Towards a Definition of Sustainability 
in and for Software Engineering

Birgit Penzenstadler
Institut für Informatik
Technische Universität München
Germany
penzenst@in.tum.de

ABSTRACT

Sustainability is not supported by traditional software engineering methods. This lack of support leads to inefficient efforts to address sustainability or complete omission of this important concept. Defining and developing adequate support requires a commonly accepted definition of what sustainability means in and for software engineering. We contribute a description of the aspects of sustainability in software engineering.

Categories and Subject Descriptors
D.2.1 [Software Engineering]: [sustainability, environment, requirements engineering, quality assurance, guidance]

1. MOTIVE: COMMON UNDERSTANDING

Although many people in software engineering (SE) are by now aware of the general definition of sustainability as “the capacity to endure” and sustainable development as “meeting the needs of the present without compromising the ability of future generations to meet their own needs” [10], there still is no concrete guidance for the different aspects of sustainability that are observable from the point of view of software engineering. This is also due to the fact that the concept of sustainability does not become adequately tangible from that definition.

The consequence is that, as the topic of sustainability is now receiving attention in software engineering, our discipline tries to contribute. However, without a thorough and commonly accepted definition of what sustainability means applied to software engineering, our individual contributions remain somewhat insular and isolated. In order to realize green computing, we need such a definition.

Problem: Traditional software engineering has not fully supported sustainability as a relevant, first-class concern. We software engineers approach specific topics that have to do with sustainability in our discipline, for example, green IT, efficient algorithms, smart grids, agile practices and knowledge management, but we lack a common understanding of the concept of sustainability and if and how it can be encompassingly applied to software engineering.

Contribution: In this paper, we describe the different aspects of sustainability from a point of view of software engineering and exemplarily illustrate their realisation in activities. This provides software engineers with a preliminary guideline on how sustainability can be incorporated into their daily practice, for example during requirements engineering and quality assurance.

Outline: After providing background on frameworks for sustainability, the relation between ICT and sustainability, and earlier approaches to definitions in the context of SE in Sec. 2 we contrast the aspects of our definition of sustainability in SE and relate it to the referenced frameworks in Sec. 3. We conclude with future work in Sec. 4.

2. BACKGROUND & FOUNDATIONS

The background for the presented definition approach is structured into frameworks for defining sustainability, the relation between ICT and sustainability, and earlier approaches to defining sustainability in the context of software engineering.

2.1 Frameworks for Defining Sustainability

The following frameworks are designed to serve for defining sustainability in general, without a specific reference to software systems. Dobson [3] suggests a framework for the comparison of sustainability notions that poses conceptual questions any conception of sustainability must answer: What to sustain? Why to sustain? Who/what concerned? Substitutability allowed?

Burger and Christen [1] propose a capability approach of sustainability. Their methodological approach is to formulate adequacy conditions for concepts of sustainability, illustrate a categorial framework with the required general concepts, and propose a conception of sustainability based on the capability approach. They consider the idea of sustainable development as a problem-solving strategy (for the “developmental dilemma” [10] and their adequacy conditions are future-orientation, normative power, justice, universality, limited natural conditions, and high-level strategic actions [1, p. 788]. The developed approach is supposed to serve as basis for empirical research in development studies.

Robert et al. [8] present the Framework for Strategic Sustainable Development that is now mainly promoted by the Swedish NGO ‘The Natural Step’. Their objective is to show how the increasing number of tools and approaches to develop sustainability (e.g., the ISO14001, Life Cycle Assess-
ment, and Ecological Footprinting) relate to each other and build on each other when used for planning for sustainability. Following a strictly analytic approach in a theoretic contribution to the discourse in sustainable development, Christen and Schmidt propose a Formal Framework for Conceptions of Sustainability that intends to solve the problem of arbitrariness. The framework consists of modules that serve to help to elaborate the elements answering the ‘what’ and ‘how’ questions.

### 2.2 Relation between ICT & Sustainability

Hilty, Lohmann, and Huang provide an overview of the fields of ICT in the service of sustainability: Environmental Informatics, Green IT, and Sustainable Human-Computer Interaction. As technological efficiency alone will not produce sustainability (cf. Jevon’s paradoxon), sustainable development requires a combination of efficiency and sufficiency strategies, inter alia by decoupling economic growth from environmental impacts and from the use of natural resources.

Furthermore, Hilty, Arnfalk, Erdmann et al. analyse the relevance of information and communication technologies for environmental sustainability. They present the impacts of ICT on environmental sustainability on different levels: first order effects like increasing electronic waste streams, second order effects such as improved energy-efficiency of production, and third order effects like a product-to-service shift or rebound effects in transport.

### 2.3 Sustainability in the Context of SE

Penzenstadler et al. provide a systematic literature review that points to two definitions of sustainability in the context of software engineering, i.e. by Mahaux et al. and by Naumann et al. Mahaux et al. simply use the Brundtland definition plus the statement that IT changes behavior and therefore has considerable effect on society and environment, which is supported by greenIT concepts and analysis of the usage processes of a software system.

Naumann et al. define “green and sustainable software” such that “direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software (...) ha(ve) a positive effect on sustainable development”, and “Green and Sustainable Software Engineering” such that the “negative and positive impacts on sustainable development (...) are continuously assessed, documented, and used for a further optimization of the software product” [p. 296].

These two works show that the term sustainability is strongly dependent on the taken perspective, and even the second definition cannot be operationalised. However, this is necessary if SE researchers want to contribute. Consequently, scope and context have to be clearly defined to be able to make any statements.

### 3. Aspects of Sustainability in Software Engineering

Applying the principles of system thinking, sustainability can be defined as preserving the function of a system over a defined time span. Consequently, there are three variables, namely system, function, and time, that need to be defined for setting the scope for a discussion on sustainability. An absolute definition has fixed variables, while a relative definition choses the variables according to the context. For sustainability aspects in software engineering, the author’s aim is to relate a relative definition of sustainability in the context of software engineering to an absolute one. This relation allows to discuss the definition in comparison to the definitions by other disciplines.

#### 3.1 Absolute and relative definition

An absolute definition of sustainability has fixed variables: the time span is various generations, the function is a satisfaction of needs, the system is humanity in its ecosystem. For a relative definition, the variables of system, function and scope still have to be chosen. This choice for the variables can be made independent of a “higher value”, and just signify plain capacity to endure. For example, I can have a manufacturing site for cars running in a way that all orders are satisfied (function “develop cars in time for orders”) but still be economically, socially, and ecologically unsustainable in various ways. If I want to avoid the latter, the function has to refer to an absolute definition, e.g., “develop cars in time for orders respecting the sustainability guidelines defined by our company”.

Applied to software engineering, the differentiation can be made by distinguishing software (engineering) for sustainability, which is related to the absolute definition, and sustainable software (or sustainability in software engineering), which is related to the relative definition.

#### 3.2 Defining aspects in software engineering

Apart from this differentiation of sustainability in software and software for sustainability, we distinguish four aspects.

The first two focus rather on the developing company and its processes, while the latter two have the system under development in scope. These aspects are derived from iterated discussions with various experts on sustainability and they shall serve as structuring means for discussing and supporting sustainability activities rather than as an apodictic differentiation.

**Development Process Aspect.**

Sustainability during the initial software development process means development with responsible use of ecological, human, and financial resources.

**System:** a software system development company (including its staff as well as its operational environment with equipment)

**Function:** to perform software development with minimized environmental impact, a sufficient economic balance, and good working conditions

**Time:** the planning period of the long-term business plan of a software development company; depends on the company’s size and the general duration of development projects, may vary from months to years

**Maintenance Process Aspect.**

Sustainability of the software system during its maintenance period until replacement by a new system includes continuous monitoring of quality and knowledge management.

**System:** the maintenance department of the software system development company
Function: to maintain and evolve a software system with minimized environmental impact, a sufficient economic balance, and well-managed knowledge

Time: planning period of the lifetime of the system; depends on the contracted maintenance period for projects, may vary from a couple of years to decades

**System Production Aspect.**
Sustainability of the software system as product with respect to its use of resources for production is achieved, for example, by using green IT principles, sustainably produced hardware components, and optimising the required logistics for assembly, etc.

System: a conjoint of developing and supplier companies that collaborates for assembling the specific product

Function: to produce a system with minimized environmental impact and a sufficient economic balance

Time: according to the project plan for production; depends on the system’s size and complexity; may vary from a couple of months to years

**System Usage Aspect.**
Sustainability in the usage processes within the application domain triggered by the software system as product takes into account responsibility for the environmental impact and designing green business processes.

System: the system under analysis with operational environment & business context in its application domain

Function: to maintain and evolve a software system with minimized environmental impact, a sufficient economic balance, and social responsibility

Time: estimated lifetime of the system; depends on system’s size and complexity; varies from years to decades

The lifecycle of a system of course continues through to replacement and disposal, but for software the activities of reengineering and refactoring are already considered within the Maintenance Process Aspect.

### 3.3 Relation to frameworks

In order to give a justification for our definition, we compare it to the definitions of the others from background section and set them in context within the frameworks.

The definition by Mahaux et al. [6] is basically included in our’s as they rely on Brundtland [10] which we consider as one possible version of an absolute definition of sustainability. The definition by Naumann et al. [9] is expressed within the four aspects, only expressed differently. By this differentiation, our definition makes it easier to assign actions to the supporting the various notions of sustainability.

Considering Burger and Christen’s [1] adequacy conditions (future-orientation, normative power, justice, universality, limited natural conditions, and high-level strategic actions), the definition needs to be extended to satisfy all of the required information. However, this can only be done within a specific context, for example, a specific company of for a specific system type, most likely only within a specific country. Consequently, also the categories named in [1] can only be fulfilled by the referenced absolute definition. Christen and Schmidt’s [2] framework modules (problem, justice, integration, criteria, and transformation) are equally given by the absolute definition. The definition may be integrated into the framework by Robért et al. [8] and, finally, Dobson’s [3] questions (what? why? who? in what respect?) are all satisfied by the aspect definitions above due to the description of the variables of system, function, and time.

Certainly, this short justification is not a sufficient discussion of the adequacy of our definition, but it is an indicator that it complies to the requirements for definitions in (general) sustainability research. Thus, future SE sustainability research may build upon this definition.

### 4. CONCLUSION AND FUTURE WORK

This paper presents a definition of the aspects of sustainability in and for software engineering. Thereby, for software engineering is how to make SE itself more sustainable and in software engineering is how we improve the sustainability of the systems we develop. The aspects have, in a next step, to be extended with a catalogue of actions to realize and/or improve them. We are currently elaborating a case study with a software development company where we align the process and a system under development with a set of sustainability principles in a pilot project.

As future work, we envision an encompassing approach for software engineering that supports building sustainability into software products from early requirements engineering on. This will also include a corresponding assessment model.

**Acknowledgement** The author would like to thank Lorenz Hilty for inspiring discussions as well as Bill Tomlinson and Debra Richardson for feedback on early ideas.

### 5. REFERENCES