A Generic Model for Sustainability with Process- and Product-specific Instances

Birgit Penzenstadler and Henning Femmer
Institut für Informatik
Technische Universität München, Germany
{penzenst|femmer}@in.tum.de

Abstract

Motivation: Software systems as we know them often have an economic purpose and/or fulfill human or social needs of their users. The economic purpose is analysed by economy itself; the latter goals are analysed in software engineering by user-centric techniques, such as service orientation. Yet, as software systems have an impact on the environment, environmental sustainability should be supported as a major goal for software development projects.

Problem: Without applicable guidance, sustainability remains an untangible ideal. Therefore, we need a definition and a concrete decomposition of sustainability to relate it to software systems development. It is not sufficient to analyse environmental sustainability on its own, but its interplay with other aspects in order to define appropriate actions and understand their effects.

Principal idea: We analyse the dimensions of sustainability, their values with respective indicators, and activities to support them. These elements compose a conceptual model that allows for analysing and constructing actions both for a company or a product point of view.

Contribution: We propose a generic sustainability model with instances for companies and projects from various case studies. We thus enable analysis, support and assessment of environmental sustainability in software engineering.

Categories and Subject Descriptors D.2.1 [Requirements Engineering]: Elicitation methods

Keywords sustainability, definition, requirements, generic reference model, case study

1. Introduction

Sustainability has been recognized as a relevant topic in software engineering, e.g., by the 2012 ICSE theme and a number of workshops. It refers to environmental, social, and economic aspects of software development and the usage of software systems. Most software systems usually are developed with an economic benefit in mind which shall also be preserved over a longer period of time. In case of software systems that are not intended to serve economic purposes, for example, open source software, the main purpose of the software system is a social benefit which can make specific work tasks easier or just plain entertainment and having fun. Taking into account that the economic and social aspects of sustainability are already analysed in most software systems, our work especially focusses on supporting the environmental perspective. Our proposed approach helps to integrate the objective of environmental sustainability in software systems as part of the requirements engineering. Hilty et al. [8] provide an analysis of the relevance of information and communication technologies for environmental sustainability and conclude that there is no such thing as a “general ICT policy for environmental sustainability” [5]. For an adequate analysis, we need a tangible decomposition of the concept of sustainability and supporting methods in software engineering. These methods can then enable to include the concept of sustainability in software engineering and help to develop such a general ICT policy for environmental sustainability.

Problem Currently, there is little guidance on how software engineering can contribute to improving the sustainability of the systems under development. Thus, sustainability is not tangible enough as a concept to actually transform it into software requirements. There is no approach available describing how to decompose sustainability for a concrete project and we are missing procedures for incorporating sustainability as an explicit goal into requirements engineering. Apart from the decomposition, we are also lacking a reference catalogue of sustainability-improving activities linked to indicators they affect. The latter would allow for a sustainability assessment in order to evaluate whether the activities affect the sustainability of a project or company.

Contribution We present a reference model for sustainability that decomposes sustainability into five dimensions: environmental, individual, social, economic, and technical sustainability. The model provides activities and relates them to values they support and assessable indicators. It is intended to serve as a reference model for a process engineer who instantiates the model for a software development company or for a requirements engineer who instantiates it for a specific system under development. The instantiation is illustrated by two examples taken out of current case studies. Our aim is to show how the aspect least supported by our current ways of developing software systems, environmental sustainability, can be aligned with the other sustainability dimensions.

2. Related Work

We identified related work within frameworks for sustainable software engineering, strategy models, i* modeling, requirements engineering techniques, and goal modeling.

Framework for sustainable software engineering Naumann et al. [8] provide a framework for sustainable software engineering.
They investigate how web pages can be developed with little environmental impact, i.e., energy-efficiently, and offer a respective guideline for web developers. In such a framework, the reference model proposed in the work at hand could be used as key element in order to better promote the method.

**Strategy models**  Gu et al. [3] propose a green strategy model that provides decision makers with the information needed to decide on whether to take “green” strategies and eventually how to align them with their business strategies. They consider such strategies as “green” which achieve lower energy consumption and perform a case study with Dutch data centers. In contrast, the paper at hand considers a broader definition of sustainability and thus gives a broader view on sustainable software engineering.

**i*-modeling**  Cabot et al. [4] report on a case study for sustainability as a goal for the organization of the ICSE’09 conference with i*-models to support decision making for future conference chairs. Stefan et al. [1] extend that work for managing environmental sustainability with quantitative goal modeling techniques. Both works provide model instances for specific case studies while the work at hand also provides a generic reference model.

**Requirements engineering techniques**  Mahaux et al. [6] present a case study on a business information system for an event management agency that advertises environment-friendly events. They assessed how well some current RE techniques support modeling of specific sustainability requirements in that case study. In contrast, our aim is to provide modeling means explicitly for integrating sustainability into the software development process as a major objective.

**Goal Modeling**  Lamsweerde [10] decomposes business goals into system requirements, but does not explicitly reflect on sustainability. His work relates positive and negative influences between goals, but our work provides activities for direct realisation.

3. Definition and Dimensions of Sustainability

We consider five dimensions of sustainability as important for the analysis of software systems. The basic dimensions for a general sustainability analysis (without referring to software systems) are individual, social, economic and environmental as defined by Goodland [2]. The three latter ones are the dimensions also known from the UN definition of sustainable development [9]. However, these four dimensions do not offer a possibility to claim and support long-term evolution of technical systems and adequacy for long-term use. Consequently, when looking at (software) systems, we need technical sustainability as an additional dimension.

**Individual sustainability**  Individual sustainability refers to the maintenance of the private good of individual human capital. The health, education, skills, knowledge, leadership and access to services constitute human capital [2]. For software engineering (SE), we have to ask: How can software be created and maintained in a way that enables developers to be satisfied with their job over a long period of time?

**Social sustainability**  Social sustainability means maintaining social capital and preserving the societal communities in their solidarity. Social capital are investments and services that create the basic framework for society [2]. For SE: Which effects do software systems have on the society (e.g. communication, interaction, government etc.)?

**Economic sustainability**  Economic sustainability aims at maintaining assets. Assets do not only include capital but also added value. This requires to define income as the amount one can consume during a period and still be as well off at the end of the period, as it devolves on consuming added value (interest), rather than capital [2]. For SE: How can software systems be created so that the stakeholders’ long term investments are as safe as possible from economic risks?

**Environmental sustainability**  Environmental sustainability seeks to improve human welfare by protecting natural resources. These are water, land, air, minerals and ecosystem services; hence much is converted to manufactured or economic capital. Environment includes the sources of raw materials used for human needs, and ensuring that sink capacities recycling human wastes are not exceeded [2]. For SE: How does software affect the environment during, inter alia, development and maintenance?

**Technical sustainability**  From a point of view of (software) systems engineering, there is another dimension that has to be considered. Technical sustainability has the central objective of long-time usage of systems and their adequate evolution with changing surrounding conditions and respective requirements. For SE: How can software be created so that it can easily adapt to future change?

The dimensions Social, Economic, Environmental and Technical can be analyzed on a micro level as well as on a macro level. The decision on which level they are investigated depends on the scope of the system under analysis. These five dimensions are not necessarily encompassing. One could argue, for example, that politics & law should be a separate dimension. On the other hand, the government can be considered as institutionalization of society and, consequently, as subdimension of the social dimension. However, for the analysis scope of software systems, we believe the five given dimensions to be an adequate representation as they serve only as structuring means for the to-be-derived model.

4. The Sustainability Model

Our proposed method comprises the generic sustainability reference model (M1 level), the respective meta model behind it (M2), and the instances (M0) derived for specific processes (companies) and systems (software products).

4.1 The Meta Model

![Diagram of the meta model of generic sustainability model (M2).](image)
The meta model is comprised by the types Dimension, Value, Indicator, Regulation, and Activity. A <Dimension> is a viewpoint, represented by a set of values that express the abstract objectives of the dimension. Each dimension is represented by a set of values. A <Value> is a rationale that is rooted in itself, and is approximated by indicators. An <Indicator> is a qualitative or quantitative metric and is related to a value. A <Regulation> is an optional element that affects a value. An <Activity> is a means to support and influence a value.

4.2 The Generic Sustainability Model

An excerpt of the generic sustainability model is provided in Fig. 2, see [7] for details. The model consists of three levels: the top level contains the dimensions; the middle level contains values, indicators, and regulations; and the lower level contains activities. Each element in the generic sustainability model is an instance of a type from the meta model explained before. In the following we will explain examples from the generic model, structured by their type.

Dimensions A dimension is an aspect of or viewpoint on sustainability, for example, environmental sustainability. As described in Sec. 3 there are five dimensions to be considered when analyzing sustainability for software systems engineering. A dimension is detailed in a set of values.

Values A value is a moral or natural good that is perceived as an expression of a specific dimension. Each of the five dimensions is represented by a set of values. Values do not necessarily belong exclusively to one dimension but can be considered for a number of dimensions, for example, healthy environment, which applies for both the environmental as well as the individual dimension.

Indicators An indicator is a qualitative or quantitative metric that expresses a specific degree or score with regard to a value, for example, satisfaction indices as qualitative metric and carbon emissions or return on investment as quantitative metrics. A set of indicators approximates a value.

Regulations A regulation is an optional element that affects (i.e. supports or enforces) a value, for example, emission regulations. Regulations commonly set limits for a specific indicator to be of legal use. Many values belonging to the different dimensions are heavily regulated, either supported or restricted in order to protect them. For example, freedom of the individual is supported by the human rights, and healthy air is supported by the European Union’s directive on carbon emissions.

Activities An activity is a measure taken to contribute to a specific value or a set of values, for example, use train for mid-distance traveling instead of aircraft. The impact of these activities on a value is measured by the indicators it influences. For the travel example, using a train instead of an aircraft improves the emissions account of the traveler. For each value, there is a number of activities that can be implemented to support a value. Thereby, the impact of an activity is measured by the indicators it influences.

The generic sustainability model is intended to serve as reference and as a basis for the instantiation of company- or system-specific instances.

5. Instantiation of the Sustainability Model

The generic sustainability model can be instantiated for development processes (companies) and for software systems (products). For the former, a sustainability goal model that represents the goals of the company is created. For the latter, we see the analysis of the sustainability model as part of the requirements engineering for a software system under development. The elicitation of indicators is related to the elicitation of key performance indicators. The resulting activities and indicators are part of the controlling during software development.

5.1 Methodical Description

Applying sustainability modeling within a company consists of two alternating phases: the analysis and the application & assessment phase. These phases and their respective steps are carried out by a process engineer (for a company-specific model) or a requirements engineer (for a product-specific model). Whereas the analysis phase mainly concretizes the definition of sustainability in the present context, the application & assessment phase starts the selected activities and supervises whether the activities are working successfully. Even though these are two separated phases we expect these phases to be alternating, thus refining and tuning the specific instance of the goal model iteratively over time. During the whole process the model guides the tailoring of values, indicators and activities to the context (company- or project-specific) by structuring the goals and giving examples and suggestions.

Analysis Phase In the analysis phase the generic goal model is tailored to the context (a specific company or software system). This formalizes what sustainability really means for a certain company or product. As depicted in the center of Fig. 3 the analysis phase includes three steps:

1. For each dimension the architect needs to instantiate the generic values that are provided by the generic goal model into context-specific goals. These goals need to be prioritized to help solving conflicts between potentially contradicting goals.

2. Afterwards the architect defines activities to implement the goals and defines indicators that make it possible to assess the state of this goal at the moment as well as in future situations. In this step he might find trade-offs between indicators that are more exact with respect to this value and indicators that are easier to apply. For example, the architect might have to decide whether to measure resource consumption by weight or by item number. The generic goal model serves as a reference for selecting activities and indicators.

3. When values and indicators for a certain dimension are selected, the architect can now relate the activities that have a positive impact on the indicators. For example, he could find reusable items in this project and set up policies that enforce this reuse.

Indicators and activities are usually developed iteratively. The architect can choose whether it is more adequate for them to start with defining activities or to list the most important indicators for a specific sustainability assessment. This way we instantiated the generic model with goals, indicators and activities specific to a company or a product.

Application & Assessment Phase After formalizing the company’s or product’s specific sustainability goal model, two parallel tasks are needed: First of all, each activity needs a person in charge (“Sustainability Activity Responsible”) for the implementation of the previously selected activities. Second, the Quality Engineer needs to continuously monitor the company’s state with respect to its sustainability model. They can do this by assessing the list of indicators that are concretely defined in the model. The results of this assessment are reported to the management and the sustainability architect, so that changes or adjustments can be made accordingly.

In summary, the main goal of the application & assessment phase is starting activities and monitoring how the company or product is performing with regards to its own definition of sustainability.
5.2 Company-Specific Instances
An excerpt of one company-specific instance of the generic sustainability model is depicted on the left in Fig. 3. According to the method described above we first instantiated the generic model in a company-specific goal model in the analysis phase. This could work like the following: For environmental sustainability we instantiate the value Reduce resource consumption into the goal Reduce resources consumption by 30% within 12 months. We find three indicators for resource consumption: physical waste, the company’s energy bill and the size of the bought items list. We can immediately see differences in how easy one can measure these indicators. From these indicators we can select activities that have an influence on the indicators. For example to reduce the size of the bought items list, we can place incentives for less resource consumption. All values, indicators and activities can be refined within the model. For example, the unspecific activity reduce waste is refined within the subactivity recycle packaging for own shipping.

In the application & assessment phase we would now need to establish the activities and measure our indicators. From this information we can deduce how well our activities are performing and whether or not we work towards the company-specific sustainability goals. The first industrial case study for a company-specific instance is currently under development in a small software development company (jambit GmbH) in a series of workshops performed within a Master’s thesis.

5.3 Product-Specific Instances
The first product-specific instances of the generic sustainability model were developed in an industrial case study of a car-sharing platform built on the recently launched mobility service DriveNow. An excerpt of the product-specific instance of the car sharing system DriveNow is depicted on the right in Fig. 3. As explained above, the analysis phase is similar to company-specific instantiations. For environmental sustainability the value reduce environmental impact is chosen. We can select similar indicators from the generic goal model as in the company-specific instantiation, such as the energy bill. Yet, we need to tailor our activities specifically to this product: For example, in this system the usage of green data centers is advised. Furthermore, the business process can be varied by offering older cars for a lower rental rate (instead of depositing of them) and thereby reducing waste, and by introducing cars with hydro or electric power. Again, in the application & assessment phase, we need to make sure that these activities are followed during the development of a product and measure the impact of activities.

6. Discussion
The previous example already shows both benefits as well as problems and open work of modeling with the generic sustainability model.

Benefits Many elements of the model (such as the physical waste indicator) reappear in various instances. Hence, much of the modeling can be reused and only needs to be selected. Second, sustainability, a rather abstract concept, is turned into a concrete and measurable property of a company or a product and corresponding actions that directly relate to improvement. By making these implicit definitions explicit via the model, it is possible to discuss and evaluate a company’s or product’s impact on sustainability.

Issues While creating the instances we were faced with the question of how to measure the impact of the proposed improvement activities for a product that does not yet exist. For example, when taking a product design decision such as using green data centers, there is no “before” that could be compared to an “after” situation as the system does not exist yet. In that case, the model can still serve well in a constructive way for decomposing sustainability goals and selecting activities to realise them, but the analytic phase after implementation reveals less concrete results.

Threats to Validity One threat to validity arises from our constructive research approach for the model. The risk is limited by our experience in developing such models, by reviews from and discussion with other researchers, and by feedback from industrial partners.
7. Conclusion

We propose a generic sustainability model with process- and product-specific instances that can help requirements engineers to analyse their projects according to the different dimensions of sustainability and choose actions for improvement. Special emphasis was put on environmental sustainability in and via software systems, as this dimension of sustainability is the one that is least supported by our traditional ways of developing software systems.

Future Work One important issue for future work is how to treat conflicting goals. Conflicts can be identified by contradicting influences on same indicators, but solving them is more complex. For example, the ever-present conflict between a tight budget and quality improvement measures is also found in the sustainability goals and activities (i.e. in indicators for economical vs. indicators for technical sustainability). However, more challenging is the question whether there are conflicts that are not as obvious and how to track them down if they do not relate to same indicators in the first place. The other major issue is to understand the reuse potential of instances (how systems-specific are they really?) and the gradual extension of the reference model by knowledge gained from instantiating the model in various companies. We expect continuous improvement by evolving the model over time in the ongoing cooperation with our industrial partners.

Acknowledgement We would like to thank Susanne Klein, Oliver Feldmann, and Jonas Eckhardt for helpful comments on drafts of this paper as well as Debra Richardson and Bill Tomlinson for feedback on early ideas.

References