The role of green it in the non-functional software development requirements: Perspectives for functional design

The role of green it in the non-functional software development requirements: Perspectives for functional design

T Velásquez[1], **A Messino**[2], **H E Navarro**[1], **and A A Ovalle**[2]
[1] Grupo de Investigación en Tecnología y Desarrollo en ingenierías, Universidad Francisco de Paula Santander, Seccional Ocaña, Colombia
[2] Grupo de Investigación IngeBioCaribe, Universidad Simón Bolívar, Barranquilla, Colombia

E-mail: tvelasquezp@ufpso.edu.co, hnavarroch@ufpso.edu.co

Abstract. Many of the research carried out on green information technology or green technologies has focused on reducing the environmental effect, on the design, development and use of increasingly eco-efficient electronic devices, in terms of the machine, that is, the hardware; but this trend has changed in recent years, due to the concern to cover all fronts in which green information technology is part; what has allowed to generate studies and investigations in another aspect, as it is the software. Some authors have addressed the issue, however, models, descriptions or implementations in the area of software are still in short supply, whose main rationale is eco-efficiency, that is, environmental strategies to reduce the impact of a product or service by increasing the efficiency of resource utilization. The research emphasizes the formulation of a guide based on the principles promulgated in green information technology as a non-functional requirement in the analysis and design of software and incorporating this guide in the software engineering projects of the systems engineering program of the Universidad Francisco de Paula Santander, Seccional Ocaña, through the content of the green information technology elective course and as an academic unit of the software engineering course.

1. Introduction

We live in a time when all actions within companies must be governed by increasing behaviors in the care of the environment, so that aspects such as energy shortages, industrial waste management, global warming or the greenhouse effect product of the bad management of the emission of gases that is reflected in the high temperatures and many other phenomena that if not controlled, will increasingly be reflected in the deterioration of the development of human life. Some authors make an interpretation of this problem referred to in the green information technology report: technologies for energy efficiency in information technology systems. In this report they emphasize that care must be a priority for governments as well as for companies and in general for the whole society [1].

In the analysis, design, development and implementation of information technology (IT), processes and activities that allow the efficient use of computational resources to minimize environmental impact to ensure social duties in the conservation and preservation of information must be taken into account.

Any IT initiative that you want to develop and implement, must be treated through a project, that is, following a schedule, a budget, complying with some phases that deliver products with a high quality and efficiency index. These initiatives developed under the management of a project are based on methodologies and models guides such as Control objectives for information and related technologies

(COBIT), extreme programming (XP), Integration of capacity maturity models (CMMI), processes unified by rational (RUP), agile and flexible methodology to manage software development among others (SCRUM), and architectures that have been evolving in response to the needs of companies; all of them with difficulties such as the granularity of the processes versus the types of projects with clear differences between engineering and management, with the due complexity of them [1].

One of these methodologies that has been traditional for the development of software is RUP, which integrates elements and processes of other methodologies and that for being one of the most used and traditional in our environment, compared to others such as XP and SCRUM that are agile methodologies, makes the processes extend more than what is due, which allows imperceptible environmental impact effects that together affect our environment. These elements, plus the suggestion for adoption in integrating into the planning and design of projects, processes of conservation of environmental aspects, make the developers keep their eyes on each time using methodologies that facilitate the work. It is necessary then to appropriate this trend in the development of software and implement actions and elements that allow what we do daily, be applied and not only preached.

2. Background

The general concept of green computing was raised [2] in the first chapter of the book "green in software engineering", where they detail the importance that sustainability, impact, for countries around the world have had environmental and pollutant emissions from industrial processes with the use and promotion of environmentally sound technologies. The book presents an overview of the aspects of sustainability in software engineering and defines its units from requirements analysis to aspects of green software measurement. Regarding software development methodologies and models focused on the requirements of green computing, [3] defined a two-level ecological software development model that covers the stages of the software life cycle. On the first level, they proposed a new green software engineering process that is a hybrid process between sequential, iterative and agile development processes to produce an environmentally sustainable one. Each stage of the software process is studied further to produce a sustainable ecological stage. They proposed green guidelines or green processes for each stage of software in the engineering process. They added to the software life cycle the requirements stage and the test stage. They also included in the first level, a complete list of metrics to measure the greenery of each stage in terms of the first-order effects of ICT in the environment for a green software engineering process. The second level explains how the software itself can be used as a tool to help with green computing by monitoring energy resources efficiently. Finally, they showed and explained the relationships that can be found between the two levels of the proposed model to make the software and product engineering process ecological and sustainable [4] also addressed this issue and designed greensoft model, a conceptual reference model for green and sustainable software, whose goal is to support software developers, administrators and software users in the creation, maintenance and use of software in a more sustainable way. The greensoft model has a comprehensive life cycle model of software products, sustainability criteria and software product metrics, procedural models for different stakeholders, and recommendations for action, as well as tools that support stakeholders in the development, acquisition, supply of, and the use of software in an ecological and sustainable way. Other authors [5] in their research “Development of energy efficient products: Models, technology-supported methods” proposed a computer system for product development that integrates product structures, digital product models and methods that incorporate concern for energy efficiency. The product development process is based on a property-driven design approach that is adapted considering an energy efficiency design. The objective of energy efficiency is oriented towards design as well as the calculation of costs that can be controlled. The integration of the product model includes information on energy consumption that can be achieved through an ontological system. The proposed computer system combines an enterprise application architecture with workflow and management modeling to support product development processes, taking into account energy efficiency. This vision is detailed in Figure 1 [5].

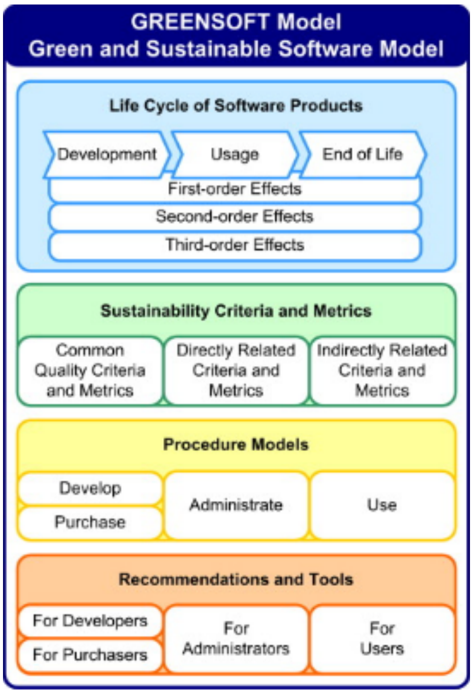
GREENSOFT Model
Green and Sustainable Software Model
Life Cycle of Software Products
Development
Usage
End of Life
First-order Effects
Second-order Effects
Third-order Effects
Sustainability Criteria and Metrics
Common Quality Criteria and Metrics
Directly Related Criteria and Metrics
Indirectly Related Criteria and Metrics
Procedure Models
Develop
Purchase
Administrate
Use
Recommendations and Tools
For Developers
For Purchasers
For Administrators
For Users

Figure 1. GreenSoft model [4].

Other authors focused on doing studies and research on models with reusable processes to improve efficiency [6], who determined that the low efficiency in the use of software engineering best practices and the poor quality of the products developed are two of the main reasons for the failure of software projects. In this work developed [6], an architectural model is described, based on the reuse of process assets and projects supported by collaboration mechanisms, which improves the efficiency of the use of software processes, and a set of quality parameters in the validation process.

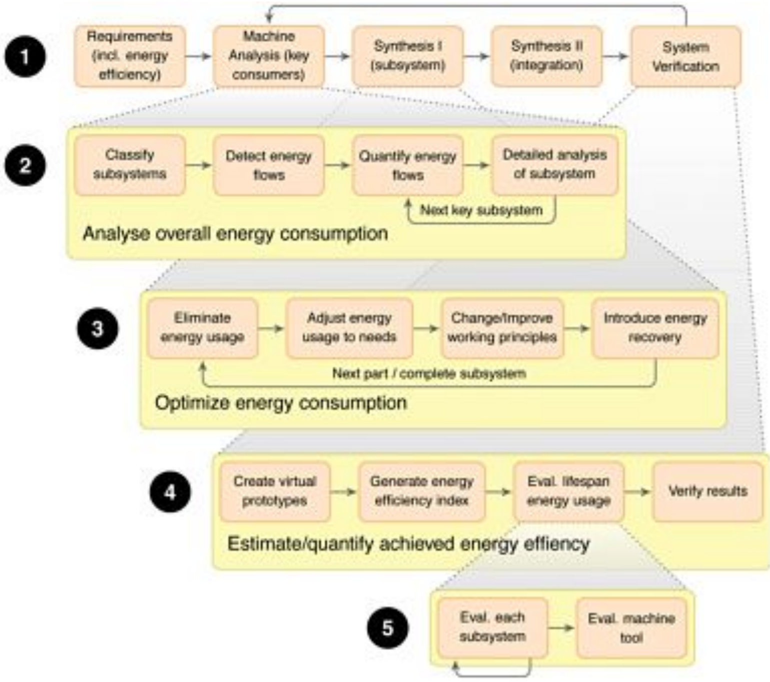
1
Requirements (incl. energy efficiency)
Machine Analysis (key consumers)
Synthesis I (subsystem)
Synthesis II (integration)
System Verification
2
Classify subsystems
Detect energy flows
Quantify energy flows
Detailed analysis of subsystem
Next key subsystem
Analyse overall energy consumption
3
Eliminate energy usage
Adjust energy usage to needs
Change/improve working principles
Introduce energy recovery
Next part / complete subsystem
Optimize energy consumption
4
Create virtual prototypes
Generate energy efficiency index
Eval. lifespan energy usage
Verify results
Estimate/quantify achieved energy effiency
5
Eval. each subsystem
Eval. machine tool

Figure 2. Model of energy production processes [5].

This model was implemented in real software projects, see Figure 2. Now, what happens if the efficiency and focus of green computing is associated with the code developed, what does it mean to design algorithms or write code in a way that incorporates physical costs into energy consumption [7] refer to the cost of using algorithms in programming and in solutions that affect energy consumption [8]. They addressed this premise with their work “Introduction to special issue on theoretical aspects of green computing”, said that the algorithmic problems involved in development solutions directly involve energy management. All the use that is given to energy in our time is based on information and communication technologies. As Dougherty also, [9] addressed this approach, with his work "The self-scaling in cloud computing and the algorithmic problems in the development of energy consumption", proposed in model-driven auto-scaling of green cloud computing infrastructure, where it concludes that cloud computing can reduce energy consumption by using virtualized computing resources to provision an application of computational resources on demand. Self-scaling is an important cloud computing technique that dynamically locates computer resources to applications to accurately match their concurrent loads, thus reducing resources that would otherwise remain inactive and waste energy. This approach is detailed in Figure 3.

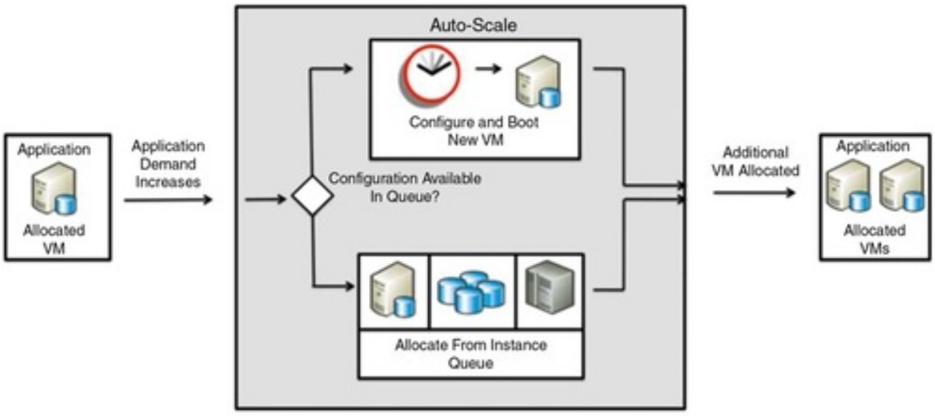
Application
Allocated VM
Application Demand Increases
Auto-Scale
Configure and Boot New VM
Configuration Available In Queue?
Allocate From Instance Queue
Additional VM Allocated
Application
Allocated VMs

Figure 3. Model-driven auto-scaling of green cloud computing infrastructure [5].

After investigations related to the models, the code and the storage processes were related [10], established relations of the software development with the metrics used and defined a methodology for the measurement of the energy efficiency of the software in abstraction layers and application development environments.

63 Open source code applications were used as a base, the results of which indicate that when integrating into framework development environments and external libraries where more robust applications are built, there is more energy efficiency than in smaller software developments.

Also found that by using different types of functional applications in software developments, different levels of energy efficiency are presented, compared to text and image editing programs and applications and the use of software. For games that allow greater power consumption due to the indiscriminate use of the processor, these measurements are detailed in Table 1 and Figure 4. Other authors who addressed this issue were Nasir Rashid [11] in the document "Developing green and sustainable software using agile methods in global software development: Risk factors for vendors" but focused on the supplier's risk factors "in which they conclude that the More pressing risks are insufficient documentation of the system, little real-time support for large projects, high administrative costs, poor presence and little knowledge of the client, little formal communication, limited support for reuse, and poor planning to long term. And finally, within the investigation carried out, we took into account the works and research of F. Ahmed [12], where they argue that everything related to green IT should be incorporated into the undergraduate courses in software requirements engineering and discuss the green

IT best practices for various software engineering activities, to increase knowledge and understanding of IT environmental sustainability and highlight opportunities for academia to incorporate these software engineering concepts with green IT.

Table 1. Effect of functional type on energy efficiency.

Functional type	Effect on average EE
Text_edit	0.308453
Erp	0.308732
Image_edit	0.511010
Game	0.619239
Web	0.677822
Ftp	0.712511
Calc	0.867131
Calendar	0.942454

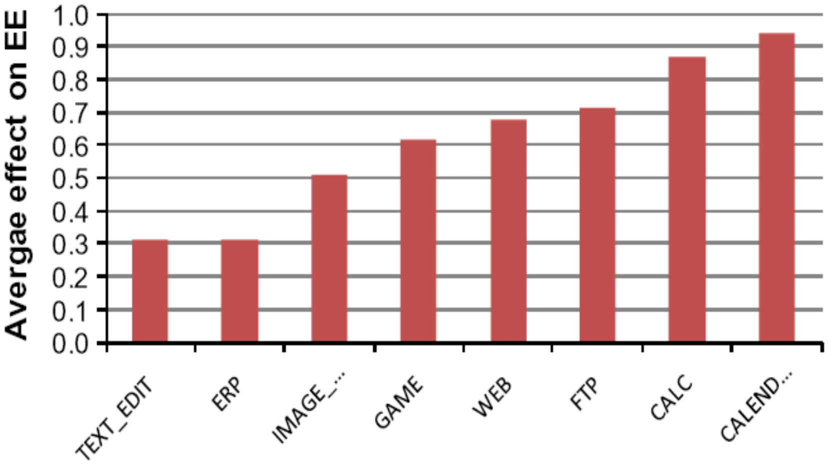

Figure 4. Effect of functional type on energy efficiency [10].

3. Methodology

The type of research will be descriptive, since it seeks to identify green IT principles adaptable to the Software development methodologies and their subsequent formulation. The focus of it will be qualitative. It will be done in the following steps:

- Construction of the state of the art to identify the stages of the “Greensoft model” and “A green model for sustainable software engineering” models with the green IT references in Information and communication technologies.
- Explore and analyze in the stages of analysis and design of the most common software development methodology, subsystems, interfaces, components and persistence processes.
- Design a guide with the integration of green IT elements and principles in the analysis and design stages of the main existing methodologies.
- Incorporate the proposed guide into the software development projects at Universidad Francisco de Paula Santander.

4. Contextualization

The research will be addressed in the systems engineering program of the Universidad Francisco de Paula Santander, Seccional Ocaña.

5. Results

Formulation of guide components according to the elements found in scientific articles and the comparison with existing methodologies. A detailed analysis of the processes and stages of the main existing methodologies including RUP, XP and SCRUM was made with the elements, definitions and conclusions of the authors in the different facets studied and then incorporated into the proposed guide.

With the research carried out, the green IT subject curriculum was designed, which is currently part of the systems engineering program's curriculum of the "Universidad Francisco de Paula Santander, Seccional Ocaña".

Formulation of a guide based on green IT as a non-functional requirement in the analysis and design of software. A guide was developed jointly with students of the course in which the pre-software development criteria, criteria for the software development stages and finally the software post-development criteria are highlighted.

Propose the incorporation of green IT elements, concepts and principles and the guide proposed as a unit of the software engineering course of the systems engineering program.

The incorporation of a unit of study of the subjects of analysis and design of systems and software engineering that will contain the initiatives of green IT focused on the development of the projects will be proposed before the curriculum of systems engineering of the Universidad Francisco de Paula Santander software and the incorporation of the green IT Guide as a non-functional requirement in the analysis and design of Software.

Integrate as a non-functional requirement the proposed guideline within the methodologies oriented in the systems engineering program of the "Universidad Francisco de Paula Santander, Seccional Ocaña". Propose the incorporation of the green IT guide as a non-functional requirement in the analysis and design of software in the practical and theoretical practical subjects of the Systems Engineering program

Formulate research proposals on the theme of green IT in Research projects of the IT Governance line of the GITYD research group of the "Universidad Francisco de Paula Santander, Seccional Ocaña".

Through the subjects of the systems engineering program, the specialization in systems audit, the master of government of IT and the GITYD research group, the inclusion of talks and seminars on the theme of green IT in the Software will be proposed.

6. Conclusions

With the formulation and implementation of the guide based on the principles of green IT and the subsequent incorporation of said guide as a non-functional requirement in the analysis and design of software applications and the reformulation of the contents of the software engineering subjects; Teachers, students and future engineers will have the necessary elements to generate and develop quality software, increasingly benevolent to environmental sustainability and the planning and administration of IT infrastructure according to the requirements of society in terms of maximizing resources generating lower environmental impacts and reducing the energy footprint caused by the use of IT, as a significant contribution of the University in software development best practices.

References

[1] López-Vallejo M, Huedo-Cuesta E and Garbajosa-Sopeña J 2008 *Informe de vigilancia tecnológica madri+d Green IT: tecnologías para la eficiencia energética en los sistemas TI* (Madrid: Fundación madri+d para el Conocimiento)

[2] Calero Coral and Piattini Mario 2015 *Green in Software Engineering* (Nueva York: Springer Publishing Company)

[3] Mahmoud Sara S, and Ahmad Imtiaz 2013 A green model for sustainable software engineering *International Journal of Software Engineering and its Applications* **7** 55-74

[4] Naumann Stefan, Dick Markus, Kern Eva, and Johann Timo 2011 The GREENSOFT Model: A reference model for green and sustainable software and its engineering *Sustainble Computing: Informatics and Systems* **1** 294-304

[5] Rünger G, Götze U, Putz M, Bierer A, Lorenz S, Reichel T, Steger D, Wenzel K, and Xu H 2011 Development of energy-efficient products: Models, methods and IT support *CIRP Journal of Manufacturing Science and Technology* **4** 216-224

[6] Sanchez-Segura M-I, Medina-Dominguez F, de Amescua A, and Mora-Soto A 2010 Improving the efficiency of use of software engineering practices using product patterns *Information Science* **180** 2721-2742

[7] Choi J W, and Vuduc R W 2013 How much (execution) time and energy does my algorithm cost? *XRDS Crossroads ACM Magazine for Students Scientific Computing* **19(3)** 49-51

[8] Pruhs K 2011 Introduction to special issue on theoretical aspects of green computing *Sustainble Computing: Informatics and Systems* **1(3)** 165-166

[9] Dougherty Brian, White Jules, and Schmidt Douglas C 2012 Model-driven auto-scaling of green cloud computing infrastructure *Future Generation Computer Systems* **28** 371-378

[10] Capra Eugenio, Francalanci Chiara, and Slaughter Sandra A 2012 Is software “green”? Application development environments and energy efficiency in open source applications *Information and Software Technology* **54** 60-71

[11] Rashid N y Khan S 2016 Developing green and sustainable software using agile methods in global software development: risk factors for vendors *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering ENASE 2016* (Rome) (New York: AMC) p. 247-53

[12] Ahmed F y Shuaib K 2012 Incorporating green IT concepts in undergraduate software requirements engineering course: An experience report *7th Iberian Conference on Information Systems and Technologies CISTI 2012* (Madrid) (Madrid: Universidad Politecnica de Madrid) p. 1-4