International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 6, Issue 4, July- August 2017 ISSN 2278-6856

COHESIVE ANALYSIS OF SUSTAINABILITY OF GREEN COMPUTING IN SOFTWARE ENGINEERING

Dr. Brijesh Khandelwal[1], Sameera Khan[2], Shama Parveen[3]

[1]Associate Professor, Amity School of Engineering & Technology, Amity University Chhattisgarh, Raipur,

[2]Assistant Professor, Amity School of Engineering & Technology, Amity University Chhattisgarh, Raipur,

[3]Assistant Professor, Amity School of Engineering & Technology, Amity University Chhattisgarh, Raipur,

Abstract
In a drive that fosters admiration for the environment with ICT, IT or software, etc., are called Green or Greening ICT/IT/Software or occasionally sustainability of green computing in IT. The problem that arises is that, as in every new discipline, there is no lucid plan of defining and conceptualizing it. However, the fact is that Green IT is not only a trend; it is becoming a necessity as more and more organizations are implementing some form of sustainable solutions. In applications in which embedded devices cooperate with ICT (information and communication technology) systems to make industrial processes more efficient, reduce waste or raw materials, and save the environment, the concept of green software becomes increasingly complex. To deal with this issue, the green software community has introduced the concepts of greening ICT or greening through ICT.
Focus of this research paper is confined largely to have an organized analysis of achieving and enhancing green computing in software engineering processes with the possible means. This may also reveal possible scope of green computing in process of software development.

Keywords: Green Computing, Software Sustainability, S/w Development Life Cycle, Software Engineering

Sustainability
The green software community has analyzed the problem and focused on two areas related to information and communication technology (ICT). Greening ICT is about lowering the energy consumption of ICT itself (computers, printers, networks and routers, datacenters, and so on) [1]. In contrast, greening through ICT refers to how ICT can decrease the energy consumption of systems it controls or monitors. Greening through ICT could have a much larger effect—an estimated 15 percent reduction of the total energy consumption by 2020. [2]. The sustainability in IS must take into account aspects such as efficiency systems, forecasting, reporting and awareness, energy-efficient home computing and behaviour modification.

There is significant evidence that a growing share of green IT will be addressed by green computing in the process of software engineering. From a management perspective, making software greener is a challenging task that involves complex tradeoffs among stakeholders. From a technical perspective, as per Luca Ardito et al. [3] several tools and good practices are available, although they are not yet well integrated in an organized framework which could able to provide software developers and designers a unifying view.

Taking into account that our focus is on software engineering (SE), adjoining figure- 1.0 summarizes the nested sustainability that relates organization to information systems or ICT and to software engineering. Nested Sustainability establishes the philosophy of getting more narrower towards software engineering (on a micro domain) for achieving green computing from macro domain of IS sustainability.

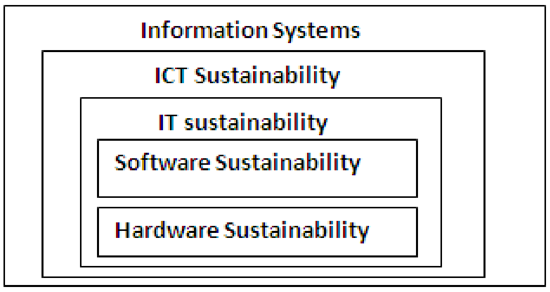

Figure 1.0- Sustainability Nesting Approach

Aligning efforts for IS/ICT/IT Sustainability
As articulated in the SIG Green Statement by Hasan H et al.[13], 'the Information Systems discipline within software engineering environment, 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'.

Watson RT et al. [25] recommend using the term IS sustainability over IT sustainability, because they consider that the exclusive focus on information technologies is too narrow.
- 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 [12] points to dematerialization and increased efficiency as the two main ways of aligning ICT with sustainability.

Software Sustainability
There are several areas in which software sustainability needs to be applied by means of software systems, software products, Web applications, data centers etc. Various works are in process, but most of this concerns data centers, which consume significantly higher energy than commercial office space [18]. One of the common areas where the growing demand of globalized software solutions made possible by Cloud infrastructure has drastically increased the energy consumption of data centers, which has become a critical issue and a threat to nature even. High energy consumption not only translates to high operational cost, which reduces the profit margin of Cloud providers, but also leads to high carbon emissions which is not environmentally friendly.

Hence, energy-efficient solutions are required to minimize the impact of software solutions by Cloud computing on the environment. In order to design such solutions, deep analysis of Cloud is required with respect to their power efficiency.

According to Calero C, Bertoa MF, Moraga MA (2013) [6], 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 [7]. Energy efficiency has never been a key requirement in the development of software-intensive technologies, and so in my belief this is a very large gray potential area for improving efficiency as also suggested by The Climate Group (2008) SMART 2020 [24].

As remarked by Easterbrook S in 2010 [10], software plays a major role, both as part of the problem and as part of the solution. The behavior of the software has significant influence on whether the energy-saving features built into the platform are effective [22]. According to Dick M et al.[8], 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’ [21].

Perceptive in Sustainable software engineering
As per Amsel N et al. [5], 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

Dick M, Naumann S (2010) [9] is having the philosophy that 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 optimization of the software product [16] 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 optimized

Manteuffel C, Loakeimidis S (2012) [19] believes that 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

According to Tate K (2006) [23], 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

The objective of green and sustainable software engineering [17] is the enhancement of software engineering which aims on-

- The direct and indirect consumption of natural resources and energy
- 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 optimize these facts

The aim of software engineering for sustainability [14] (SE4S) is to make use of methods and tools in order to Achieve

Green Software
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

Erdelyi K [11] points out that researchers have to pay attention to the effect of software within Green IT.

What will not solve the purpose up to any measure is that- In core software development (specifically coding), we can save energy by efficiently using computing resources and avoiding recurring work, such as recompiling.

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 [4], a fact that leads us to outline some of the definitions that can be found for the term green software.

Murugesan and Gangadharan [20] 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 fulfill following three high-level requirements:

i. The required software engineering processes of software development, maintenance and disposal must save resources and reduce waste.
ii. Software execution must save resources and reduce waste.
iii. Software must support sustainable development.

Again according to Erdelyi K [11], green software is 'an application that produces as little waste as possible during its development and operation'.

The software development life cycle and related development tools and methodologies rarely, if ever, consider energy efficiency as an objective [7]

Conceptually, 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.

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. The practices which apply engineering principles to software by taking into consideration environmental aspects is referred to as green in software engineering.

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' [15].

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.

Analyzing Efforts for Greenness by/in Software Development Greenness by software: which dedicated software engineering methods and tools are required to model, analyze and minimize the ecologic/environmental impact of industrial solutions by means of software?

- Industrial case studies such as smart home, smart cities, smart embedded systems and lessons learnt from software engineering perspective
- Means to make software solutions dynamically adaptive in their greenness attributes
- Means to measure the effect of green-ness by software
- Means to make hardware greener by the use of software

Greenness in Software (Suggested approach)
Following table 1.0 describe about which dedicated software engineering methods and tools are required during the software development process to model, analyze and minimize the ecologic impact/environmental impact including the energy consumption of software systems?

Table 1.0 Software Development Life Cycle and Greenness Efforts

Phase/ Stage in Software Development Life Cycle	**Efforts**	**Linkage, if any**
Requirement Analysis phase:	Means to obtain and model greenness requirements	To begin with; No Linkage
	Means to identify and analyze the opposite arrangements among the functional requirements, greenness requirements	Linked to former means of obtaining models

	themselves and other worth attributes	
Design Architectural phase:	Means to model greenness attributes of software architecture	To begin with; No Linkage
	Means to analyze the trade-offs among the functional requirements, greenness attributes themselves and other quality attributes	Linked to earlier means
	Methods to evaluate the greenness of software architecture	Linked to earlier means
	Means to trace greenness attributes to the greenness requirements	Linked to earlier means
Implementati on phase	(Domain-specific) languages to implement greener software	To begin with; No Linkage
	Compilation methods to achieve greener software	No Linkage
	IDE support to guide programmers in implementing greener software	No Linkage
	Means to measure the greenness of implementations	Linked to earlier means
	Means to assess the trade-offs between the functionality, the greenness and other quality attributes	Linked to earlier means
Testing phase	Metrics, methods and tools to generate test scenarios to assess the greenness of software	To begin with; No Linkage
	Energy profiling techniques	No Linkage
Practical Operational Phase	Means to monitor the greenness attributes	To begin with; No Linkage
	Means to dynamically adapt the software to fulfil its greenness attributes, e.g. energy aware computing methods	Linked to means of Implementat ion phase
	Means to profile energy consumption and to carbon foot printing	No Linkage
	Applications that help to reduce energy consumption in facility management, in production, mobility, and in embedded systems	No Linkage

The State of Confront

One overarching goal of much of the energy consumption research is to produce a model that generalizes across many applications. The use-case of such a model is that developers do not have access to expensive hardware and cannot accurately measure the energy consumption of their applications – thus they must rely upon estimations based on different kinds of analyses and models. But these generalized models suffer from the range of hardware, operating systems, environment, software domains, and versions of software. In the mobile arena the wide-range of screen-sizes, memory sizes, and kinds of processors tends to hamper such generalized models. Furthermore the Android ecosystem is considered fragmented in terms of hardware and software.

The issue of different operating systems is also relevant, Windows and Linux do not share the same code base and handle energy management differently. Android includes customizations distinct from Linux as well. Furthermore there are different versions and distributions of Linux, Windows, Android, OSX, and iOS. Thus measurements from one environment might not hold for another. Furthermore generalizable models suffer from a lack of data. Energy traces are not prevalent in the operational data within Github git repositories or other publicly available repositories. Continuous integration tools tend not to measure or estimate energy.

Conclusive Cohesive Analysis and Recommendations
Though, at present, it seems to be a trivial to achieve green computing through software engineering, yet incessant research in this area will open several windows which are not even realized. 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 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 Computing with software engineering is a promising research area; therefore there are numerous confronts. It is our firm belief that in the next few years we will witness research findings and practical applications that we could never even imagined at the present time.

References

[1] S. Murugesan, "Harnessing Green IT: Principles and Practices," IT Professional, vol. 10, no. 1, 2008..
[2] Smart 2020: Enabling the Low Carbon Economy in the Information Age, tech. report, Climate Group, 2008; www. smart2020.org/_assets/fi les/02_ Smart2020Report.pdf.
[3] Luca Ardito, Giuseppe Procaccianti, and Marco Torchiano "Understanding Green Software Development: A Conceptual Framework": IT Pro January/February 2015.
[4] 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.
[5] Amsel N, Ibrahim Z, Malik A, Tomlinson B (2011) Toward sustainable software engineering: NIER track. In: 2011 33rd international conference on software engineering (ICSE)
[6] Calero C, Bertoa MF, Moraga MA (2013) A systematic literature review for software sustainability measures.
[7] Capra E, Francalanci C, Slaughter SA (2012) Is software "green"? Application development environments and energy efficiency in open source applications. Inform Software Tech 54 (1)
[8] Naumann S , Dick M, Drangmeister J, Kern E(2013) Green software engineering with agile methods in green and sustainable software (GREENS).
[9] Dick M, Naumann S (2010) Enhancing software engineering processes towards sustainable software product design. Proceedings of the 24th International Conference EnviroInfo, Cologne/Bonn, Germany. Shaker, Aachen., Greve K, Cremers AB (eds) EnviroInfo 2010: Integration of environmental information in Europe.
[10] 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.
[11] Erdelyi K (2013) Special factors of development of green software supporting eco sustainability. In: IEEE 11th international symposium on intelligent systems and informatics.
[12] 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-energyand-carbon-report.pdf. Accessed on April 2014
[13] 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.
[14] IDC (2009) Aid to recovery: the economic impact of IT, software, and the Microsoft ecosystem on the global economy
[15] ISO/IEC/IEEE 24765 (2010) Systems and software engineering – Vocabulary
[16] ISO26000:2010 (2010) Guidance on social responsibility.https://www.iso.org/obp/ui/#iso:std: iso:26000:ed-1:v1:en
[17] 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.
[18] Koomey J (2011) Growth in data center electricity use 2005 to 2010. Analytics, Oakland, CA, August 1. http://www.analyticspress.com/datacenters.html
[19] Manteuffel C, Loakeimidis S (2012) A systematic mapping study on sustainable software engineering: a research preview. In: 9th Student colloquium.
[20] Murugesan S, Gangadharan GR (eds) (2012) Harnessing Green IT: principles and practices. Wiley, UK. ISBN: 978-1-119-97005-7
[21] 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).
[22] Steigerwald B, Agrawal A (2011) Developing green software. https://software.intel.com/enus/ node/183291?developing+green+software
[23] Tate K (2006) Sustainable software development: an agile perspective. Addison-Wesley, Upper Saddle River, NJ
[24] The Climate Group (2008) SMART 2020: The Global eSustainability Initiative, Brussels- Enabling the low carbon economy in the information age.
[25] 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).

AUTHOR

Dr. Brijesh Khandelwal presently working as Associate Professor at Amity School of Engineering & Technology, Amity University Chhattisgarh, Raipur. Dr. Khandelwal has rich & diverse experience in academia and has several publications in International/ National Journals & Conferences. Dr. Khandelwal did MCA from Lucknow University in year 1994. In 2001, he became Sun Certified Programmer with Sun Microsystems. He was awarded

PhD (Appllied Economics) from University of Lucknow in year 2007. He did MBA in 2010 from Punjab Technical University. In 2010 he also became licentiate in Life Insurance from Insurance Institute of India, Mumbai. He also has been awarded PhD on Computer Science in year 2017 from Shri Venkateshwara University, Gajraula.

Sameera Khan is a Assistant Professor in the Department of Computer Science and Engineering, Amity University Chhattisgarh, Raipur. She is pursuing PhD from Chhattisgarh Swami Vivekanand Technical University. She received her Master OF Technology (CSE) from the same University and Bachelor of Engineering (CSE) in 2008 from Raipur Institute of Technology, Raipur affiliated to Pt. Ravishankar University, Raipur (C.G).Her research interest are Digital Image Processing and Artificial neural network.

Shama Parveen is a B.E. in Computer Science Engineering from Chhattisgarh Swami Vivekanand Technical University and M.Tech in Computer Science Engineering from Amity University, Noida, Uttar Pradesh. Her research area includes cyber crime, cyber security and emerging technologies. Currently she is working as Assistant Professor in Amity School of Engineering & Technology, Amity University Chhattisgarh, Raipur.