“Good Design is Environmentally-Friendly.” – Discussion of Rams‘ Principle in Context of the Software Life Cycle

Sven Gerber¹, Sandra Gerber², Peter Krüger³, Hans-Knud Arndt¹

Abstract

Dieter Rams publicized a statement that good design of products is environmentally friendly. This paper discusses software, as an intangible product, with environment and terms related to environment. To measure software in context of environment we discuss two accepted methods and show the shortage of these methods in connection with software. We also mention the analogy between software life cycle and general product life cycle. By considering all the aspects, we reasoned that ecological design of software is inherently covered by non-functional requirements which take up product characteristics concerning the whole software life cycle. Finally we discuss if Rams’ statement related to nowadays understanding of environment and propose a revised statement.

1. Introduction

During his work as designer and leader of the design department at the Braun Corporation, Dieter Rams defined the “Ten Principles for Good Design” (Rams 1995; Godau 2003). These principles claim general rules for designing products, and also the environmental aspect is mentioned by these “good design is environmentally-friendly” (Vitsoe 2012). So these theses were created by his work designing tangible products. In the modern age, is it reasonable to implement these principles for intangible products like software, too. A huge amount of software applications will be published every day, not to mention updates or add-ons. In our perception there is a gap of proof validity of Rams’ thesis “good design is environmentally-friendly” in consideration to intangible products, especially software products. In our paper we will discuss how this principle could be implemented for intangible products. So the two keywords ‘sustainability’ and ‘design’ come to the fore: sustainability enclose amongst others the big topic environment-friendliness.

And what is about design? “Today, everybody knows the phrase design, but the perceptions about the meaning of design disagree remarkable.” (Godau 2003) At the end of 16th century were the term design mentioned at the first time. In the Oxford English Dictionary the term was defined as “a scheme conceived by man for something that should be implemented. Secondly its meaning as a drawn concept for an artwork, or object from applied art which is mandatory for execution of this creation.” (Godau 2003)

In 1996, Steve Jobs explained in an interview what design is and what not: "Some people think, design means how it looks. But of course, if you dig deeper, it's really how it works. To design something really well, you have to 'get it'. You really grok what it's all about.” (Young and Simon 2005)

¹ Otto-von-Guericke-University Magdeburg – Faculty of Computer Science, email: {sgerber | hans-knud.arndt}@iti.cs.uni-magdeburg.de
² Eudemonia Solutions AG, sandra.gerber@eudemonia-solutions.de
³ Volkswagen AG, email: peter.krueger6@volkswagen.de
⁴ For instance, Braun creates and produces till today electric shaver or household appliance.
Design is not only product development, design affects the whole product, the product affects the way of use, the way of use affects the interpersonal relations. Hence, design is not an artistically or technical phenomena, instead it is a sociocultural one. (Hammer 2008)

In newer literary sources it is obvious that the mindset about design has changed. The success of new product’s design is based on proper arrangement of the technical feasibilities, the desirability for users, and the viability for the company. (Russo et al. 2009) Companies develop and apply methods like design engineering, design thinking, or early prototyping. All methods respect product’s life cycle as well as the understanding to design a product as an essential step in product’s life cycle.

Environmental friendliness is up to the relation between organism and their natural environment in a positive meaning. In context of processes or products it means a non-destructive impact of these to the environment. In opposite to environmental friendliness represents environmental compatibility the reducing of negative environmental impact. Due to that, environmental friendliness aims beyond achieving norms and critical values (Kotulla 1995).

The term environmental friendliness implies the meaning of ecological. This depicts a connection to sustainability. Sustainable development means the search for possibilities, to “meet the needs and aspirations of the present without compromising the ability to meet those of the future.” (Brundtland 1987) To find solutions you have to check economical, ecological und social aspects of organizations. Sustainability reports are one tool for organizations to show their efforts for sustainable development and behavior. With the GRI Guidelines (Global Reporting Initiative 2011) and “The Ten Principles” of Global Compact (Brundtland 1987) there are two frameworks available to create useful reports. In these reports the main focus usually lies on tangible items. For them it is relatively easy to measure amounts. Another framework for getting environmentally characteristic numbers is to make a life cycle assessment corresponding to DIN EN ISO standard 14040 (DIN Deutsches Institut für Normung e.V. 2006a).

On balance there is an inherent relationship between design and environment but a big challenge to denominate this relationship for intangible products like software. Inspired by Rams’ principles, we try to grab the connection for software in this article.

2. Software Life Cycle

Software is an intangible product and can be defined as “computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.” (The Institute of Electrical and Electronics Engineers 1990) Software products, or technical products in general, have a life cycle, and accordingly an end of application. (Schatten et al. 2010; Herrmann 2009)

The OXFORD DICTIONARY defines the term life cycle as “the series of changes in the life of an organism including reproduction.” In relation to software it is explained as “a system for managing software throughout its life cycle”. (Oxford University Press) Technical systems can be considered in life phases and life cycles. Life phase’s consideration means including views on material flow and energetically significant resources. A well-known metaphor for this concept is “from cradle to grave”. Life cycles concept includes also thinking in phases but here is the output of utilization in focus. Regarding the utilization over the whole product life results typically in a Gaussian distribution curve, also known as classical product life cycle. (Herrmann 2009)

The software life cycle can be defined as “the period of time that begins when a software product is conceived and ends when the software is no longer available for use.” Depending from focused paradigm some development phases may overlap or be performed iteratively. (The Institute of Electrical and Electronics Engineers 1990) The software development cycle is part of the whole software life cycle and represents the development stage (see Table 1). It is the period of time that begins with the decision to develop a software product and ends when the software is delivered. This term is sometimes synonymously
used for the time a developer takes care about the software product (The Institute of Electrical and Electronics Engineers 1990).

Classical concepts for software development processes are the waterfall model (Brandt-Pook/Kollmeier 2008), spiral model (Kossiakoff/Sweet 2005), or the V-model (Goll 2011; Brandt-Pook/Kollmeier 2008; Dumke 2003). New development paradigms have the classical models extended or shifted. The paradigms give ideas for organizing the development stage. Incremental approaches like prototyping (Kossiakoff/Sweet 2005), Rational Unified Process (Goll 2011), Scrum (Cohn 2010), or extreme programming (Brandt-Pook/Kollmeier 2008) have common objectives: i.e. agile and flexible implementing of changes, to shorten development duration, bringing in results faster.

### Table 1
Comparison of Life Cycles

<table>
<thead>
<tr>
<th>Software Life Cycle (Schatten et al. 2010)</th>
<th>Stage</th>
<th>General Product Life Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Requirements and specification</td>
<td></td>
<td>(Pfeiffer/Bischoff 1981)</td>
</tr>
<tr>
<td>- Planning</td>
<td></td>
<td>- Evaluation and selection of alternatives</td>
</tr>
<tr>
<td>- Draft and Design</td>
<td></td>
<td>- Research</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Prototype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Preparation for Production and Distribution</td>
</tr>
<tr>
<td></td>
<td>Development-Stage</td>
<td>(Fritz/Oelsnitz 2001; Herrmann 2009)</td>
</tr>
<tr>
<td>- Implementation and Integration</td>
<td>Market-Stage</td>
<td>- Market Introduction</td>
</tr>
<tr>
<td>- Operating and Maintenance</td>
<td></td>
<td>- Growth</td>
</tr>
<tr>
<td>- Uninstalling</td>
<td>Removal-Stage</td>
<td>- Maturity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Saturation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Decline</td>
</tr>
<tr>
<td></td>
<td>(Fritz/Oelsnitz 2001)</td>
<td>- Recycling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elimination</td>
</tr>
</tbody>
</table>

3. **Methods for Measurement**

We present two possible methods for gathering facts and figures of an environmentally-friendly product development to enable correct decisions along life cycles. The first method, life cycle costing, provides the collecting and analysis of costs during the life cycle, the second one is for the analysis of environment aspects during the life cycle, too. Both methods can theoretical use for intangible products, but it is hard to find practical examples for this.

3.1 **Life Cycle Costing**

The detection of life cycle costing is defined as a process of the economic analysis of estimating the whole costs of purchasing, ownership and disposal of a product. The process could execute for the whole life cycle of a product, sub product or particular phase of life cycle. The aim of the detection is to get input data
for decisions in some or all phase of life cycle of a product (DIN Deutsches Institut für Normung e.V. 2005).

Without going in depth at this point, for detecting of life cycle costs the following steps have to been executed: cost breakdown structure, product/work breakdown structure, choosing cost categories and cost elements, estimation of costs, representing the results and if adaptive especially environment aspects. Further information and application notes contain the standard DIN EN 60300-3-3.

The standard also point out that all decisions for a product regarding to design, production, the using and so on include environmental impacts and affect the cost of ownership and disposal.

3.1.1 Cost Estimation

Cost estimation bases on assumptions and is thus fault-prone, but empirical values accomplish transparency and knowledge for coming projects. Assumptions are mostly prepared at the beginning of a project. The accuracy of estimation grows with the progress of the project, especially when per iteration a new estimation workshop will be done and the remaining effort will be estimated. There are two principles to estimate (Oestereich/Weiss 2008):

- In the detail-principle, also called micro-estimation, single components of an object will be estimated.
- In the re-use-principle the analogies of experiences from other projects are the background. The estimator has to take care in using the last principle: experiences from earlier projects are not for all parts of the project available and estimations are related to teams or individuals.

The selection of the following methods can be applied for cost estimation in software engineering (Oestereich/Weiss 2008):

- Analogy-/relation-methods: like lines of code, function value, story points etc. The function value method will be detailed later.
- Weighting-method: with this method the total estimation results from the sum of the weighted single-estimations.
- Multiplication-method: there is an equation between the costs of the project and project complexity to the monthly staff member costs and the monthly staff member output.
- Percentage rate method: the estimation per phase and discipline will be determined.

3.1.2 Function Value Analysis Method

A method for measuring the volume of completed software artifacts is the analysis of function’s value. Function value analysis is a methodic procedure to provide functions in demanded quality, low costs, and reliability. (Zarnekow 2007) The value is the “relationship between positive and negative consequences (output and input, or benefits and sacrifices)”. It can be seen as assessment of an object. The objective view on value is seen as relationship between benefit and expense (Thomson et al. 2003). DELL’ISOLA expresses the objective value in the formula (Dell'Isola 1997), whereby quality is here considered as satisfying customer needs, and a function of a product is represented in the form of to do something:

\[
\text{Value} = \frac{\text{Function} + \text{Quality}}{\text{Costs}}
\]

Value analysis involves the testing of functions required by customers as design objectives. (Thomson et al. 2003) The function value analysis method described by ZARNEKOW includes three major steps:

1. Identifying functions of a product or service
2. Assigning a monetary value to the function
3. Producing the function by respecting demanded quality and minimal total cost.
Principle of this method is thinking in functions instead of thinking in costs. That means, the output or benefit of a product is focused, and not costs to build it. The value corresponds to the lowest costs which have to be spending that the product affords the defined functions. (Zarnekow 2007) To sum up, this method provides an approach to analyze and evaluate software products.

3.2 Life Cycle Assessment

To rate the environmental-friendliness of a creation process, organizations can use life cycle assessment. This method is used to make material flows comprehensible and to evaluate them in the whole life cycle of a process or service.

The life cycle assessment is divided into four parts, which are connected to each other. The first part is the definition of scope and goal of the study. Hereby, attention should be paid on the fact that the selected boundaries have impact on the level of environmental-friendliness, because of the possibility to hide critical processes. In part two the quantitative material flows will be collected in the form of a balance. This part is called life cycle inventory. The difficulty in this phase is the collection of values in practice. Often it isn’t possible to measure exact values for one process or product, because of technical restrictions or economic reasons. The collected and balanced values will be used in the third part to perform a life cycle impact assessment. There were several procedures developed to rate the environmental effect of emission, for example CML-01 from the Netherlands, a method of ecological shortage. Finally in fourth part data from life cycle inventory and life cycle impact assessment will be rated in consideration of selected scope. (DIN Deutsches Institut für Normung e.V. 2006b)

4. Discussion

In section 1 the term environmental friendliness was exposed. A quantitative assessment of environmental friendliness is hard to achieve, because of lack of available metrics. One possible treatment is the measurement of sustainability, for instance with life cycle assessment. At this juncture the life cycle assessment refers nor to environmental friendliness, nor to environmental compatibility, but rather to ecology as one column of sustainability. Ecology just regards the aspect if something happens in accordance with nature. An evaluation for fulfillment of standard or compliance with safety values will not conducted. The life cycle assessment gives no advice about the level of environmental friendliness of a product. Thus, there is only a weak connection to Rams’ principle “Good design is environmentally-friendly.” A shortcoming of Rams’ principle is his explicit relation to the term “environmentally friendly”. Following the previous derivation of terms, it isn’t possible for a product to achieve this requirement. Instead, a solution could be to rephrase his thesis to “Good Design is environmentally compatible.” Environmental compatibility is relative and has point of reference. References may be standards, safety values, or ethical behavior.

All analysis methods of life cycle assessment have an initial point for considering products and their environment. The origin of each product is the input of raw materials. These raw materials are tangible and measurable. In contrast, developing intangible products as result of intellectual work have not a direct application of raw materials. This difficulty entails all four parts of the life cycle assessment. The system boundaries can be defined individually, e.g. including data center, excluding power generation, including production of computer parts or not, so there is no consistent standard, which environment is mandatory to get a significant result from life cycle assessment. For example, raw materials are used to generate electricity, but an accurate mapping which percentage of power is used for a certain software product cannot be achieved. Too far is the progress of system integration or concepts like XaaS (everything as a service, i.e. software as a service). In these concepts is the IT infrastructure outsourced and therefore IT infrastruc-
ture is not directly available for measurements. In fact, we reason the application of life cycle assessment is limited and has to be adapted for intangible products to get a significant statement.

We mentioned the method life cycle costing, too, because it is an informative method to estimate costs in the life cycle of tangible products. Developing software is in creative process and is mostly individual production. Stakeholder and requirements are changing from application to application. These individual factors complicate estimating costs. Empirical values give reliable advise if exist many results for the estimated object. For individual production, like software production, the objects vary and therefore estimating is less precise. This method for estimating costs is only limited capable to give any information about sustainable of a product. Rather it is in indicator for distribution of costs concerned to a product.

In section one, we have already shown that design is more than the external appearance. Design has to be considered on different tiers of a product and can also be affected on these tiers. In context of Rams’ principle the relation between design and environmental friendliness could not sufficient resolved. The production, utilization, and removal of products imply also an encroachment of the environment (Triebel 1997). In consequence, products cannot be declared as environmentally friendly. Instead the summation of environmentally compatible product characteristics is seen as ecologically. The use of the term “ecological” shall suggest environmental friendliness but has the meaning of environmental compatibility. (Triebel 1997)

In (Triebel 1997) are nine principles of ecological design presented. These principles refer to tangible industrial products, too. A few principles of those can be applied to parts of software life cycle (see Table 2). In fact, it is remarkable that the mentioned ecological principles are covered by non-functional requirements in software development (see International Organization for Standardization 2011). That means, for software products already exist criteria which are similar to design principles of tangible products and therefore also have an inherently ecological purpose.

<table>
<thead>
<tr>
<th>Chosen Ecological Principles of Design with Relevance to Software (according to TRIEBEL)</th>
<th>Non-functional Requirements according to ISO 25010 (Product Quality Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reducing Power Consumption in all Phases of Life Cycle</td>
<td>Performance Efficiency</td>
</tr>
<tr>
<td>Stable Principles of Construction</td>
<td>Reliability</td>
</tr>
<tr>
<td>Modular Design</td>
<td>Contained in Maintainability</td>
</tr>
<tr>
<td>Design which is easy for Repair and Maintenance</td>
<td>Maintainability</td>
</tr>
<tr>
<td>High Comfort for Handling and Use</td>
<td>Usability</td>
</tr>
<tr>
<td>Design which is easy for Disassembly</td>
<td>Contained in Portability</td>
</tr>
</tbody>
</table>
5. Summary

In this paper we have elicited the relation between Ram’s principle of environmentally friendly product design and environmental compatibility, sustainability, and ecology. We referred to software life cycle and tried to show the software life cycle in comparison to a general, tangible product’s life cycle. Then we gave an overview about methods to measure general products; the life cycle assessment and the life cycle costing. By discussion the terms environmental friendliness and environmental compatibility in section four, we suggest rephrasing Ram’s principle to “Good design is environmentally compatible.” In our opinion this message points out better the today’s notion of an ecological product. Furthermore, we discussed the shortcoming of life cycle assessment as method to measure environmental compatibility. Also the method life cycle costing provide not enough information to measure the environment. By closer examination of the ecological design principles, we reasoned that these principles nowadays are comprised in software products, too. Those aren’t named explicitly as ecological software design principles, but inherit ecological aspects and are mentioned as non-functional requirements in the product quality model of the standard ISO 25010.

Bibliography
