Life-cycle Assessment in Software Engineering
(Position paper)

Matthias Galster
mgalster@ucalgary.ca

Abstract

Life-cycle assessment, as applied to conventional product development, can be applied in the Software Engineering domain to determine the environmental impact of a software product itself, but also the environmental impact of a software development process.

1. Introduction

In the context of developing conventional products (physical items), the Integrated Product Policy (IPP) seeks to minimize environmental degradation by looking at the phases of a product life-cycle, and taking actions where most effective [1]. Such a life-cycle usually includes the extraction of natural resources, product design, manufacturing, assembly, marketing, distribution, sale and the disposal. Life-cycle assessment (LCA) provides a quantitative and scientific basis for IPP. LCA assesses environmental aspects associated with a product over its entire life-cycle.

A product life-cycle as outlined above covers aspects similar to what we find in software development processes or software product life-cycles. Thus, the concepts of IPP and LCA might also be applied in a Software Engineering (SE) context.

2. LCA in Software Engineering

The work proposed in this position paper not only aims at treating sustainability as first-class (non-functional) product requirement, but also considers the process of developing the software product. In SE, sustainability and environmental impact can be related to

1) a software product,
2) a software development process (process for one particular product or the software development process applied in an organization), and
3) the whole software product life-cycle (development, usage, maintenance, etc.).

Applying LCA in SE would help address these three areas of concern. To achieve this, we can adapt existing LCA tools, such as SimaPro[1] or GaBi[2]. Those tools usually require inputs and outputs from the production line to assess environmental impact. Often, these tools use assumptions (or less specific "background" data) and more solid facts (more specific "foreground" data). The benefit of these tools is that they include various methods for impact assessment. For example, in the Eco-indicator method [2] available in SimaPro, the environmental damage caused by agricultural practices is determined by empirical data from botanists. However, these tools and the implemented methods do not support types of resources or data relevant in SE. Moreover, such tools require detailed modeling of life-cycles or development processes. Thus, the goal of the presented research is twofold: 1) Provide means for modeling software development processes from a sustainability perspective. 2) Gathering, validation and documentation SE relevant sustainability data. Besides fundamental empirical studies, online resources[3] provide useful information for gathering such data.

An environmental impact analysis in SE (based on LCA) could include three basic steps: 1) *Classification* sorts all resources into classes based on their effect on the environment (e.g., resources that contribute to the greenhouse effect). *Characterization* aggregates resources within each class to produce an effect score. As some resources may have a more intense effect than others, we would need to apply weighting factors to the different resources. 2) To gain a better understanding of the relative size of an effect, a *normalization* step is required. Each effect calculated for a product / process is benchmarked against the known total effect for this class. Normalization allows us to see the relative contribution compared to each already existing effect. 3) Not all effects are equally important. In the *evaluation* phase, the normalized effect scores are multiplied by a *weighting* factor representing the relative importance of the effect.

---
[1] www.pre.nl/simapro
[2] www.gabi-software.com
[3] see www.pre.nl/LCAsearch or www.life-cycle.org

3. Example application of LCA in SE

In the following, we present a fictitious example for applying a small portion of LCA to an SE problem. The goal is to evaluate two software design alternatives with regard to their environmental damage impact. We focus on three damage indicators: the damage to human health (HH), the damage to ecosystem quality (EQ), and the damage to the stock of exploitable energy resources (ER). The theoretical foundations of the applied method are discussed in [3]. We used the Triangle Tool[4], a simple decision support tool to illustrate the weighting of different environmental effects when comparing alternatives.

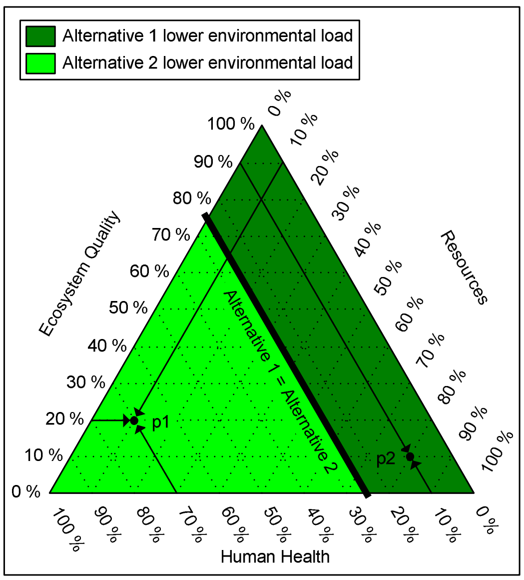

Figure 1 Comparison based on Triangle Tool

The tool uses the Eco-indicator 99 method, a damage-oriented method for life-cycle impact assessment [2, 4]. Thus, the tool requires three normalized damage scores for HH, EQ and ER. Such scores can be calculated with tools, e.g., SimaPro. We used fictitious scores for illustration purposes as input for the Triangle Tool[5].

Figure 1 shows a comparison of two design alternatives. The bold line indicates the line of indifference. p1 indicates where HH is weighted 70%, ER 10% and EQ 20%. p2 indicates where ER is weighted 0%, HH 10% and EQ 90%. This means, Alternative 2 should be chosen if the weighting is based on p1 (i.e., HH is most important, ER is least important) and Alternative 1 should be chosen if the weighting is based on p2 (EQ is most important). Eventually, this approach, as part of a LCA in the context of SE, would help evaluate design choices with regard to their environmental damage potential.

[4] www.pre.nl/eco-indicator99/download_triangle.htm

[5] Also, current LCA tools make it difficult to calculate damage indicators for software products.

4. Potential benefits

Potential benefits of applying LCA in SE are:

1) As sustainability is competing with other (non-) functional requirements and software project constraints, investigating the ecological impact of software and development processes helps make a more informed decision about a) design decisions, and b) the modeling of software development processes and life-cycles. For example, as stated on www.life-cycle.org, Microsoft found that by replacing traditional distribution of software by DVD/CD disk kit with electronic distribution means, a 91% reduction in carbon emissions from a life-cycle perspective can be achieved.
2) As found in the context medical equipment, IPP and LCA not only help reduce adverse environmental aspects and carbon footprint, but, at the same time, improve business performance [5]. Thus, IPP might not only act as driver for product innovation but also for software process improvement.
3) Software process simulation could be extended towards the environmental impact of processes (e.g., to investigate the environmental impact of different development paradigms). Also, new predictor models could enable predictions about product sustainability based on software project properties.

5. References

[1] European Commission. (2009). Environment - Integrated Product Policy. [Online]. Available: http://ec.europa.eu/environment/ipp/.

[2] M. Goedkoop and R. Spriensma. (2000). The Eco-indicator 99 - A damage oriented method for Life Cycle Impact Assessment (Methodology Report). [Online]. Available: http://simapro.rmit.edu.au/LIT/LCA/EI99_METHODOLOGY_V2.PDF.

[3] P. Hofstetter, A. Braunschweig, T. Mettier, R. Mueller-Wenk, and O. Tietje, "The Mixing Triangle: Correlation and Graphical Decision Support for LCA-based Comparisons," *Journal of Industrial Ecology*, vol. 3, pp. 97-115, October 1999.

[4] The Netherlands Ministry of Housing and Spatial Planning and the Environment. (2000). Eco-indicator 99 - Manual for Designers. [Online]. Available: http://www.pre.nl/download/EI99_Manual.pdf.

[5] Siemens Medical Solutions. (2007). Ecological Product Information for Diagnostics Imaging Devices. [Online]. Available: http://www.eup-ecodesign.com/files/Siemens%20Medical%20Solutions.pdf.