ‘sustainability’ without actually linking it to any particular context. To do so, we will first summarise the main definitions of sustainability.

Sustainability is a widely used term and refers to the capacity of something to last for a long time. Some more precise definitions are as follows:

- The Collins dictionary [16] defines sustainability as ‘the ability to be maintained at a steady level without exhausting natural resources or causing severe ecological damage’.
- A similar definition of ‘sustainable’ can be found in Merriam-Webster: ‘of, relating to, or being a method of harvesting or using a resource so that the resource is not depleted or permanently damaged’ [39].
- According to [9], a sustainable world is broadly defined as ‘one in which humans can survive without jeopardizing the continued survival of future generations of humans in a healthy environment’.
- In [49], the authors affirm that ‘sustainability can be discussed with reference to a concrete system (ecological system, a specific software system, etc.), therefore, global sustainability implies the capacity for endurance given the functioning of all these systems in concert’.
- ‘Sustainability is the capacity to endure and, for humans, the potential for long-term maintenance’ [47].
- From another perspective, sustainability can be viewed as ‘one more central quality attribute in a row with the standard quality attributes of correctness, efficiency, and so forth’ [47]. These same authors also defined the term *sustainable development* as that which ‘includes the aspect to develop a sustainable product, as well as the aspect to develop a product using a sustainable development process’.
- The Brundtland report from the United Nations (UN) defines sustainable development as the ability to ‘meet the needs of the present without compromising the ability of future generations to satisfy their own needs’ [62]. According to the UN, sustainable development needs to satisfy the requirements of three dimensions, which are the society, the economy and the environment.
- In [2], the author identifies the same dimensions as the aforementioned UN report for sustainable development: economic development, social development and environmental protection:
 - ‘Environmental sustainability ensures that the environment is able to replenish itself at a faster rate than it is destroyed by human actions. For instance, the use of recycled material for IT hardware production helps to conserve natural resources.
 - Social development is concerned about creating a sustainable society which includes social justice or reducing poverty. In general all actions that promote social equity and ethical consumerism.

- The economic pillar ensures that our economic growth maintains a healthy balance with our ecosystem; it integrates environmental and social concerns into business’.

Of all the definitions above, the most widely used is that established by the Brundtland report of the United Nations (UN) [62].

If we take a close look at the definitions, we can observe that there are two fundamental pillars underpinning sustainability: ‘The capacity of something to last a long time’ and ‘the resources used’.

Another aspect that is related to sustainability, and that can be found in the literature, has to do with the topic to which it is applied: information systems, ICT, software, etc.

Taking into account that our focus is on software engineering (SE), Fig. 1.2 summarises the different levels of sustainability that relate organization to information systems and to software engineering.

In the following sections, we will present some definitions for each of the levels in Fig. 1.2. We have worked mainly with papers published in the area of software, software engineering and information systems, because that is what this book focuses on. This means that we will not present an exhaustive study on definitions (i.e. on those beyond the scope of this book), but we believe our work will provide a snapshot of how things are interpreted in the software engineering area.

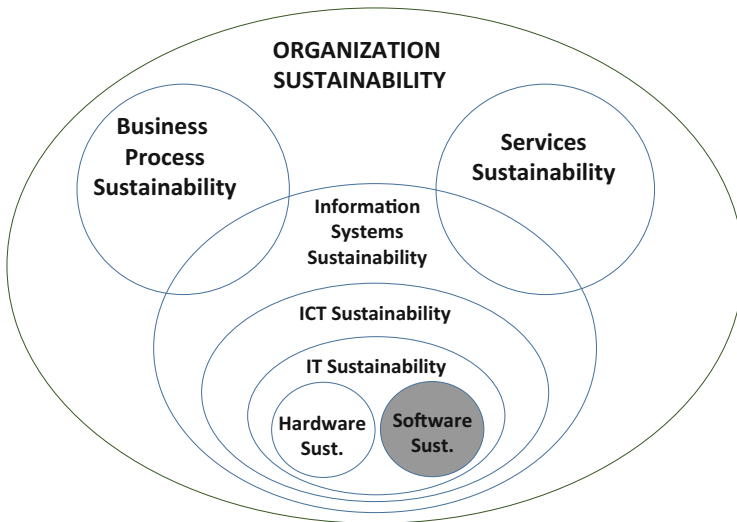


Fig. 1.2 Sustainability levels

1.2.1 *IS Sustainability*

It must be noted that, in the literature, authors do not differentiate between ‘IS sustainability’ and ‘sustainable IS’ (the same applies to the other levels), so we take these concepts as equivalent in this book.

As articulated in the SIGGreen Statement, ‘the Information Systems discipline can have a central role in creating an ecologically sustainable society because of the field’s five decades of experience in designing, building, deploying, evaluating, managing, and studying information systems to resolve complex problems’ [27].

The authors of [64] recommend using the term IS sustainability over IT sustainability, because they consider that the exclusive focus on information technologies is too narrow.

As remarked by [53], it is only through process change and the application of process-centred techniques, such as process analysis, process performance measurement and process improvement, that the transformative power of IS can be fully leveraged in order to create environmentally sustainable organisations and, in turn, an environmentally sustainable society.

Taking this one step further, we contend that IS researchers must consider process-related concepts when theorising about the role of IT in the transformation towards sustainable organisations. This will not only allow us to better understand the transformative power of IS in the context of sustainable development but will also enable us to proceed to more prescriptive, normative research that has a direct impact on the implementation of sustainable, IT-enabled business processes [53].

Although there are some groups working on information systems and environmental friendliness, it is difficult to find suitable IS sustainability concepts. Most of the work being done is about Green IS. In [13], it is considered that the sustainability in IS must take into account aspects such as efficiency systems, forecasting, reporting and awareness, energy-efficient home computing and behaviour modification. Finally, the book focuses on Green Business Process Management consolidating the global state-of-the-art knowledge about how business processes can be managed and improved in the light of sustainability objectives [63].

1.2.2 *ICT/IT Sustainability*

Donnellan et al. [4] remark that sustainable ICT can develop solutions that offer benefits both internally and across the enterprise:

- Aligning all ICT processes and practices with the core principles of sustainability, which are to reduce, reuse and recycle
- Finding innovative ways to use ICT in business processes to deliver sustainability benefits across the enterprise and beyond

The Ericsson report [24] points to dematerialisation and increased efficiency as the two main ways of aligning ICT with sustainability.

Following the definition provided by [61], IT sustainability is a shorthand for ‘global environmental sustainability’, a characteristic of the Earth’s future, in which certain essential processes persist for a period of time comparable with human lives.

1.2.3 *Software Sustainability*

There are several areas in which software sustainability needs to be applied: software systems, software products, Web applications, data centres, etc. Various works are in process, but most of this concerns data centres, which consume significantly higher energy than commercial office space [36].

As noted in [10], the way to achieve sustainable software is principally by improving power consumption. Whereas hardware has been constantly improved so as to be energy efficient, software has not. The software development life cycle and related development tools and methodologies rarely, if ever, consider energy efficiency as an objective [11]. Energy efficiency has never been a key requirement in the development of software-intensive technologies, and so there is a very large potential for improving efficiency [59].

As remarked by [21], software plays a major role, both as part of the problem and as part of the solution. The behaviour of the software has significant influence on whether the energy-saving features built into the platform are effective [56].

In [49], it is said that ‘The term Sustainable Software can be interpreted in two ways: (20) the software code being sustainable, agnostic of purpose, or (24) the software purpose being to support sustainability goals. Therefore, in our context, sustainable software is energy-efficient, minimizes the environmental impact of the processes it supports, and has a positive impact on social and/or economic sustainability. These impacts can occur direct (energy), indirect (mitigated by service) or as rebound effect [30]’.

According to [18], sustainable software is ‘software, whose impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software are minimal and/or which have a positive effect on sustainable development’.

These authors subsequently use the same definition for the concept of green and sustainable software. They therefore define green and sustainable software as ‘software, whose direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software are minimal and/or which has a positive effect on sustainable development’ [46]. They consider that direct impacts are related to resources and energy consumption during the production and use of software, while indirect impacts are effects from the software product usage, together with other processes and long-term systemic effects.

One of the most complete definitions is the one proposed by [34], which considers that green and sustainable software is software whose:

- ‘Direct and indirect consumption of natural resources, which arise out of deployment and utilization, are monitored, continuously measured, evaluated and optimized already in the development process
- Appropriation and utilization aftermath can be continuously evaluated and optimized
- Development and production processes cyclically evaluate and minimize their direct and indirect consumption of natural resources and energy’

Another related term is sustainable computing. It is used to transfer the political concept of sustainability to computer systems, including material components (hardware) as well as informational ones (software); it includes development as well as consumption processes [40].

As commented at the beginning of Sect. 1.2, the literature contains some definitions of sustainable (or sustainability), while others refer to the term *green* (or *greenability*).

This phenomenon is especially noteworthy in the case of software, because various authors such as [46] and [34] use both terms synonymously. We believe that this approach is faulty and that it ought to be avoided, since we are talking about two different concepts, as will be seen in due course.

What does seem true, however, is that software sustainability, although still in its early stages, is a very important research topic that will be of great importance in the next few years. That said, general work on its significance is needed. The goal of that work would be to raise awareness on the part of all those involved with software: the companies that develop software, those who buy it and also the people who use it.

1.2.3.1 Software Engineering Sustainability

One part of the software sustainability is the software engineering sustainability. Within the context of software engineering, not many proposals have tackled the concept of sustainability [50]. In a recent update of this work, the authors observed that the number of proposals has increased considerably over the last 2 years [49]. This fact serves to demonstrate that there is an ever-growing concern to tackle sustainability in the context of software engineering.

Sustainability should generally be taken into account from the very first stages of software development. That is not always feasible, since it is not easy to change how developers work. Moreover, there is little guidance on how software engineering can contribute to improving the sustainability of the systems under development [48]. In this work, the authors consider five dimensions of sustainability that are important for the analysis of software systems:

- **Individual sustainability:** This refers to the maintenance of the private good of individual human capital. Health, education, skills, knowledge, leadership and access to services constitute human capital [52]. For software engineering (SE),

we have to ask: ‘How can software be created and maintained in a way that enables developers to be satisfied with their job over a long period of time?’

- **Social sustainability:** This means maintaining social capital and preserving the solidarity of societal communities. Social capital is investments and services that create the basic framework for society [52]. For SE, we ask: ‘What effects do software systems have on society (e.g. communication, interaction, government)?’
- **Economic sustainability:** This aims to maintain assets. Assets include not only capital but also added value. This requires a definition of income as the “amount one can consume during a period and still be as well off at the end of the period, as it devolves on consuming added value (interest), rather than capital” [52]. For SE, the question is: ‘How can software systems be created so that the stakeholders’ long-term investments are as safe as possible from economic risks?’
- **Environmental sustainability:** This seeks to improve human welfare by protecting natural resources such as water, land, air, minerals and ecosystem services; hence, much is converted to manufactured or economic capital. Environment includes the sources of raw materials used for human needs, as well as ensuring that sink capacities recycling human wastes are not exceeded [39]. For SE, we pose the question: ‘How does software affect the environment during, inter alia, development and maintenance?’
- **Technical sustainability:** From a point of view of (software) systems engineering, there is another dimension that has to be considered. Technical sustainability has the central objective of long-time usage of systems and their adequate evolution with changing surrounding conditions and respective requirements. For SE: How can software be created so that it can easily adapt to future change?

There are many definitions of sustainable software engineering. We present some of these in Table 1.1. It is clear that there are many more works which use the term *sustainable software engineering*.

1.3 From Sustainability to Greenability

As detected in several definitions, sustainability is generally considered from three dimensions (the social, the economic and the environmental) provided by the UN [62].

If we apply the definition to our context, the third dimension, the one related to the technical aspects, is the one that we call the ‘green’ dimension. Figure 1.3 shows this in diagram form.

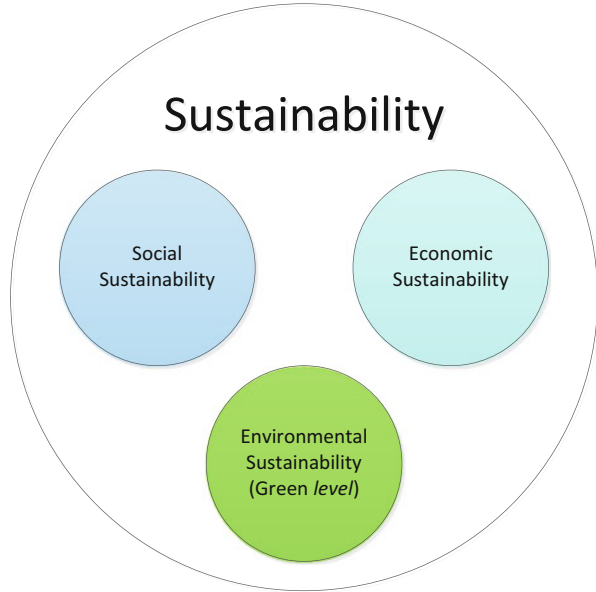
Taking this distinction as a basis, in the next section we will show the definitions of green applied to each one of the levels in Fig. 1.2. As happened in the case of sustainability, in the literature authors use the terms *green* and *greenability*

Table 1.1 Sustainable

Reference	Term	Definition
[3]	Sustainable software engineering	Sustainable software engineering aims to create reliable, long-lasting software that meets the needs of users while reducing environmental impacts; its goal is to create better software so we will not have to compromise future generations' opportunities
[37]	Sustainable software engineering	Sustainable software engineering aims to create reliable, long-lasting software that meets the needs of users while reducing the negative impact on the economy, society and the environment
[33]	Sustainable software engineering	Sustainable software engineering is the art of defining and developing software products in a way so that the negative and positive impacts on sustainability that result and/or are expected to result from the software product over its whole life cycle are continuously assessed, documented and optimised
[58]	Sustainable software engineering	Sustainable software engineering is the development that balances rapid releases and long-term sustainability, whereas sustainability is meant as the ability to react rapidly to any change in the business or technical environment
[19]	Green and sustainable software engineering	Green and sustainable software engineering is the art of developing green and sustainable software with a green and sustainable software engineering process. Therefore, it is the art of defining and developing software products in a way, so that the negative and positive impacts on sustainable development that result and/or are expected to result from the software product over its whole life cycle are continuously assessed, documented and used for a further optimisation of the software product
[35]	Green and sustainable software engineering	The objective of green and sustainable software engineering is the enhancement of software engineering, which targets <ol style="list-style-type: none"> 1. The direct and indirect consumption of natural resources and energy 2. As well as the aftermath that are caused by software systems during their entire life cycle, the goal being to monitor, continuously measure, evaluate and optimise these facts
[31]	Software engineering for sustainability	The aim of software engineering for sustainability (SE4S) is to make use of methods and tools in order to achieve this notion of sustainable software

(e.g. Green IS and IS Greenability) synonymously; we will do the same, presenting definitions found for both concepts.

Fig. 1.3 Sustainability dimensions



1.3.1 *Green IS*

At the top level, we found the Green IS concept. Chen et al. [14] unite the terms *Green IT* and *Green IS* and suggest that ‘Green IS & IT refers to IS & IT products (e.g., software that manages an organization’s overall emissions) and practices (e.g., disposal of IT equipment in an environmentally-friendly way) that aims to achieve pollution prevention, product stewardship, or sustainable development’.

The authors of [64] define Green IS as inclusive of Green IT, extended with people, processes, software and information technologies to support individual, organisational or societal goals (Fig. 1.4).

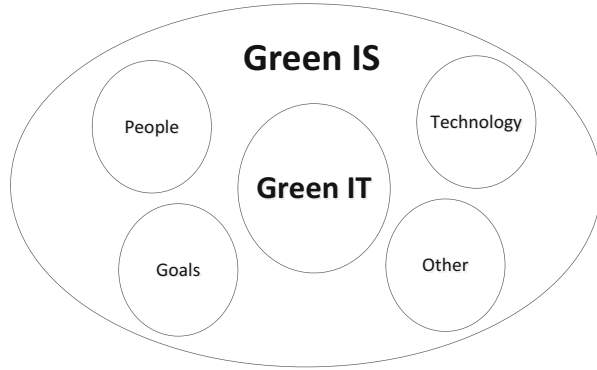
The Green Book [29] focuses on Green IS projects, programmes and initiatives as potential influences on the sustainability of organisations and communities under threat from climate change and other aspects of environmental degradation.

1.3.2 *Green ICT/IT*

The literature provides us with a variety of definitions of the concept of Green IT. The term *Green IT* refers to the relationship between IT and energy efficiency [11].

In [6], the authors state that Green IT means using technology efficiently while taking into account the triple bottom line: ‘economic viability, social responsibility and environmental impact’.

Fig. 1.4 Green IS and Green IT



The same author in [5] presents eco-computing and green computing as synonyms of Green IT, defining them as a set of best practices for the optimal use of computing resources. Green practices in technology can cover several phases of the product or service life cycle: from acquisition to recycling and final disposal.

In [45], the author considers that Green IT ‘refers to the study and practice of designing, manufacturing, and using computer hardware, software, and communication systems efficiently and effectively with no or minimal impact on the environment’. In his opinion, Green IT is also about ‘using IT to support, assist, and leverage other environmental initiatives and to help in creating green awareness’. The author refined the definition in [44] in the following manner: ‘Green IT is the study and practice of designing, manufacturing, and using computers, servers, monitors, printers, storage devices, and networking and communications systems efficiently and effectively with minimal impact on the environment. It includes environmental sustainability, the economics of energy efficiency, and the total cost of ownership, which incorporates the cost of disposal and recycling. Green IT is also about the application of IT to create energy-efficient, environmentally sustainable business processes and practices’.

Very similar definitions are provided in the following pieces of work:

- In [22], ‘the aim of Green IT is to produce as little waste as possible during the whole IT lifecycle (development, operation and disposal)’.
- In [35], ‘Green IT considers the optimizing the resource and energy consumption of ICT itself, induced during the whole life cycle, and tries to optimize it’.
- In [28], Green IT ‘denotes all activities and efforts incorporating ecologically friendly technologies and processes into the entire life cycle of information and communication technology’.
- In [15], the authors use the term *green computing*. This term ‘refers to environmentally sustainable computing which studies and practices virtually all computing efficiently and effectively with little or no impact on the environment’. The green computing term is the same as Green IT [4].
- In [42], ‘Green IT is a systematic application of environmental sustainability criteria to the design, production, sourcing, use and disposal of the IT technical

infrastructure as well as within the human and managerial components of the IT infrastructure in order to reduce IT, business process and supply chain related emissions and waste and improve energy efficiency’.

- In [7], the authors consider Green IT and green computing as synonymous, defined as the study and practice of designing, manufacturing, using and disposing of computers, servers and associated subsystems efficiently and effectively with minimal or no impact on the environment. Green IT thus encompasses hardware assets, software assets, tools, strategies and practices that help improve and foster environmental sustainability.
- In [54], the authors consider that the definition of Green IT is broad, as it can be applied to situations where IT enables greenhouse gas emission reductions and to situations where IT enables structural changes that lead to changes in broader societal patterns, which takes us closer to the low-carbon society and leads to further emission reductions.

A different definition is provided in [1]. In this work, the author believes that Green IT can be described by dividing IT-related issues into four different fields:

- ‘Field 1 concerns the IT product itself and the energy and environmental impact they cause and in particular the products people use on a daily basis. This field is important to gain (sic) credibility to Green IT solutions because it is difficult to take Green IT seriously if the products needed have not been undertaken by the process of diminishing environmental impact.
- Field 2 is about transportation, communication and virtual mobility. This field is pictured as two separate parts, which represents the transportation of goods and the transportation of people.
- Field 3 is about community planning on all levels, ranging from whole regions, cities and small towns down to the personal household planning level.
- Field 4 handles the production and consumption patterns. IT opens possibilities to measure environmental impact on production and consumption and following a product or a service throughout its entire lifecycle enables control over the total environmental effect’.

In [41], the authors explain that Green IT initiatives can range from those that focus on reducing IT infrastructure’s carbon footprint to those that transform a business. Green IT can be deployed to support a variety of sustainability initiatives, such as those to measure carbon footprints, monitor the environmental impact of business practices, reduce waste in business processes, lower resource consumption or increase energy efficiency and reduce greenhouse gas emissions.

From our point of view, one of the definitions that best expresses how the term *Green IT* is tackled in the literature, at the same time as being more thorough and precise, is the definition provided in [48], which encapsulates all the definitions in [15, 22, 28, 35].

For a deeper insight into Green IT, we recommend the book *Harnessing Green IT* [43], in which the idea is to give a holistic perspective on Green IT by discussing its various facets and showing how to embrace them strategically.

As remarked by [22], however, over a long time, the topics of Green IT involved only research dealing with hardware. It is clear that, independently of the efforts made until now, software is also part of IT and must be taken into account when talking about Green IT.

Apart from the previous definitions of general Green IT, there is an important aspect to be taken into account, which is related to the difference between Green *in* IT and Green *by* IT. The next section introduces this difference, along with the definitions found between them.

1.3.2.1 Green by IT Versus Green in IT

The main difference between Green *in* IT and Green *by* IT is the role played by the IT and the focus of the greenness. As indicated by [61], the difference depends on considering IT as a producer to handle the emissions produced by the IT gadgets themselves or considering IT as an enabler to enable reduction of emissions across all areas of an enterprise. This difference was also highlighted recently in [22], where it is stated that IT can contribute to eco-sustainability in two ways: on the one hand, Green IT (*Green in IT*), when IT itself has an impact on the environment, and on the other hand Green *by* IT, when IT provides tools for making tasks environment friendly.

This means that when the goal pursued is to reduce the energy consumption and the resources used by IT, we are talking about Green *in* IT. When the focus is on using IT to achieve more environment-friendly systems in other domains, then this is Green *by* IT. This same idea is set out in [35], where it is stated that IT can contribute to sustainability from two perspectives. On the one hand, IT can support sustainability by optimising the resources and energy consumption of ICT itself, as induced during the whole life cycle. This concept is called Green IT (*Green in IT*). On the other hand, IT can support sustainability by providing ICT solutions that reduce the environmental impact in other fields [23]. This is the concept known as Green *by* IT.

As we know, IT is composed mainly of software and hardware; this means that the same considerations can be applied at these levels; thus, we can have green in software, green in hardware, green by software and green by hardware (Fig. 1.5).

Finally, we can combine the BY and the IN aspects in software and in hardware. We have called this green software and green hardware, respectively, which together make up Green IT. In Fig. 1.5, these relationships are shown in the form of a diagram.

We will use these concepts when presenting the different definitions found because, as shown previously, there are discrepancies between the concepts and the meanings given by the different authors.

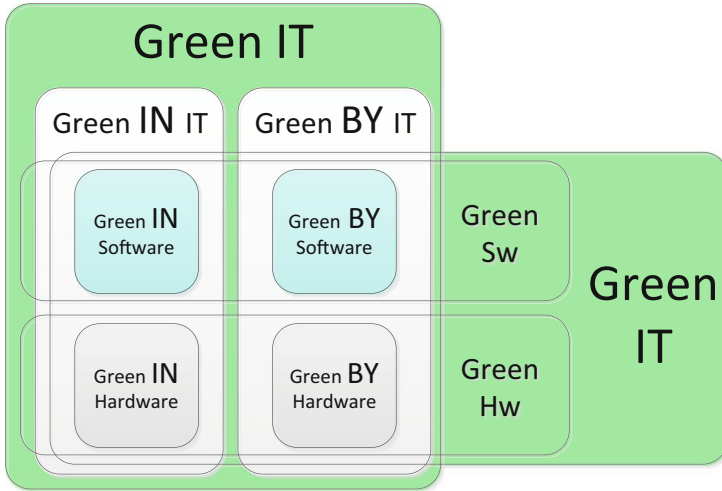


Fig. 1.5 Green software, green hardware and Green IT

We will therefore use the terminology in Fig. 1.5 to unify the different definitions. Readers will find in *italics* and parentheses the alternative concept to that used by the authors.

In [6], the authors talk about Green IT 1.0 and Green IT 2.0, defining both as follows: Green IT projects can be divided into two categories:

1. Green for IT (*Green in IT*), projects aiming to reduce the environmental impact of IT, also known as Green IT 1.0. For example, 10 GbE (10 gigabit per second Ethernet), clean energy to power data centres, hardware virtualisation, cloud computing services (i.e. software as a service (SaaS), Web services, infrastructure as a service, developing platform as a service), data centre outsourcing and co-location services, IT asset disposal and recycling services, IT energy measurement, localised cooling, managed printing services, PC power management software, storage capacity optimisation, thin clients (i.e. low-cost terminals limited to user interface (UI) processing, data processing being run on the server)
2. IT for Green (*Green by IT*), projects aiming to reduce the environmental impact of operations using IT, also known as Green IT 2.0. For example, process automation, remote collaboration, TelePresence, and resource usage management (energy, water, paper, CO₂), for example, Project 2 degrees

The same names of Green 1.0 and 2.0 are used by [44] but in another sense. These authors argue that we are now marching towards the second wave of Green IT. The first wave, Green IT 1.0, was internally focused on re-engineering IT products and processes to improve their energy efficiency and meet compliance requirements. Green IT 2.0 is externally focused on business transformation, sustainability-based IT innovations and enterprise-wide sustainability [44].

The definition given in [25] highlights the fact that there are two concepts used, depending on the nature of IT: Green IT (*Green in IT*), defined as the IT sector's own activity and its impact on environmental efficiency, and green applications of

IT (*Green by IT*) or IT for Green, defined as the impact of IT on the environmental productivity of other sectors, particularly in terms of energy efficiency and their carbon footprint.

A slightly different definition of the concepts is the one given by [42], where waste (materials or substances which harm the environment or demand surplus energy or resources) is used as a definition criterion: Green IT (*Green in IT*) is to produce as little waste as possible during the whole IT life cycle (development, operation and disposal), and Green by IT aims at producing as little waste as possible by means of IT.

In [51], the author talks about sustainability for software engineering (how to make SE itself more sustainable) and sustainability in software engineering (how to improve the sustainability of the systems we develop). Although the author talks about sustainability, we think she refers to what we call green software engineering and, more concretely, to green in software engineering (from the point of view of the process and of the product).

As can be observed, most of the authors use Green IT and Green by IT instead of Green in IT and Green by IT. We maintain that this is confusing because conceptually Green in and by IT are part of Green IT. That is why we have decided to use Green in IT instead of only Green IT, giving Green IT an upper level that contains Green in IT and Green by IT.

1.3.3 Green Software

As remarked in the Intel technical article *Impact of Software on Energy Consumption*, much of the computer energy used (and saved) is based on the effectiveness of hardware energy efficiency and the hardware power states of the computer. But software has an impact as well, in two ways: while running a ‘workload’ and while ‘idle’.⁴

Until recently, the greater part of the work done within the Green IT industry was related to the area of hardware, focusing mainly on improving the energy efficiency of hardware.

Hardware is of course fundamental, but hardware and software together form a whole; one has no meaning without the other. It thus seems self-evident that research work needs to be broadened to include software. As [22] points out, researchers have to pay attention to the effect of software within Green IT.

The trend has been changing in the last few years, and new pieces of work related to the area of green software are emerging. However, there is no common definition of green software [1], a fact that leads us to outline some of the definitions that can be found for the term *green software*.

⁴ <https://noggin.intel.com/content/impact-of-software-on-energy-consumption>

Murugesan and Gangadharan [43] define green software as environment-friendly software that helps improve the environment. The authors classify green software into four categories:

- Software that is greener (consumes less energy to run)
- Embedded software that assists other things in going green (smart operations)
- Sustainability-reporting software (or carbon management software)
- Software for understanding climate change, assessing its implications and forming suitable policy responses

Green software is defined in [57] as software that must fulfil three high-level requirements:

1. The required software engineering processes of software development, maintenance and disposal must save resources and reduce waste.
2. Software execution must save resources and reduce waste.
3. Software must support sustainable development.

According to [22], green software is ‘an application that produces as little waste as possible during its development and operation’.

1.3.3.1 Green by Software Versus Green in Software

As happened with Green IT, green software can be divided into green by software and green in software. Again, the main difference is whether the goal pursued is to have more environment-friendly software or if it is rather to produce software that helps the environment. Figure 1.6 shows this in diagram form.

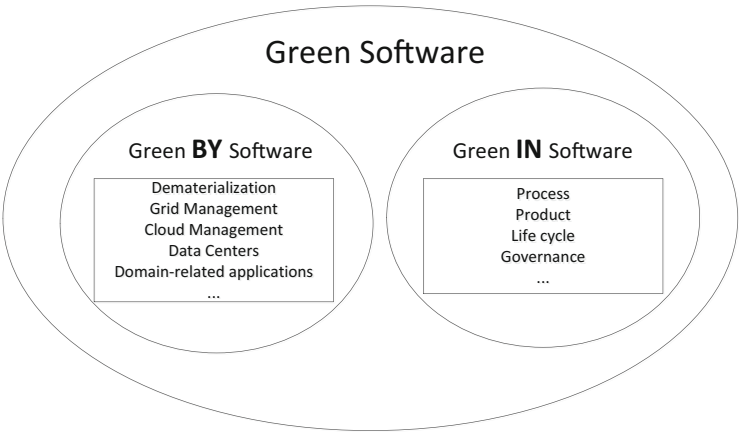


Fig. 1.6 ‘Green by’ and ‘Green in’ software

Green by software appeared some time ago. In general, green by software covers software developed for domains that work in the preservation of the environment, as well as software that helps to manage energy-intensive applications.

On the other hand, green in software is related to how to make software in a more sustainable way resulting in a more sustainable product (this is called green software engineering). The next section will discuss this.

Of course, green in software also includes other aspects aside from software development, such as governance.

1.4 Green in Software Engineering

Green in software engineering is part of green in software and therefore of green software; green in software engineering is the focus of this book. Its main goal is to include green practices as part of the software development process, as well as the rest of activities that are part of software engineering (see Fig. 1.7).

ISO/IEC/IEEE Systems and Software Engineering Vocabulary (SEVOCAB) defines software engineering as ‘the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software’ [32].

Based on this definition, we can define green in software engineering as those practices which apply engineering principles to software by taking into consideration environmental aspects. The development, the operation and the maintenance of software are therefore carried out in a green manner and produce a green software product (Fig. 1.8).

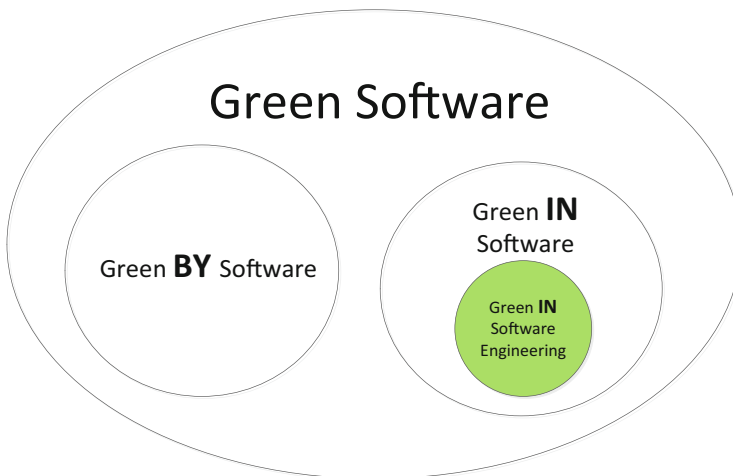


Fig. 1.7 Green in software engineering

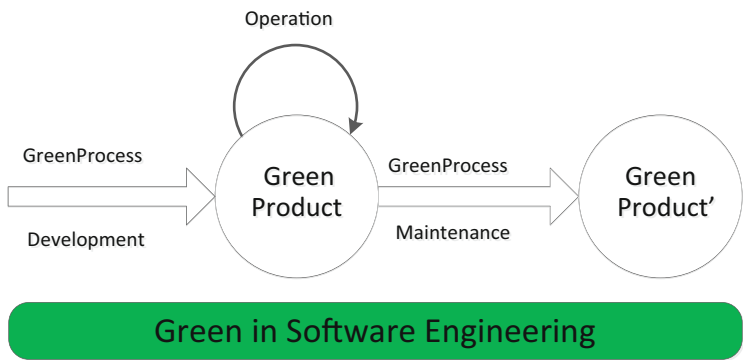


Fig. 1.8 Green in software engineering

In [22], the author explains that software engineering can be green in three ways [60]: (a) by producing green software, (b) by producing software to support environmental consciousness (green by software) and (c) by producing less waste during the development process.

As can be observed, the author mixes green by software with green in software in defining green software engineering. Taking into account the definition of software engineering, it seems that ways (a) and (c) fit, but (b) is does not.

This book gives information on the efforts that are being made nowadays in the arena of green in software engineering. The following chapters will provide information about the different areas of SWEBOK [65], attempting to give a complete snapshot of the present state of the art [32].

1.5 Other Green Concepts in This Book

In this section, we provide a list of the definitions related to green software given in the different chapters of this book but not yet dealt with by us. To that end, Table 1.2 shows the chapter in which the term appears along with the term itself and its definition. It should be underlined that the definition is not necessarily proposed by the authors themselves; it may have been taken from the literature but used in the authors' work. This means that it is advisable to look up the chapter that contains the definition if the reader wishes to find out the exact source of a particular term. It should also be noted that the chapters in this book include definitions of concepts already treated in this chapter that we have not taken into consideration to avoid unnecessary repetitions.

Table 1.2 Green definitions

Chapter number	Term	Definition
2	Software engineering environment (SEE)	It describes the network of people, software, hardware and infrastructure involved in the construction of software
	Design for sustainable behaviour	It denotes how designers can influence users to act in a more environment-friendly manner with respect to their use of products, services and environments
3	Green and sustainable software product	It is a product that should have little impact on the sustainable development and, if it is its specific benefit, promote the pursuit of sustainable objectives
4	Green software services (GSS)	When green software is delivered as online services
5	Green strategy	It is a plan of action intended to accomplish a specific environmental goal
	Green goal	It is an objective that an organisation sets itself to achieve and which is quantified where practical
6	Software engineering for sustainable development	Addresses issues and questions of where and how software and software engineering can help sustainable development
	Green quality	It is only indicated by some quality indicators, either indirectly or directly
	Sustainability of software systems	Systems which generate much waste can be considered more harmful to the environment than those systems that are better at recycling
	Green quality factors	They are factors that define how software supports sustainable development
	Resource efficiency (in software engineering)	It is related to software life cycle, including software design, management, maintenance and disposal
	Software execution resource efficiency	It is related to software execution and software platform usage
	Software client process resource efficiency	It is related to how software stakeholders benefit from software and its software system
	Triftness	It is a factor that evaluates how software reduces waste
	Social sustainability	It is a factor that evaluates how software supports social equality
7	Sustainable	Capable of being upheld; maintainable; and to sustain as 'to keep a person, community, etc., from failing or giving way'; to keep in being; to maintain at the proper level; to support life in; nature, etc., with needs
	Requirements engineering for sustainability	It denotes the concept of using requirements engineering and sustainable development techniques to improve the environmental, social and economic

(continued)

Table 1.2 (continued)

Chapter number	Term	Definition
		sustainability of software systems and their direct and indirect effects on the surrounding business and operational context
	Green requirements engineering	It denotes the same concept as requirements engineering for sustainability with a specific focus on the direct and indirect environmental impacts of systems
	Environmental requirements	Requirements with regard to resource flow, including waste management, can be elicited and analysed by life cycle analysis (LCA)
9	Green software maintenance	It is performed during the entire software working cycle and ends with the retirement of the software product, undertaking at this point all the required activities to reduce the environmental impact of the retired software. It includes modification of code and documentation in order to solve possible deviation of the greenability requirements (or the implementation of new ones), without modifying the original functionality of the source code
	Ecological debt	The cost (in terms of resource usage) of delivering a software system with a greenability degree under the level of the non-functional requirements established by stakeholders, plus the incurring cost required to refactor the system in the future
10	Sustainable software development	It refers to a mode of software development in which resource use aims to meet (product) software needs while ensuring the sustainability of natural systems and the environment
	Greenability	Degree to which a product lasts over time, optimising the parameters, the amounts of energy and the resources used
	Energy efficiency	Degree of efficiency with which a software product consumes energy when performing its functions
	Resource optimisation	Degree to which the resources expended by a software product, when performing its functions, are used in an optimal manner
	Capacity optimisation	Degree to which the maximum limits of a product or system parameter meet requirements in an optimal manner, allocating only those which are necessary
	Perdurability	Degree to which a software product can be used over a long period, being, therefore, easy to modify, adapt and reuse
	Greenability (in use)	Degree to which a software product can be used by optimising its efficiency, by minimising environmental effects and by improving the environmental user perception

(continued)

Table 1.2 (continued)

Chapter number	Term	Definition
	Efficiency optimisation	Optimisation of resources expended in relation to the accuracy and completeness with which users achieve goals. Relevant resources can include time consumption, software resources, etc.
	User’s environmental perception	Degree to which users are satisfied with their perception of the consequences that the use of software will have on the environment
	Minimisation of environmental effects	Degree to which a product or system reduces the effects on the environment in the intended contexts of use
	Quality in use	It is the degree to which a product or system can be used by specific users to meet their needs in order to achieve specific goals with efficiency, freedom from risk, greenability and satisfaction in specific contexts of use

1.6 Challenges and Future Work

Several efforts trying to highlight the importance of including green aspects within software engineering have been undertaken in recent years.

Our task is to raise awareness among software developers (software industries, development departments, etc.) as well as users, who hold in their hands the responsibility of choosing and demanding a software that is more respectful of the environment.

If we achieve this, the whole software development ecosystem will be forced to adopt greener software processes and produce greener software products if they want to remain competitive.

As the issue of green software develops and strengthens, the terminology used will also become clearer. In this chapter, we have attempted to gather the main terms used today. We are certain that these are subject to modification, evolving as the area itself grows in maturity and thereby solving some of the currently present inconsistencies and lack of precision.

Green in software engineering is a nascent research area, so there are plenty of challenges. It is our firm belief that in the next few years we will see research findings and practical applications that we could never even imagine at the present time.

References

1. Abenius S (2009) Green IT & Green software – time and energy savings using existing tools. In: Environmental informatics and industrial environmental protection: concepts, methods and tools. Shaker Verlag, Aachen, pp 57–66
2. Adams WM (2006) The future of sustainability. Re-thinking environment and development in the twenty-first century: technical report, IUCN
3. Amsel N, Ibrahim Z, Malik A, Tomlinson B (2011) Toward sustainable software engineering: NIER track. In: 2011 33rd international conference on software engineering (ICSE), pp 976–979
4. Donnellan B, Sheridan C, Curry E (2011) A capability maturity framework for sustainable information and communication technology. *IT Prof* 13(1):33–40
5. Bachour N (2012) Green IT project management, Chapter 7. In: Hu W, Kaabouch N (eds) Sustainable ICTs and management systems for green computing. IGI, Hershey, PA. ISBN 978-1-4666-1839-8
6. Bachour N, Chasteen L (2010) Optimizing the value of Green IT projects within organizations. In: Green technologies conference
7. Bose R, Luo X (2012) Green IT adoption: a process management approach. *Int J Account Inform Manag* 20(1):63–67
8. Brodtkin J (2008) Economy driving Green IT initiatives. *Netw World* 25(48):16
9. Brown B, Hanson M, Liverman D, Merideth R (1987) Global sustainability: toward definition. *Environ Manag* 11(6):713–719
10. Calero C, Bertoa MF, Moraga MA (2013) A systematic literature review for software sustainability measures. In: GREENS 2013: Second international workshop on green and sustainable software, pp 46–53
11. Capra E, Francelanci C, Slaughter SA (2012) Is software “green”? Application development environments and energy efficiency in open source applications. *Inform Software Tech* 54(1):60–71
12. Castelo B, Lima M (2006) Corporate social responsibility and resource-based perspectives. *J Bus Ethics* 69:111–132. doi:[10.1007/s10551-006-9071-z](https://doi.org/10.1007/s10551-006-9071-z), Springer 2006
13. Cazier JA, Hopkins BE (2011) Doing the right thing for the environment just got easier with a little help from information systems. In: Proceedings > Proceedings of SIGGreen workshop. Sprouts: working papers on information systems, vol 11, issue 10. <http://sprouts.aisnet.org/11-10>
14. Chen AJ, Watson RT, Boudreau MC, Karahanna E (2009) Organizational adoption of green IS & IT: An institutional perspective. In: ICIS 2009 proceedings, 142. Retrieved from <http://aisel.aisnet.org/icis2009/142>
15. Chia-Tien Dan L, Kai Q (2010) Green computing methodology for next generation computing scientists. In: Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual, pp 250–251
16. Collins (2013) Collins dictionary. <http://www.collinsdictionary.com/>
17. Dahlsrud A (2008) How corporate social responsibility is defined: an analysis of 37 definitions. *Corp Soc Responsibility Environ Manag* 15:1–13. doi:[10.1002/csr.132](https://doi.org/10.1002/csr.132), Wiley InterScience
18. Dick M, Drangmeister J, Kern E, Naumann S (2013) Green software engineering with agile methods in green and sustainable software (GREENS). In: 2013 2nd International Workshop, pp 78–85
19. Dick M, Naumann S (2010) Enhancing software engineering processes towards sustainable software product design. In: Greve K, Cremers AB (eds) *EnviroInfo 2010: Integration of environmental information in Europe*. Proceedings of the 24th International Conference EnviroInfo, Cologne/Bonn, Germany. Shaker, Aachen, pp 706–715
20. Du W, Pan SL, Zuo M (2013) How to balance sustainability and profitability in technology organizations: an ambidextrous perspective. *IEEE Trans Eng Manag* 60(2):366–385. doi:[10.1109/TEM.2012.2206113](https://doi.org/10.1109/TEM.2012.2206113)

21. Easterbrook S (2010) Climate change: a grand software challenge. In: FoSER 2010, November 7–8, Santa Fe, New Mexico, USA, ACM 978-1-4503-0427-6/10/11, pp 99–103
22. Erdelyi K (2013) Special factors of development of green software supporting eco sustainability. In: IEEE 11th international symposium on intelligent systems and informatics (SISY), pp 337–340
23. Erdmann L, Hilty J, Goodman P (2004) Arnfalk the future impact of ICTs on environmental sustainability in technical report EUR 21384 EN. Technical report series EUR 21384 EN. European Commission; Joint Research Centre; IPTS – Institute for Prospective Technological Studies, Seville
24. Ericsson (2013) Ericsson energy, carbon report. On the impact of the networked society. EAB-13:036469 Uen. Ericsson AB. <http://www.ericsson.com/res/docs/2013/ericsson-energy-and-carbon-report.pdf>. Accessed on April 2014
25. Faucheux S, Nicolai I (2011) IT for Green and Green IT: a proposed typology of eco-innovation. *Ecol Econ* 70(11):2020–2027
26. Green Book (2000) European Commission, March 2000
27. Hasan H, Molla A, Cooper V (2012) Towards a green IS taxonomy. In: Proceedings of SIGGreen workshop. Sprouts: Working papers on information systems, vol 12, issue 25, <http://sprouts.aisnet.org/12-25>
28. Hedwig M, Malkowski S, Neumann D (2009) Taming energy costs of large enterprise systems through adaptive provisioning. In: International conference on information systems (ICIS 2009), paper 140. Retrieved from <http://aisel.aisnet.org/icis2009/140>
29. Helen H (2010) Taking the green IS message to the world. In: SIGGreen proceedings, pp 139–142. <http://siggreen-icis2010-workshop.wikispaces.com/file/view/SIGGreenICIS2010-WorkshopEBook.pdf/189649215/SIGGreenICIS2010WorkshopEBook.pdf>
30. Hilty LM, Arnfalk P, Erdmann L, Goodman J, Lehmann M, Wäger PA (2006) The relevance of information and communication technologies for environmental sustainability – a prospective simulation study. *Environ Model Software* 21(11):1618–1629
31. IDC (2009) Aid to recovery: the economic impact of IT, software, and the Microsoft ecosystem on the global economy
32. ISO/IEC/IEEE 24765 (2010) Systems and software engineering – Vocabulary
33. ISO26000:2010 (2010) Guidance on social responsibility. <https://www.iso.org/obp/ui/#iso:std:iso:26000:ed-1:vl:en>
34. Johann T, Dick M, Kern E, Naumann S (2011) Sustainable development, sustainable software, and sustainable software engineering: an integrated approach. In: 2011 International symposium on humanities, science & engineering research (SHUSER), pp 34–39
35. Kern E, Dick M, Naumann S, Guldner A, Johann T (2013) Green software and green software engineering – definitions, measurements, and quality aspects. In: First international conference on information and communication technologies for sustainability, pp 87–94.
36. Koomey J (2011) Growth in data center electricity use 2005 to 2010. Analytics, Oakland, CA, August 1. <http://www.analyticspress.com/datacenters.html>
37. Manteuffel C, Loakeimidis S (2012) A systematic mapping study on sustainable software engineering: a research preview. In: 9th Student colloquium, pp 35–40
38. Masanet E, Shehabi A, Ramakrishnan L, Liang J, Ma X, Walker B, Hendrix V, Mantha P (2013) The energy efficiency potential of cloud-based software: a U.S. case study. Lawrence Berkeley National Laboratory, Berkeley, CA. http://crd.lbl.gov/assets/pubs_presos/ACS/cloud_efficiency_study.pdf. Accessed in April 2014
39. Merriam-Webster (2013) <http://www.merriam-webster.com/>
40. Mocigemba D (2006) Sustainable computing. *Poiesis & Praxis Int J Technol Assess Ethics Sci* 4(3):163–184
41. Mohan K, Ramesh B, Cao L, Sarkar S (2012) Managing disruptive and sustaining innovations in Green IT. *IT Prof* 14(6):22–29
42. Molla A, Cooper VA, Pittayachawan S, IT and Eco-sustainability (2009) Developing and validating a Green IT readiness model. In: International conference on information systems (ICIS 2009). Paper 141. Retrieved from <http://aisel.aisnet.org/icis2009/141>

43. Murugesan S, Gangadharan GR (eds) (2012) *Harnessing Green IT: principles and practices*. Wiley, UK. ISBN: 978-1-119-97005-7
44. Murugesan S, Laplante PA (2011) IT for a greener planet. *IT Pro* January/February, 16–20
45. Murugesan S (2010) Making it Green. *IT Prof* 12:4–5
46. Naumann S, Dick M, Kern E, Johann T (2011) The greensoft model: a reference model for green and sustainable software and its engineering. *Sustain Comput Informat Syst* 1(4):294–304
47. Penzenstadler B, Fleischmann A (2011) Teach sustainability in software engineering? In: 2011 24th IEEE-CS conference on software engineering education and training (CSEE&T), pp 454–458
48. Penzenstadler B, Femmer H (2013) A generic model for sustainability with process and product-specific instances. In: *Proceedings of the 2013 workshop on green in/by software engineering 2013*, ACM, Fukuoka, Japan, pp 3–8
49. Penzenstadler B, Raturi A, Richardson D, Calero C, Femmer H, Franch X (2014) Systematic mapping study on software engineering for sustainability (SE4S). In: 18th International conference on evaluation and assessment in software engineering
50. Penzenstadler B et al (2012) Sustainability in software engineering: a systematic literature review for building up a knowledge base. In: 16th international conference on evaluation and assessment in software engineering (EASE)
51. Penzenstadler B (2013) Towards a definition of sustainability in and for software engineering. In: SAC'13. ACM 978-1-4503-1656-9/13/03, pp 1183–1185
52. Goodland R (2002) Sustainability: human, social, economic and environmental, *Encyclopedia of global environmental change*. Wiley, UK
53. Seidel S, vom Brocke J (2010) Call for action: investigating the role of business process management in green IS. In: *SIGGreen proceedings*. <http://siggreen-icis2010-workshop.wikispaces.com/file/view/SIGGreenICIS2010WorkshopEBook.pdf/189649215/SIGGreen-ICIS2010WorkshopEBook.pdf>, pp 132–133
54. Sobotta A, Sobotta I, Gotze J (eds) (2010) *Greening IT. How greener IT can form a solid foundation for a low-carbon society*. The Greening IT Initiative Italy. ISBN 978-87-91936-02-9
55. Sroufe R, Sarkis J (eds) (2007) *Strategic sustainability: the state of the art in corporate environmental management systems*. Greenleaf, Sheffield, UK
56. Steigerwald B, Agrawal A (2011) Developing green software. <https://software.intel.com/enus/node/183291?wapkw=developing+green+software>
57. Taina J (2011) Good, bad, and beautiful software. In search of green software quality factors. *CEPIS Upgrade* XII 4:22–27
58. Tate K (2006) *Sustainable software development: an agile perspective*. Addison-Wesley, Upper Saddle River, NJ
59. The Climate Group (2008) *SMART 2020: enabling the low carbon economy in the information age*. The Global eSustainability Initiative, Brussels
60. Tomlinson B, Silberman SS, White J (2011) Can more efficient IT be worse for the environment? *Computer* 44(1):87–89. doi:10.1109/MC.2011.10, ISSN:0018-9162
61. Unhelkar B (2011) *Green IT strategies and applications. Using environmental intelligence*. CRC, Boca Raton, FL
62. United Nations World Commission on Environment and Development (1987) Report of the World Commission on Environment and Development: our common future. In: United Nations conference on environment and development
63. vom Brocke J, Seidel S, Recker J (eds) (2012) *Green business process management: towards the sustainable enterprise*. Springer, Berlin, p XII, 263 p
64. Watson RT, Boudreau M-C, Chen AJW (2010) Information systems and environmentally sustainable development: energy informatics and new directions for the IS community. *MIS Q* 34(1):23–38
65. IEEE (2014). *SWEBOK V3.0. Guide to the Software Engineering Body of Knowledge*. Bourque, P. and Fairley, R.E. (eds.), IEEE Computer Society

Green in Software Engineering

Calero, C.; Piattini, M. (Eds.)

2015, XII, 327 p. 111 illus., 68 illus. in color., Hardcover

ISBN: 978-3-319-08580-7