iJRASET

International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

www.ijraset.com

Call: ○ 08813907089   E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Elegant Approach For Green Software Engineering- Green Software Development Life Cycle (GSDLC)

Harsha Pati[1] Alpana Sonje[2]
Ashoka Center for Business and Computer Studies. Nashik

Abstract: green software engineering is the attempt to apply "green" principles known from hardware products also on software products, software development processes and their underlying software process models. This research paper proposed design of conceptual reference model supporting it professionals and software users in the sustainable development and usage of software. Beginning with a detailed analysis, we shaped a conceptual reference model, which has the objective to support software developers and software users in creating and using software in a more sustainable way. The "green software" reference model comprises a holistic lifecycle model for software products, sustainability criteria and metrics of software products, a "green software engineering" procedure model, and recommendations for actions.
Keywords: green software engineering, energy consumption, sustainable, green it

I. INTRODUCTION

The author in [2] 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. In [11], 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'. The increasing amount of the energy consumption of today's IT solutions significantly contributes to green house gas emissions. "Green Computing" or "Green IT" emphasizes on the need for reducing the environmental impacts of IT solutions by reducing their energy consumption and their green house gas emissions. Among others, green computing can be achieved in software and by software. While greening by software aims at saving energy (or other resources) by the help of software, greening in software aims at reducing the environmental impact caused by the software itself. Besides saving energy and aiming for efficiency, green computing is a complex trade-off between efficiently using ay required resource and keeping the environmental impact low. This has consequences for architectural decisions. 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 [7], 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). In [10], 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. All of this can be summarised under the concept of 'strategic sustainability', introduced by [14]. Most people claim that they will pay more for a green product [6]. In early 2010, the ISO 26000 standard [9] 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.Green-ness in the software is an emerging quality attribute that must be taken into the account in each phase of the software development process at each level of the IT system from the application level via middle-ware to operating system and hardware. Achieving green-ness by software requires methods and techniques that support finding, realizing, and measuring software solutions that make infrastructure smarter, virtualize processes, contribute to dematerialization or new solutions like smart grids. Typical examples are applications that help to reduce energy consumption in facility management, in production, mobility, and in embedded systems. The analysis of all factors that have an environmental impact and the search for the optimal trade-off

1903

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

therefore has to be included in software development methods.

A. Motivation

Rising energy demands and the growing negative environmental impact from the increased adoption of IT services are motivating the green movement in IT, which places great importance on the design and implementation of green solutions. Green IT is applicable to a range of high-tech domains, including datacenters, mobile computing, and embedded systems. Annual global carbon dioxide emissions recently reached 9.1 billion tons, the highest level in human history — 49 percent higher than in 1990 (the Kyoto reference year). At least 2 percent of global carbon dioxide emissions can be attributed to IT systems, and further increases are expected as new IT systems are deployed daily. Reducing IT systems' energy consumption and related carbon dioxide emissions is a vital undertaking.

Most studies and regulatory controls focus on hardware-related measurement, analysis, and control for energy consumption. However, all forms of hardware systems are controlled by software components. Although software systems don't consume energy directly, they affect hardware utilization, leading to indirect energy consumption. Therefore, it's important to engineer software so that its energy consumption is optimized. The software engineering research domain has recently been paying attention to sustainability, as the increased number of publications, empirical studies, and conferences on the topic demonstrate.

Green IT aims for minimal environmental impact from the design, production, and use of computers, servers, monitors, printers, storage devices, and networking and communications systems. It focuses on product and process efficiency, in terms of environmental sustainability, as well as applying IT to create energy-efficient, environmentally sustainable business processes and practices. IT can support, assist, and leverage other environmental initiatives and help in creating green awareness.

II.   LITERATURE REVIEW

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 [5]. In [3], 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'.

Green software engineering is very recent topic of research in field of software engineering. It got very immediate response internationally by researchers group due to its connection with environment. In the year 2010 software sustainability considerations safety is an emergent property that arises when the system components interact within an environment Albertao, Xiao, Tian, Lu, Zhang and Liu (2010) introduce software engi-neering metrics that can be used to assess eco-nomic, social and environ-mental sustain-ability of soft-ware projects. Capra, Francalanci and Slaughter (2011) [5] proved that different designs of software functional appli-cations could cause differ-ent significant effect on energy efficiency.

Lami, Fabbrini and Fusani (2012) integrate greenness cul-ture in develop-ing software by referring to popular stand-ards in evaluat-ing the process capability and sustainability. Kern, Dick, Naumann and Hiller (2014) introduce carbon footprint calculation method for software prod-uct life cycle. Kocak, Apltekin and Bener (2014) investigate relationships between quality and environmental attributes.

In developed countries researches are mainly focused on green and sustainable software engineering. But in developing countries like India there is no more research on this. The developing countries are developing more software applications and ICT products alone for the developed countries, but Green sustainable software engineering process is a major issue that is to be solved to make efficient software models.

Significance of the study in the context of current status:

The India IT industry has witnessed tremendous growth is the past decade, resulting in large IT companies and inceptions of several promising start-ups and creating software products. Rising energy demands and the growing negative environmental impact from the increased adoption of IT services are motivating the green movement in IT, which places great importance on the design and implementation of green solutions.

Like green IT generally, greening in software seeks to reduce the environmental impact of the software itself. Greenness is thus an emerging software quality attribute to consider, as well. Software companies are beginning to confront the conflict between being as environmentally friendly as possible and customer pressure for new functional requirements and high quality. Yet, software systems can also play a proactive role in saving energy by providing feedback about the way they consume resources and, ideally, leading people to change behaviors and create greener processes. In a recent update of this work, the authors observed that the number of proposals has increased considerably over the last 2 years [12]. This fact serves to demonstrate that there is an ever-growing concern

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

to tackle sustainability in the context of software engineering.

III.   GREEN SOFTWARE DEVELOPMENT LIFE CYCLE

As noted in [4], 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 [5]. 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 [16].

As remarked by [8], 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 [15]. Software developer use five generic framework activities during the development of small programs, the creation of large Web applications and for the engineering of large, complex computer-based systems. These framework activities are communication, planning, modeling, construction and deployment.The details for any software development process may be depend on the nature of the software that they are system software, application software, ubiquitous software ,AI based or embedded software but the framework activities remain the same.
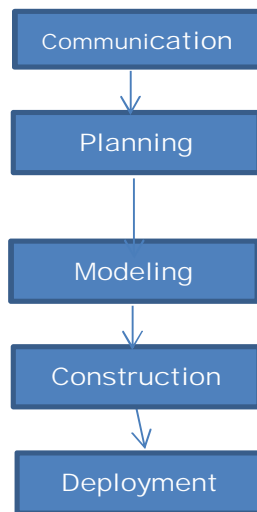

Fig 1. Green Software Engineering Life Cycle

GSDLC also uses the same framework activities with sustainability effects. So the objective of this GSDLC is to assign criteria to the different life cycle phases that lead to or result in sustainability relevant effects.The first framework activity involves heavy communication and collaboration with the customers (and other stakeholders) and encompasses requirements gathering and other related activities. In planning activity software developers establishes a plan for the software engineering work for describing the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced and a work schedule. After completion of Planning framework activity modeling phase start which encompasses the creation of models that allow the developer and the customer to better understand software requirements and the design that will achieve those requirements. In all above three phases GSDLC suggest in many aspect for work upon by energy consumption can be optimized like Hardware Obsolescence and the Carbon Footprint.   With the agreement of developer and the client modeling is converted in to soft form and coding is started. Coding generation can be done manual or by using any RAD tool. Construction activity also consider generation of test cases and its application on software for the testing that is required to uncover errors in the code. The model takes aspects into account like Efficiency, Memory Usage, Idleness and Number of Methods which reflects size of application. During the all above framework activities a section Green team remain active for continual reviews and previews for assess the problem identification, designing, coding   and develop for find the alternative solutions in order to choose the better "more sustainable" solution.

 The continuous Process Assessment activity quantifies and assesses impacts on sustainability, which result from the software development process itself. Process Assessments (mainly impacts that result from the development phase). Thus it covers impacts

1905

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

over the whole lifecycle of the software which results in best practices regarding sustainability issues of software products or development processes. In deployment activity the software is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

IV.   METHODOLOGY

The methodology for this research will be through primary source of information as well as secondary source.

Primary Source:-

Data will be collected through questionnaire, interviews and observations.

Structured questionnaire will be prepared for software developers and prime users

Interview will be conducted for stake holders of software development industries.

Experiments will be conducted to analyses efficiency of software in different circumstance and conditions.

Secondary Source:- Data will be collected from journals, books, magazines and electronic sources.

Basically, Green software engineering is about to optimize use of energy consumption by software. As we soft wares are not directly consumes the energy instead the we measured the energy consumption of a specific combination of hardware components. Since software brings hardware to consume energy by execution software components Embedded softwares , Operating systems and application software have different energy consumption which depends on how their code is. In our experiment we test different software in lab We power meter for measure energy consume by software. By analyzing data we find that open system softwares are optimized energy consumption. Like Mozilla consume less energy then internet explorer . Linux based operating system save more enegery then windows operating system.

V.   CONCLUSION

GSDL Model is approach which works upon Green sustainable software engineering.The conclsion of this research is awareness among the developers regarding Green approach is very important. The Wrong practice habits and improper use and choice of Hardware and software platforms bats the maximum run in direction of more energy consumption.

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]   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

[3]   Bachour N, Chasteen L (2010) Optimizing the value of Green IT projects within organizations.In: Green technologies conference

[4]   Calero C, Bertoa MF, Moraga MA (2013) A systematic literature review for software sustainabilitymeasures. In: GREENS 2013: Second international workshop on green and sustainable software, pp 46–53

[5]   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):60–71

[6]   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

[7]   Du W, Pan SL, Zuo M (2013) How to balance sustainability and profitability in technologyorganizations: an ambidextrous perspective. IEEE Trans Eng Manag 60(2):366–385.doi:10.1109/TEM.2012.2206113

[8]   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

[9]   ISO26000:2010 (2010) Guidance on social responsibility. https://www.iso.org/obp/ui/#iso:std:iso:26000:ed-1:v1:en

[10] 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

[11] Murugesan S (2010) Making it Green. IT Prof 12:4–5

[12] 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

[13] 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)

[14] Sroufe R, Sarkis J (eds) (2007) Strategic sustainability: the state of the art in corporate environmental management systems. Greenleaf, Sheffield, UK

[15] Steigerwald B, Agrawal A (2011) Developing green software. https://software.intel.com/enus/node/183291?wapkw¼developing+green+software

[16] The Climate Group (2008) SMART 2020: enabling the low carbon economy in the informationage. The Global eSustainability Initiative, Brussels

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 ⟋ (24*7 Support on Whatsapp)