Classifying Green Software Engineering - The GREENSOFT Model

Stefan Naumann, Eva Kern
Institute for Software Systems
Environmental Campus Birkenfeld
P.O. Box 1380, D-55761 Birkenfeld
(s.naumann|e.kern)@umwelt-campus.de

Markus Dick
Sustainable Software Blog
http://sustainablesoftware.blogspot.de
sustainablesoftwareblog@gmail.com

Introduction. Up to now several relationships between Information and Communication Technology (ICT) and Sustainable Development (SD) are published. However, especially in the field of energy aware or green software there is a lack of detailed descriptions. Since this field is rising, it is useful to formulate some definitions and take a look at the life cycle of software. These classifications can also help to develop a research agenda for energy aware software and its development.

Green Software and Green Software Engineering. In many cases, the reason for establishing energy efficient software and systems is to achieve a longer battery life or generally to reduce costs. On top of that, moving to the ecological part of sustainability, there is the potential to decrease energy and resource consumption by ICT to support a SD. A first impression on how software influences the life cycle of the hardware by requiring more and more resources is given by Hilty. The so called "Software Bloat" denotes the effect that the availability of more powerful hardware in the near future, relaxes software developers efforts to produce highly efficient code [Hilty 2008]. A methodology to measure and incrementally improve the sustainability of software projects is presented by [Albertao et al. 2010]. They say it is advisable to implement sustainable aspects continuously, divided into the following phases: Assessment Phase, Reflection Phase, and Goal Improvement Phase. In order to make the different sustainability issues manageable, they pointed out properties of a quality model [Albertao et al. 2010]. Based on the life cycle of software, Taina proposed metrics [Taina 2011] and a method to calculate the carbon footprint of software [Taina 2010]. To do so, he analyzed the impacts of each software development phase for a generic project. The resulting carbon footprint is mainly influenced by the development phase, but also by the way how the software is delivered and how it will be used by the customers. The main problem regarding the calculation is that detailed data is required, which is often not available. Summarizing, we have to look after the (software) product and also after the process which produces this product. In any case, the impacts on environment and sustainable development have to be considered. That leads to the following definition: "Green and Sustainable Software is software, whose direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software are minimal and/or which has a positive effect on sustainable development" [Dick et al. 2010]. Consequently, Green and Sustainable Software Engineering should produce Green and Sustainable Software in a sustainable way.

The Software Life Cycle. Fig. 1 depicts an overview for the life cycle of software and its relationship to different levels of effects. It is inspired by the effect differentiation of [Berkhout et al. 2011] who distinguish between first order effects (effects resulting directly from the product, e.g. energy consumption), second order effects (usage results, e.g. effects of dematerialization by software), and third order or rebound effects (e.g. when an energy-efficient product leads to more energy consumption in total). In Fig. 1 we distinguish between the life cycle phases development, usage, and end of life.

The GREENSOFT Model. To summarize these different aspects of Green and Sustainable software we developed the GREENSOFT Model [Naumann et al. 2011]. This is a conceptual reference model for "Green and Sustainable Software", which has the objective to support software developers, administrators, and software users in creating, maintaining, and using software in a more sustainable way. The model comprises the shown holistic life cycle model for software products, sustainability criteria and metrics for software products, procedure models for different stakeholders, and recommendations for action, as well as tools that support stakeholders in developing, purchasing, supplying, and using software in a green and sustainable manner.

Summary and Conclusions. In our paper we described a life cycle inspired view for Green Software and its engineering. At first, we have to distinguish between the process and the product itself. Regarding the product and following the model, it is necessary to specify metrics and measurements and clar-

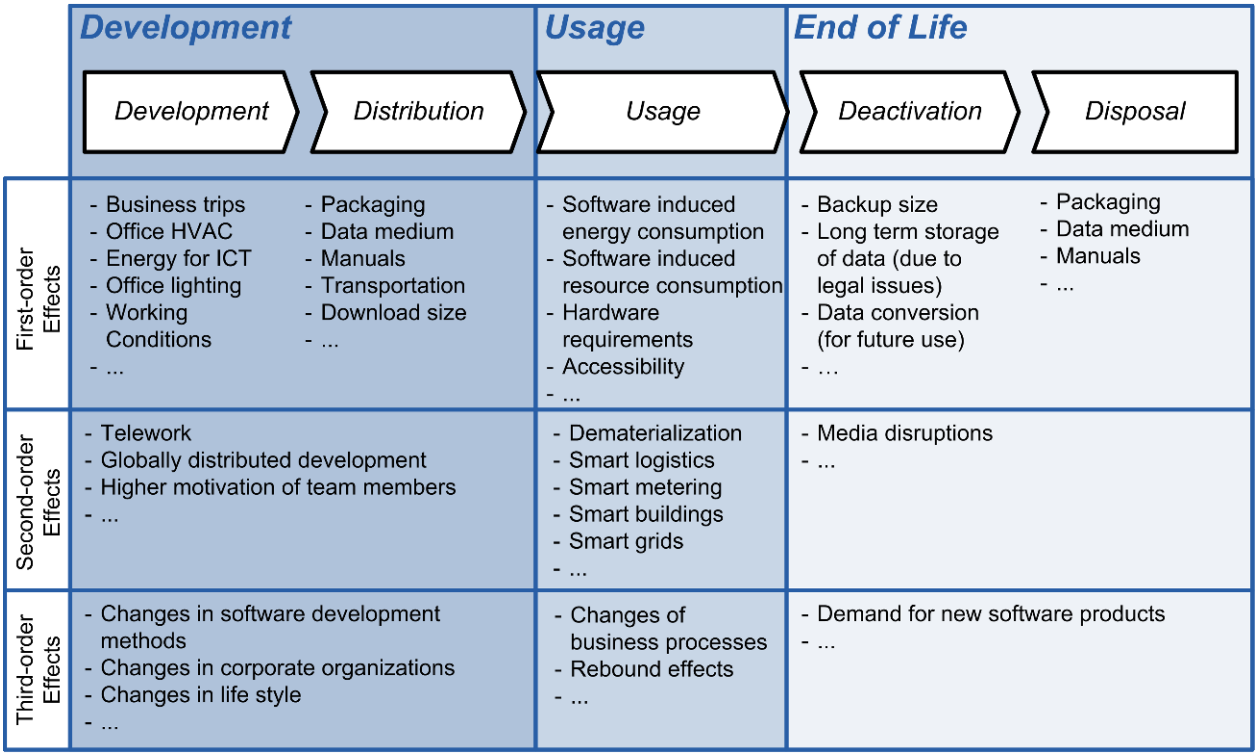

Figure 1: Software Life Cycle and Effects of different Phases (www.green-software-engineering.de/images/downloads/green_and_sustainable_software_product_life_cycle_96dpi_web.png), Accessed 30 March 2013

ify, how software products can be compared regarding their energy consumption. Here, it is necessary to define usage scenarios especially for standard software in order to compare different products. Another possibilty is to compare the energy consumption of different versions or releases. Here, an integration of measuring energy consumption into the continous integration process might be useful. Regarding the process, additional aspects of the sustainability of software production should be taken into account. Here, aspects like heating, greenhouse gas footprint or energy consumption have to be considered.

References

1. Albertao, F., Xiao, J., Tian, C., Lu, Y., Zhang, K. Q., and Liu, C. 2010. Measuring the Sustainability Performance of Software Projects. In 2010 IEEE 7th International Conference on e-Business Engineering (ICEBE 2010), Shanghai, China, 369-373.

2. Berkhout, F. and Hertin, J. 2001. Impacts of Information and Communication Technologies on Environmental Sustainability: speculations and evidence. Report to the OECD. http://www.oecd.org/dataoecd/4/6/1897156.pdf. Accessed 30 March 2013.

3. Dick, M., Naumann, S., and Kuhn, N. 2010. A Model and Selected Instances of Green and Sustainable Software. In What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience. IFIP Advances in Information and Communication Technology 328. Springer, Berlin, Heidelberg, 248-259.

4. Hilty, L. M. 2008. Information technology and sustainability. Essays on the relationship between ICT and sustainable development. Books on Demand, Norderstedt.

5. Naumann, S., Dick, M., Kern, E., and Johann, T. 2011. The GREENSOFT Model: A Reference Model for Green and Sustainable Software and its Engineering. SUSCOM 1, 4, 294-304. doi:10.1016/j.suscom.2011.06.004

6. Taina, J. 2010. How Green Is Your Software? Software Business. First International Conference, ICSOB 2010, Jyväskylä, Finland, June 21-23, 2010. Proceedings. Lecture Notes in Business Information Processing 51, 151-162.

7. Taina, J. 2011. Good, Bad, and Beautiful Software - In Search of Green Software Quality Factors. CEPIS UPGRADE XII, 4, 22-27.