



HAL
open science

Eco-Conscious Software Development: A Comprehensive Guide for Sustainable Practices

Ryan Vernex, Rébecca Deneckère

► To cite this version:

Ryan Vernex, Rébecca Deneckère. Eco-Conscious Software Development: A Comprehensive Guide for Sustainable Practices. International Baltic Conference on Digital Business and Intelligent Systems (Baltic DB&IS), University of Vilnius, Jun 2024, Vilnius, Lithuania. 10.1007/978-3-031-63543-4_2 . hal-04703849

HAL Id: hal-04703849

<https://hal.science/hal-04703849v1>

Submitted on 20 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Eco-Conscious Software Development: A Comprehensive Guide for Sustainable Practices

Ryan Vernex and Rébecca Deneckère

Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, Paris, France
ryanvernex@yahoo.fr , denecker@univ-paris1.fr

Abstract. According to the World Health Organization, climate change is identified as a significant threat, primarily driven by excessive greenhouse gas emissions. While the natural greenhouse effect is essential for maintaining a habitable planet, human activities contribute to an overproduction of these gases, leading to accelerated global climate warming. In response to this, the concept of Green IT emerges, encompassing Information and Communication Technologies that mitigates environmental harm. Although Green IT is commonly associated with hardware, its software aspect plays a critical role in controlling energy consumption. This work try to answer to this challenge by offering a guide to improve the development process with a sustainable approach through a literature review and interviews of several professionals.

Keywords: Eco-design; Development; Guide; Sustainability; Sustainable policy.

1 Introduction

According to the World Health Organization (WHO), climate change is one of the main threats of the century. The primary actor in this phenomenon of climate disruption is the excessive emission of greenhouse gases. Even if the greenhouse effect is essential for the planet to be habitable for humans, human activity tends to excessively increase their production. The immediate consequence is an excessive absorption of heat, inevitably leading to global climate warming. According to [1], Information and Communication Technologies (ICT) accounted for 4% of the global greenhouse gas emissions in 2020. This report also predicts an increase to 14% by 2040. It is in this context that the concept of Green IT makes sense. There is a strong relationship between ICT and sustainability [2] and Green IT refers to "all ICT whose use helps reduce the harmful effects of human activity on the environment." Today, Green IT is primarily recognized for its positive impacts on infrastructure and hardware [3].

However, the software aspect represents a crucial domain because it controls the use of hardware and, consequently, the energy consumption of devices. Thus, [4] emphasizes the importance of developing products that manage their consumption. There are a lot of works addressing this problem. For instance, [5] addresses the topic of ecology in software engineering. This work primarily focuses on metrics and measurement methods for the consumption of web services applications. Their research objective is to propose a solution that can guide architectural decisions for a project

utilizing web services. Similarly, [6] propose a ranking of programming languages based on their ecological characteristics. It also attempts to establish a connection between energy consumption, runtime, and memory usage. Thus, the language with the best energy performance is not necessarily the one that executes the fastest. [2] proposes a literature review focalized on the past ten years research that treat the topic of sustainability and ICT. This work emphasized the need to have a holistic approach to sustainability in ICT. This context leads us to the following research question: ***How to guide the application development process towards a sustainable approach?***

The research methodology is explained in Section 2 and Section 3 analyses the results. An eco-design guide is described in section 4 and a description of the evaluation is made on Section 5. We conclude in section 6.

2 Research Methodology

Our research adopted a design science approach to develop a sustainability guide for software development. We followed the seven guidelines preconized by [7].

- Guideline 1 : Design as an artefact. Our objective was to define a guide to lead the development process towards a more sustainable approach.
- Guideline 2: Problem relevance. Sustainability is a contemporary challenge, and there are numerous works related to this issue in the context of software development.
- Guideline 3: Design evaluation. Our initial evaluation involved expert feedback, and we are planning a more comprehensive assessment to validate and refine the proposed guide.
- Guideline 4: Research contributions. We are proposing a guide for sustainability in software development. We have identified three levels of involvement for sustainability and structured our findings into these three levels.
- Guideline 5: Research rigor. To ensure rigor, we conducted a thorough review of existing literature to define the guide's elements. Additionally, we conducted interviews to validate and enhance the guide's effectiveness. Details of our protocol methods are elaborated further in this section.
- Guideline 6: Design as a search process. The creation of our guide involved synthesizing pertinent information from the literature review and insights provided by domain experts.
- Guideline 7 : Communication of research. The resulting guide provides a holistic view accessible to stakeholders involved at different levels of sustainability engagement.

We followed the process illustrated in the figure 1. Each step is described in the rest of the section.

Define Research Question. Our main research question is *How to guide the application development process towards a sustainable approach?* To this end, we studied how to measure the usage of resources in ICT solutions (what resources, what

measurements, which tools), what solutions currently exist to limit the usage of resources in ICT solutions, and we propose a guide summarizing our findings.

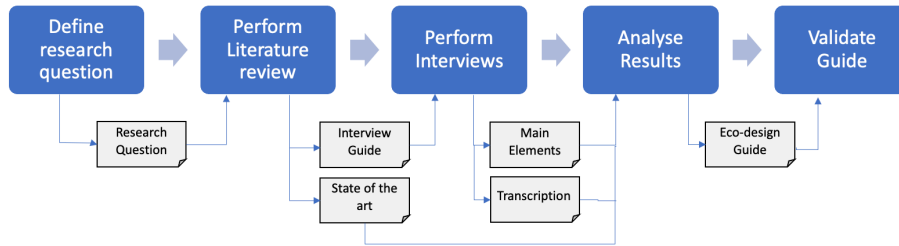


Fig 1: Research Process

Perform Literature Review. The literature review allowed us to identify existing tools for measuring resource consumption and then analyze eco-design methods that have already been applied. We used the Scopus API to search through the database and obtain the papers corresponding to the following set of inclusion criteria:

- (1) use of specific keywords like “energy” and “consumption” and “software development”, or “green” and “software” and “process” with the use of two keywords strings : TITLE-ABS-KEY(energy) AND TITLE-ABS-KEY(consumption) AND TITLE-ABS-KEY(software development) AND PUBYEAR > 2021 AND (SUBJAREA(COMP) OR SUBJAREA(ENGI)), and TITLE-ABS-KEY(green) AND TITLE-ABS-KEY(software) AND (TITLE-ABS-KEY(process) OR TITLE-ABS-KEY(practice)) AND PUBYEAR > 2021 AND (SUBJAREA(COMP) OR SUBJAREA(ENGI));
- (2) the paper proposes methods or tools for the development process with a sustainability axis;
- (3) the paper is a primary research;
- (4) the paper is in English.

Perform Interviews. In order to support and challenge the results obtained in the state of the art, we confronted them with the opinions of professionals. This method brings a crucial qualitative aspect to the proposal of our guide. We chose participants who work daily in the context of Green IT and the profiles differ based on their various professions. Table 1 shows a quick presentation of each participant.

The interviews aimed to gather as much information and added value as possible to enrich the research. In this context, we conducted semi-structured interviews [8]. Each interview was constructed around three parts.

- Context Reminder: First and foremost, it seemed essential to put the interlocutor in or back into the context. Thus, we systematically explained the issue, the outline, but above all, the goal of the research. This brief presentation of the context allowed the interviewees to better position themselves and guide their responses to connect with the context.
- Presenter Presentation: Subsequently, the interlocutors introduce themselves, their role, and what they think they can contribute during this exchange.

- Rebound Phase: Once the interlocutor is introduced, we ask the predefined questions. Based on the responses, we can follow up on specific topics by seeking more details or by directly posing new questions. For instance, some of the questions were the following: what are the most relevant resource consumption measures in the context of a development project? What measurement tool(s) have you already used? What is your definition of eco-design? What eco-design method/process have you already experimented with? How can it be improved?

Table 1. Interviewees.

	Role	Missions
1	Green IT Manager at BTI - Advisory	- Support clients (B2B context) in their Green IT initiatives.
2	Sustainable IT Analyst at Accenture	- Compile a set of tools related to responsible digital practices, focusing on open-source and free software. - responsible for supporting companies in the evolution of their eco-responsible initiatives.
3	Full-Stack Developer at CarbonScore.	- Develop a B2B SaaS tool that enables all employees of the client company to track real-time indicators such as CO2 emissions or water consumption.
4	Consultant	- Engage in the development of companies' responsible digital initiatives.

All the interviews had been carefully recorded and transcribed by hand.

Analyze Results. We studied the interviewees responses to answer the research question and elaborate our guide to lead the development process by respecting sustainability requirements. We ensured inter-rater reliability by cross-referencing remarks from the interviewees and retaining only those that aligned across multiple responses. Our principle was to give greater weight to remarks that were consistently mentioned by multiple interviewees, as they were deemed more significant in our analysis.

Validate Guide. Following the exploration of the ideas highlighted in the first round of interviews, we developed our proposal of a guide. It was therefore relevant to seek validation for this work. Semi-structured interviews also allow us to build on the responses and address new questions. This approach adds fluidity to the exchange and, most importantly, enables us to extract the maximum value that the participants can provide. Here are the different steps of this exchange with the same interviewees as in the first round:

- Reminder of the context: Several months had passed since the initial exchange with each of the interviewees, so we had to reintroduce the research context and the general problematic.
- Presentation of the research work: We provided a detailed presentation of the eco-design guide. During this presentation, we strongly encouraged our interlocutors to interrupt us if they had any questions or misunderstandings. It allowed us to rephrase or better elaborate on certain points.
- Questioning/critique phase: Once the presentation was complete, we gathered our interlocutor's feedback on the proposal. We also invite them, if necessary, to

suggest other research angles that we may not have addressed but could be interesting.

3 Measurement of Energy Resource Consumption and Potential Solutions

The literature review and initial round of interviews emphasized the significance of precise resource consumption measurement in software development, particularly in defining the measures used and recommending the tools for these measurements. Furthermore, the state of the art also presents a range of solutions aiming at reducing this consumption and two of the interviewees insisted on the use of a specific business model aiming at improving the sustainability aspects.

3.1 Measure the Resources Consumption

Information and communication technologies generate significant resource consumption throughout their life cycle. In recent years, technological innovation, coupled with increasingly intense marketing strategies and a continuously growing demand, has massively contributed to a rapid turnover of electrical/electronic equipment [9].

Measure types. In this section, we will identify and analyze relevant measures in the context of software development projects. We will take the opportunity to explain the impact of each of these measures to understand how to exploit them to the fullest. We classified the measures used in the literature, following the typology proposed in [3]. There are 6 categories of measures: (a) related to component consumption (hard drive, screen), (b) related to code analysis, (c) related on runtime, (d) based on estimates, (e) related to the development process, and (f) others. On this classification we can select three measures directly linked to the development part: (b), (c), and (e).

- Measures related to code analysis. These measures consider several indicators, as:
 - the number of executed instructions,
 - the number of direct sub-classes, or inheritance richness,
 - the lack of cohesion on methods: Number of pairs of functions with shared instance variables subtracted from the total number of pairs of functions without shared instance variables, and
 - the weighted methods per class: sum of complexities of methods defined in a class [10].
- Measures on runtime. The indicators used to evaluate these measures are:
 - the energy consumption during execution, often expressed in kilowatt-hours (kWh) [11],
 - the number of network exchanges: it means requests and responses; reducing unnecessary data exchanges can contribute to better energy efficiency,

- the refresh rate¹: if too high, this can lead to overuse of the CPU and memory, and
- the response time: reducing response times improves the overall efficiency of the application and can potentially result in lower energy consumption [12].
- Measures related to the development process. They are taking three indicators into account:
 - The number of iterations required to complete the development of a feature/module. A too high number would imply greater resource consumption and thus indicate a potential issue in the design.
 - The level of documentation (quantity and quality). A rich documentation can enhance the overall productivity of a development team. However, unnecessary, redundant or overly detailed documentation is time consuming so should be avoided too.
 - The automated testing structure. A good test coverage ensures the longevity of the product and enables greater efficiency.

Measuring tools. Today, more and more tools are emerging in the IT environment to help measure resource consumption. All interviewees emphasized that the chain of responsibility includes all stakeholders of the software, from users to management. We can then categorize the tools based on the scope they cover, that is, the scale at which the tool can be exploited: either at the user level, the development level, or the management level, as shown in Figure 2.

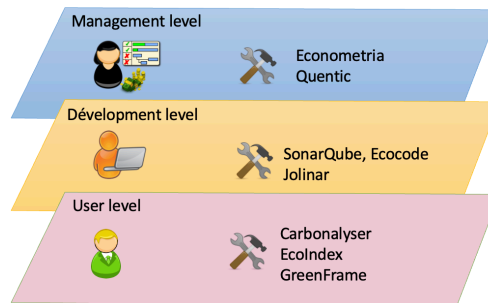


Fig 2: Measuring Tools Examples

User level. Let's start at the lowest scale, at the level of awareness of an individual user. Several tools exist, directly addressed to the user. There is the Firefox browser extension named Carbonalyser². This tool provides real-time information about the electrical consumption and associated greenhouse gas emissions of your internet browsing. This type of tool primarily aims to provide transparency to users of a product to raise awareness about its associated consumption. On the same way, EcoIndex³ calculates the environmental performance of a site and its environmental footprint

¹ <https://www.makeuseof.com/test-monitor-refresh-rate-best-sites/>

² <https://theshiftproject.org/en/carbonalyser-browser-extension/>

³ <https://www.ecoindex.fr/>

(weight of greenhouse gas emissions and water consumption generated by the page) [10]. Other tools rely on a measurement system that takes into account user scenarios, like GreenFrame⁴.

Development level. Many measurement tools are accessible to application designers with the aim of better managing the ecological footprint of the developed systems [13]. Sonar⁵ has developed the SonarQube tool to detect clean code structures. Bad code may have a negative ecological impact so it is important to identify the cleanliness of the used code. Plugins can also be added to SonarQube, as Ecocode, a collective project aiming to reduce environmental footprint of software at code level. Based on a list of static code analysers it will highlight code structures that may have a negative ecological impact. [14] proposes Jolinar, a tool for measuring the energy consumption of an IT system. This tool is accessible to technical profiles (developers, architects) but also to less technical profiles through an intuitive interface. [14] shows that the choice of using an algorithm in an application can be crucial for the product's energy consumption.

Management level. The entire responsibility should not rest solely on the developers. Indeed, it is important that project managers, often higher-ups, are aware of this aspect and make decisions accordingly. Tools exist also for this level of responsibility, such as Econometria⁶, Quentic⁷, etc.

It is important to note that all the tools and methods mentioned in this section represent only a tiny portion of what exists in the market today. In summary, we can see that the ecological impact of any IT system is everyone's concern, and each person can contribute to the overall improvement of the process.

3.2 Solutions to Limit Consumption

In the literature, various types of solutions are available for addressing this issue. These solutions can be broadly categorized into technical solutions and functional solutions. To enhance readability and clarity, technical solutions are further divided into two parts based on their general purpose: those aimed at developing software with a primary goal of limiting resource consumption, and those designed without a primary focus on Green IT principles.

Technical solutions for software whose primary goal is resource consumption limitation. Limiting resource consumption is crucial for sustainable development and environmental protection. By efficiently managing resources, we reduce waste, promote environmental sustainability, and contribute to economic resilience. This approach aligns with the principles of a circular economy, emphasizing the reuse and recycling of materials to minimize environmental impact. Recognizing the importance

⁴ <https://greenframe.io/>

⁵ <https://www.sonarsource.com/products/sonarcloud/>

⁶ <https://ecometrica.com/>

⁷ <https://www.quentic.com/>

of resource consumption limitation is vital for maintaining a balance between human development and environmental well-being. We can find several solutions of this type in the literature, such as Power API, SmartWatts, JoularJX, etc.

Power API is a tool developed by the University of Lille. It is intended to serve as a toolbox for measuring the power consumption of a program. Applications built using PowerAPI are referred to as PowerMeters [15]. A PowerMeter can then measure the power consumed by a machine or a group of machines. This calculation can be done either in real-time or retrospectively once the monitored software is no longer in use. This type of tool can be very relevant in a Cloud context for managing the power allocated to the containers used. Indeed, by measuring and analyzing real-time consumption, it is possible to adapt configurations such as the CPU of a container. The SmartWatts software also proposes a concept of this type [15].

JoularJX⁸ is a tool that allows tracking the evolution of energy consumption by method. The real added value here is the granularity of the process. It is interesting for the developer to know which method is responsible for a potential excess consumption.

Technical solutions for software not primarily designed with a Green IT perspective. Here, it is interesting to explore how the usage of such software can be modeled to achieve Green benefits. Let's take the example of Kubernetes⁹. It is an open-source container orchestration system widely used to automate the deployment, scaling, and management of containerized applications. Its primary function is the management of deployments and their configurations. However, Kubernetes also offers the possibility of configuring automatic scaling of its clusters. This feature allows optimizing resource utilization without compromising application performance.

Functional solutions. There are functional/organizational solutions that can be used in a software. For instance, the "Software Carbon Intensity (SCI)" methodology is proposed¹⁰. It is based on energy consumption, carbon intensity, and additional carbon emissions related to the amount of carbon emitted during the construction and recycling of the hardware used. The goal of this measurement is to provide a relevant indicator of the eco-design level of a product. The SCI then helps make decisions regarding the system design of an application to directly influence related carbon emissions.

3.3 Sustainable Business Models

The development of sustainable business models represents an innovative element of business strategy. Various industries and business categories have employed these models to concurrently address their economic, environmental, and social objectives. There is an increasing amount of works on sustainable business model (a simple search on the Scopus database with the keyword "sustainable business model" shows around 1000 results only for the last 4 years. [16] proposes four main types of solutions: designing a sustainable value proposition, designing sustainable value creation, designing

⁸ <https://www.nouredine.org/articles/joularjx-2-a-leap-forward-in-green-software-analysis>

⁹ <https://kubernetes.io/>

¹⁰ <https://sci.greensoftware.foundation/>

sustainable value delivering, and generating sustainable partnership networks for creating and delivering such sustainable value which can meet the social, environmental, and economic benefits at the same time.

[17] highlight four types of business models that depend on the context: Sustainable Software Solution, Sustainable Product-Service System, Sustainability intelligence, and Digital Sustainable Platform.

- *Sustainable Software Solution*. This involves selling software with features that can impact the client's resource consumption. This model is most commonly found in a B2B context and corresponds to what is called “IT for green” solutions. Let’s look into the following example: a company currently offers a range of training and advice on various ways to save water consumption. Here, the business model could evolve by introducing software that allows customers to measure the evolution of their water consumption. In doing so, this company takes advantage of securing and retaining its customer base. It is the same kind of business model utilized by CarbonScore, the company presented by our 3rd interviewee.
- *Sustainable Product-Service System*. This model combines the sale of products and services by offering sustainable options for both. A simple example of this business model is the rental of a product rather than its sale. Often, the company takes advantage of this to sell additional IT services related to the product. For instance, A company sells phones with interchangeable components (as the Fairphone¹¹ company). Here, a service related to the product is its maintenance, often involving simply replacing one of the product's components rather than the entire device.
- *Sustainability intelligence*. These solutions integrate data analytics with sustainability techniques to offer transparency in companies' ecological sustainability assessments. This kind of solution helps to support sustainable decision-making and identify new sustainability improvement opportunities.
- *Digital Sustainable Platform*. This involves the establishment of a circular exchange platform, often aimed at individuals. This platform can be used to address specific environmental issues and help promote a circular economy. The Too Good To Go platform is a very good example of a circular economic platform.

However, as highlighted by our interviewees, it is important to be wary of potential negative rebound effects from this type of platform.

4 Eco-Conscious Software Development Guide

In this section, we present a guide that draws its logic from both the analyzed state of the art and the interviews conducted with professionals in the field. Sustainability in the development process impact three aspects : the business model, the development and the management. The highlighted idea here is the necessity of combining these three different aspects to effectively establish sustainable development within a company. It is essential to note that, in the context of initiating responsible digital transformation, it

¹¹ <https://www.fairphone.com/>

is possible to have implemented only two or even just one of these aspects. For instance, the transformation can start with choices made by the management and subsequently evolve over time to complete implementation and business model aspects.

4.1 The Importance of a Sustainable Business Model

We highlighted the importance of implementing a sustainable business model in section 3. A business model outlines how value is created, delivered to customers, and captured by the company. It serves as a mediating framework that aligns overarching business strategies with operational processes, playing a crucial role in achieving a competitive advantage that directly influences a company's performance [17].

Sustainable business models influences all the stakeholders, as shown in figure 3.

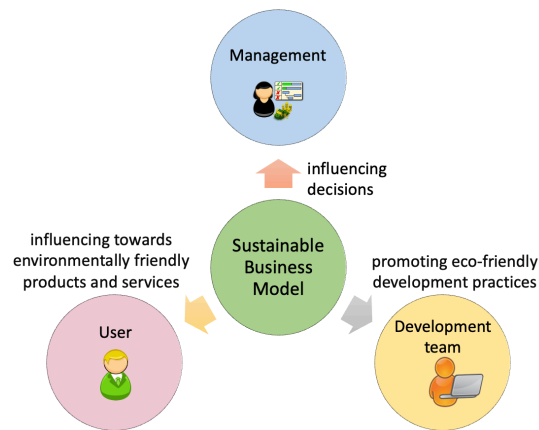


Fig 3: Influences of a Sustainable Business Model

The holistic approach depicted in this figure helps illustrate how the various aspects of eco-design interconnect to create a sustainable ecosystem in the development and use of products and services. Sustainable business models influence the decisions made by the management; they also influence the development team by promoting development practices which satisfy sustainable requirements; but they also influence the users in their choices of sustainable products and services.

4.2 Impact Sensitive Management

Management must also consider sustainability. By applying the company's sustainable policy, certain impact requirements must be considered throughout the development process.

We illustrate the management activities to consider these impacts by using the MAP formalism [18] to show the high level of abstraction of the process (see figure 4). The management has to satisfy two main intentions: to define and to validate the impact requirements.

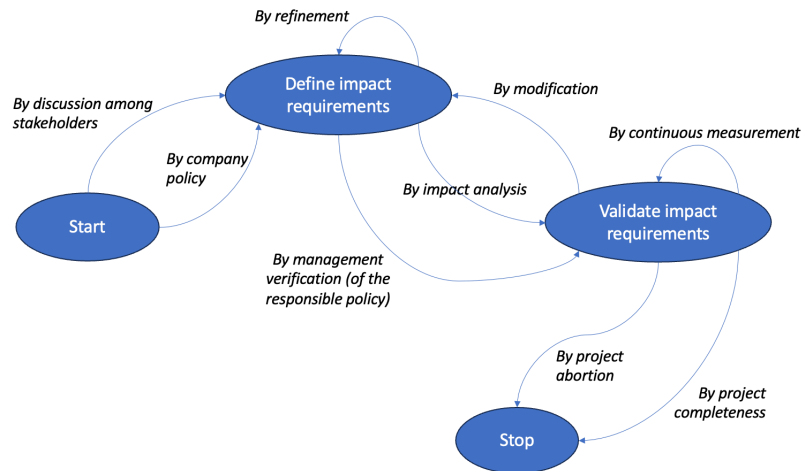


Fig 4 : Impacts Requirements Management Process Model

Define impacts requirements.

By discussion among stakeholders. The manager's role is to bring together all project stakeholders. This step is necessary to inform and, more importantly, clarify the objectives inherent to the project with each stakeholder. Indeed, before the implementation begins, it is crucial that all stakeholders align on the same goals. To achieve this, it is relevant for the manager to organize dedicated meetings for these discussions. During these exchanges, managers of different involved teams discuss the possibilities of implementing the solution based on the resources and skills of their respective teams.

By company policy. A sustainability or responsible policy outlines an organization's commitment to integrating environmentally friendly, socially responsible, and ethical practices into its operations. It includes reducing environmental impact, promoting fair labor practices, transparent governance, sustainable supply chains, circular economy principles, and innovation for environmentally friendly products. This policy reflects a long-term commitment to social and environmental responsibility, meeting the expectations of stakeholders, including consumers, investors, employees, and society.

By modification. If the requirements are not completely validated, it is possible to update the project and the impact requirements. In this case, the management will need to undergo another impact analysis to ensure that the changes align with the policy.

Validate impact requirements. During the project framing, the manager is responsible for ensuring that it adheres to the eco-design guidelines established by their leadership and aligned with the sustainable business model.

By management verification. The project validation is not subject to an impact analysis. This scenario can occur when the project in question is directly aligned with the goal of improving the environmental footprint of the company.

By impact analysis. The manager can delegate team members to conduct an impact analysis. This task may resemble a traditional feasibility study for a project.

By continuous measurement. As soon as the implementation of the solution begins, the manager must ensure the monitoring of environmental impact indicators. It involves verifying predefined indicators, especially during the impact analysis. To ensure the relevance of these measures throughout the project, it is necessary to repeat them periodically. For example, in a project following monthly delivery cycles, it may be relevant to measure the impacts before each production release.

Stop. There are two ways to stop this process. Either the project is aborted (*By project abortion*) or everything has been going well and the impact requirements have all been validated correctly for this project (*By project completeness*).

4.3 Approaches for Eco-conscious Development

In this section, we address best practices and processes to be applied by members of the development team. We present two different processes based on the context: either the solution is completely new or it is a solution already existing but requiring a redesign to enhance sustainability.

Redesign of an existing solution. Here, the goal is to propose a relevant process in the context of redesigning an application solution already in use within a company. The objective is to maintain the application, add new features, and potentially decommission other functionalities. The first step is to list all existing functionalities of the system. Then, for each functionality, an assessment is needed with the aim of making a decision. The following decision tree (figure 5) serves as a guide for this decision-making process.

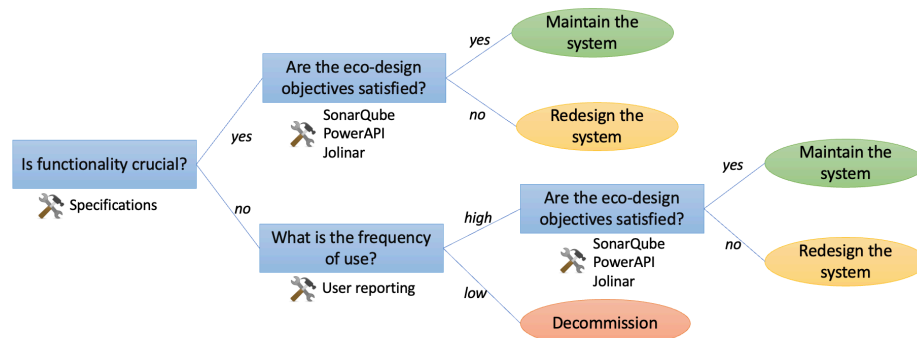


Fig 5 : Decision tree on the functionalities

This diagram includes some tools mentioned earlier to illustrate some tool possibilities. The use of this tree is valuable for classifying needs for each part of the project (here separated into functionalities).

Implementation of a new solution. This involves a thorough analysis of the stated need. The comprehensive analysis of the specifications should enable the establishment of various implementation scenarios. These scenarios may differ, particularly in the choices of application design. It may, therefore, be relevant to conduct feasibility studies upfront based on the various implementation possibilities. Table 2 shows that there are three possible decisions for choosing the implementation of a feature, depending on the importance of the functionality and its environmental impact. The feature can either be automatically implemented or discontinued, but it can also be studied more in-depth.

Table 2. Decision matrix for functionalities.

Functionality importance	Bad impact	Average impact	Good impact
Facultative	To phase out	To phase out	To study deeper
Useful	To phase out	To study deeper	To implement
Crucial	To study deeper	To implement	To implement

For each feasibility study, we establish the list of features that matched with the matrix above. Subsequently, the results of each study can be compared, and a decision can be made on the most interesting combination of functionalities. To make this decision, it is necessary to present the various possible scenarios to end-users. Involving users in the design process helps facilitate the decision-making on the scenario to implement. At the same time, it provides users with a better understanding of development choices, especially regarding features that are not feasible.

4.4 A Multi-level Approach for Improved Sustainability

We can illustrate the proposed guide with a multi-level approach, as shown in figure 6. This figure shows the interactions between the different levels and how the diverse activities can impact the artefacts and the next activities. Moving from top to bottom, we have the business model, management, and implementation levels.

We start from the left at the business model level. The initial and arguably most crucial step is critically analyzing the existing business model within the company to recognize opportunities for improvement and alignment with one of the patterns discussed above. Once we've chosen the pattern that best fits, we initiate the transition toward this new business model. This marks the descent into the management domain.

The leadership team articulates the new model through the initial projects stemming from it. Then, the task then falls to the management, to select the team, manage implementation resources, and breathe life into the first project.

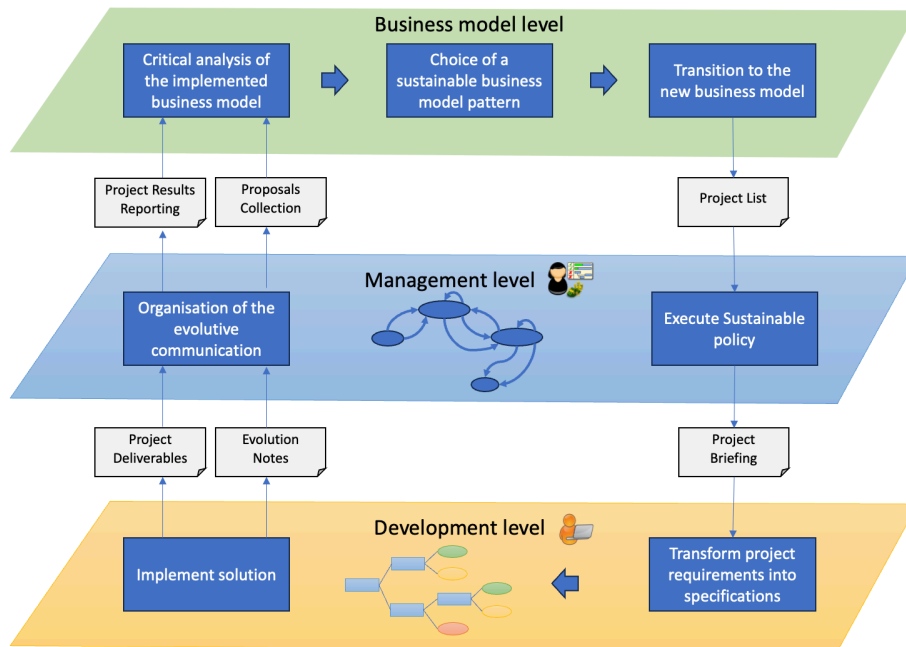


Fig 6 : Interactions between the Different Levels

The implementation team's primary mission is, of course, to develop the solution. However, it extends beyond that, initiating a bottom-up logic. Implementation teams may possess insights that those at higher levels, whether in management or leadership, might not have. Therefore, it's valuable to gather their opinions and ideas to potentially evolve the business model.

It becomes the management's responsibility to orchestrate this evolving communication, ensuring it travels upward to the highest level—back to the business model—and then engaging in discussions for continuous improvement. This process ultimately forms a loop that continues.

In addition to this summary diagram highlighting the interactions between the three levels, it is interesting to detail the content and purpose of each of the artifacts:

- The project list: Once a business model pattern is selected, the management begins to establish projects that address the needs arising from the new BM. The projects are then allocated among the different management teams.
- The project briefing: For each project, the manager must carry out a document . including various elements: Identification of different stakeholders, Distribution of responsibilities among people, Definition of project objectives, Identification of the risks, and Definition of all project deliverables to be provided.
- Project deliverables: This refers to all the documents to be provided at the end of the project. Numerous elements may be included depending on the project context:

Documentation, Product user manual, Specifications, Functional prototype, and so on.

- Evolution notes: At the end of the project, the implementation team produces a document summarizing all the remarks or improvement ideas submitted by team members.
- Project results reporting: Following project delivery, the management team needs to collect the results. This involves success indicators (KPIs). A report can then be generated based on a predefined time period (3 months, 6 months, 1 year).
- Proposals collection: This involves translating the previously established evolution notes by the implementation team. Managers must first assess the feasibility of the ideas submitted. They can also support these ideas by providing resource plans able of translating these ideas into projects.

5 Evaluation of the Guide

We validated the guide through another round of interviews to gather the experts' perspectives on the proposed guidelines.

This allowed us to validate certain elements that had not been discussed with all respondents during the initial round of interviews. For example, the impact of having a specific business model was only addressed by two of our interviewees, but this proposal was appreciated by all during the evaluation. Similarly, the process recommended for management was refined to align with the feedback from our respondents.

All interviewees endorsed the multi-level figure proposal, emphasizing the importance of having this holistic view of the problem and highlighting the relevance of the proposed cycle.

6 Conclusion

The aim of this study was to provide a guide to lead the development process towards a more sustainable approach.

We looked into the literature in order to define the key elements to a sustainable development, interviewed four experts, and presented our guide, the result of our study. This guide is based on exchanges between the three layers: business model, management, and implementation. It stresses about the importance to use a sustainable business model to guide a sustainable approach. It looks more closely on the managing process of impacts requirements at the management level. It offers guidelines in the implementation of a solution, if the main goal is to transform an existing solution into a more sustainable one or if it is to implement a brand new solution.

To continue this work, we first plan to execute a more complete evaluation of the proposed guide to complete our study. We then plan to conduct a systematic literature review to extract best practices in this domain (for instance, does open-source collaboration or the use of agile methodologies offers a more sustainable approach to the development?). These practices will be linked to the guide, enhancing its capability to provide more effective guidance within the context of eco-design.

References

1. TheShiftProject : Déployer la sobriété numérique, Rapport TheShiftProject (2020)
2. Hassmann, T., Westner W. Conceptualizing sustainability in the context of ICT. A literature review analysis. Conference on Computer Science and Intelligence Systems (FedCSIS 2023), Warsaw, Poland (2023).
3. Ergasheva, S. , Khomyakov, I. , Kruglov, A. , Succi, G.: Metrics of energy consumption in software systems: a systematic literature review, IOP Conference Series: Earth and Environmental Science (2019)
4. Lami, G., Fabbrini, F., Fusani, M.: A methodology to derive sustainability indicators for software development projects”, International Conference on Software and System Process (2013)
5. Kiesl, F.: Metrics for Measuring Greenability in Web-based Software Systems, Johannes Kepler University Linz (2021)
6. Pereira, M., Ribeiro, F., Saraiva, J. : Towards a green Ranking for programming Languages, Universidade do Minho (2017)
7. Van der Merwe, A., Gerber, A., Smuts, H. Guidelines for conducting design science research in information systems. Annual Conference of the Southern African Computer Lecturers' Association. Cham: Springer International Publishing (2019).
8. Edwards, R., & Holland, J. (What is qualitative interviewing? (p. 128). Bloomsbury Academic (2013)
9. Wang, F., Kuehr, R., Ahlquist, D., Li, J.: E-waste in china, a Country report, United Nations University (2013)
10. Watson, A. H., Wallace, D. R., & McCabe, T. J. Structured testing: A testing methodology using the cyclomatic complexity metric (Vol. 500, No. 235). US Department of Commerce, Technology Administration, National Institute of Standards and Technology (1996)
11. Singh, V. K., Dutta, K., VanderMeer, D.: Estimating the Energy Consumption of Executing Software Processes," 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, pp. 94-101 (2013)
12. Sun, X., Zhang, Q., Medina, M.A. , Liu, Y. , Liao, S. : A study on the use of phase change materials (PCMs) in combination with a natural cold source for space cooling in telecommunications base stations (TBSs) in China, Appl. Energy 117, 95–103 (2014)
13. Josephson, C., Peill-Moelter, N., Pan, Z., Pfaff, B., & Firoiu, V. The Sky is not the limit: untapped opportunities for Green Computing. ACM SIGENERGY Energy Informatics Review, 3(3), 33-39 (2023)
14. Noureddine, A., Islam, S. , Bashroush, R.: Jolinar: Analysing the Energy Footprint of Software Applications (Demo), The International Symposium on Software Testing and Analysis (2016)
15. Fieni, G., Rouvoy, R., & Seinturier, L. Smartwatts: Self-calibrating software-defined power meter for containers. In 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID) (pp. 479-488). IEEE (2020)
16. Nosratabadi, S., Mosavi, A., Shamshirband, S., Zavadskas, E. K., Rakotonirainy, A., & Chau, K. W. Sustainable business models: A review. *Sustainability*, 11(6) (2019).
17. Böttcher, T. P., Empelmann, S., Weking, J., Hein, A., & Kremer, H. Digital sustainable business models: Using digital technology to integrate ecological sustainability into the core of business models. *Information Systems Journal* (2023).
18. Rolland, C., Prakash, N., Benjamin, A., A Multi-Model View of Process Modelling, Requirements Engineering, Springer-Verlag London Ltd., vol. 4, n° 4 (1999)