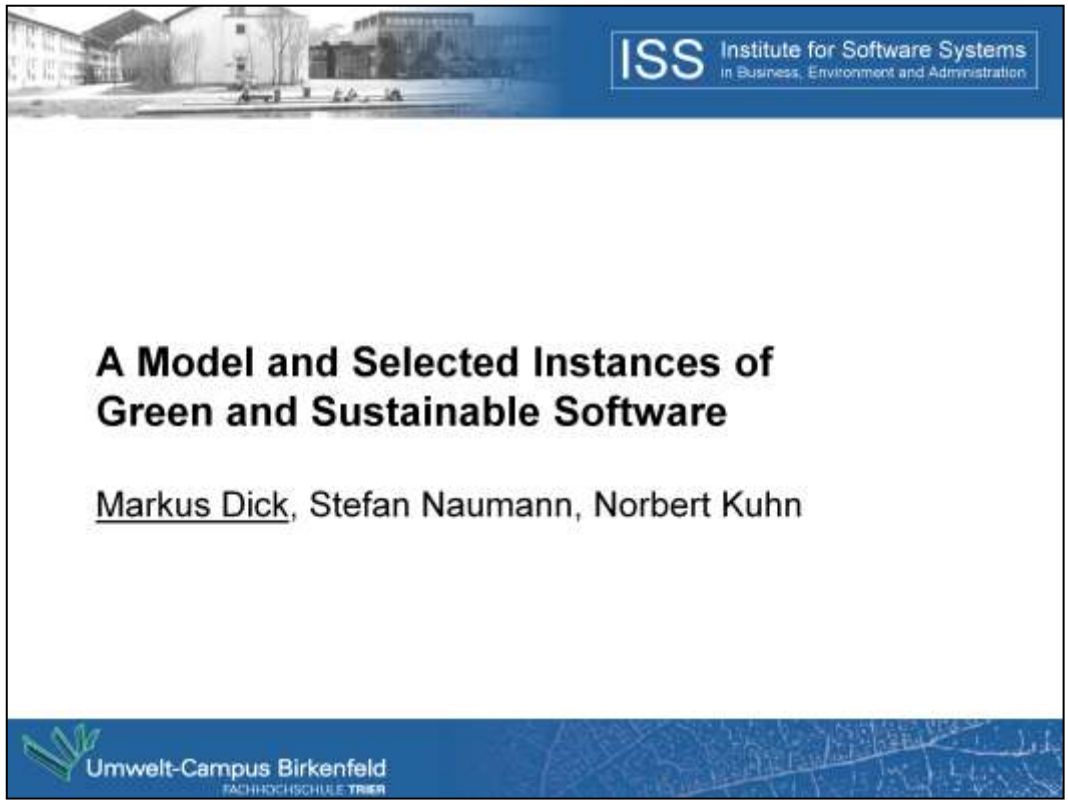


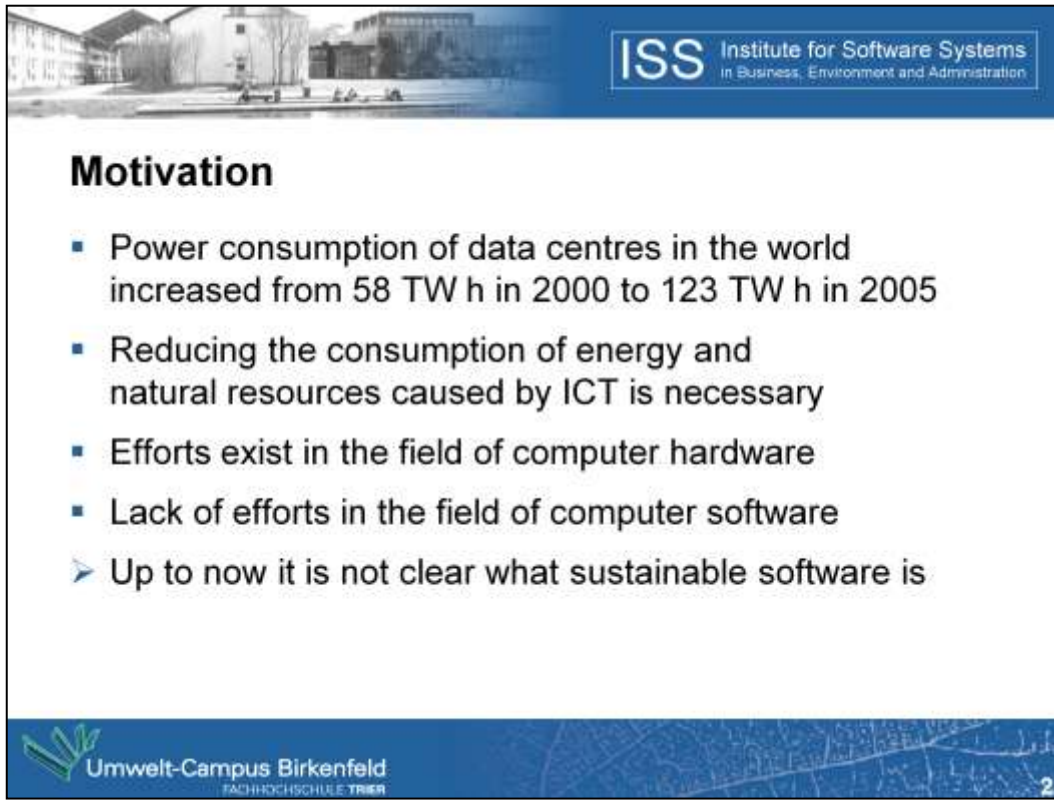
Markus Dick, Stefan Naumann, Norbert Kuhn
{m.dick, s.naumann, n.kuhn}(at)umwelt-campus.de

Trier University of Applied Sciences, Umwelt-Campus Birkenfeld
Campusallee, D-55768 Hoppstädten-Weiersbach, Germany

http://www.green-software-engineering.de/

This presentation corresponds to the following paper:
Dick, Markus; Naumann, Stefan; Kuhn, Norbert: ***A Model and Selected Instances of Green and Sustainable Software***. In: Berleur, Jacques; Hercheui, Magda; Hilty, Lorenz (eds.): What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience. IFIP Advances in Information and Communication Technology. 9th IFIP TC9 International Conference, HCC 9 2010 and 1st IFIP TC 11 International Conference, CIP 2010. Held as Part of WCC 2010. Brisbane, Australia, September 2010 Proceedings. Boston: Springer, 2010, pp. 248 - 259.

The project “Green Software Engineering” (GREENSOFT) is sponsored by the German Federal Ministry of Education and Research under reference 17N1209. The contents of this document are the sole responsibility of the authors and can under no circumstances be regarded as reflecting the position of the German Federal Ministry of Education and Research.



The power consumption of data centres in the world increased from 58 TW h in 2000 up to 123 TW h in 2005 and is still increasing
Hence, reducing the consumption of energy and natural resources caused by ICT is necessary
Where manifold efforts exist in the field of computer hardware (that is: Green-IT), there is a lack of models, descriptions, or realizations in the field of computer software.
Especially, there are hardly any systematic methods available that try to integrate sustainability aspects in software product design and development.
Up to now it is not clear what the term sustainable software means or what sustainable software is.

Outline

I. Definition of “Sustainable Software”

II. A Model for Sustainable Software Engineering

III. Selected Instances of the Model

IV. Summary & Outlook

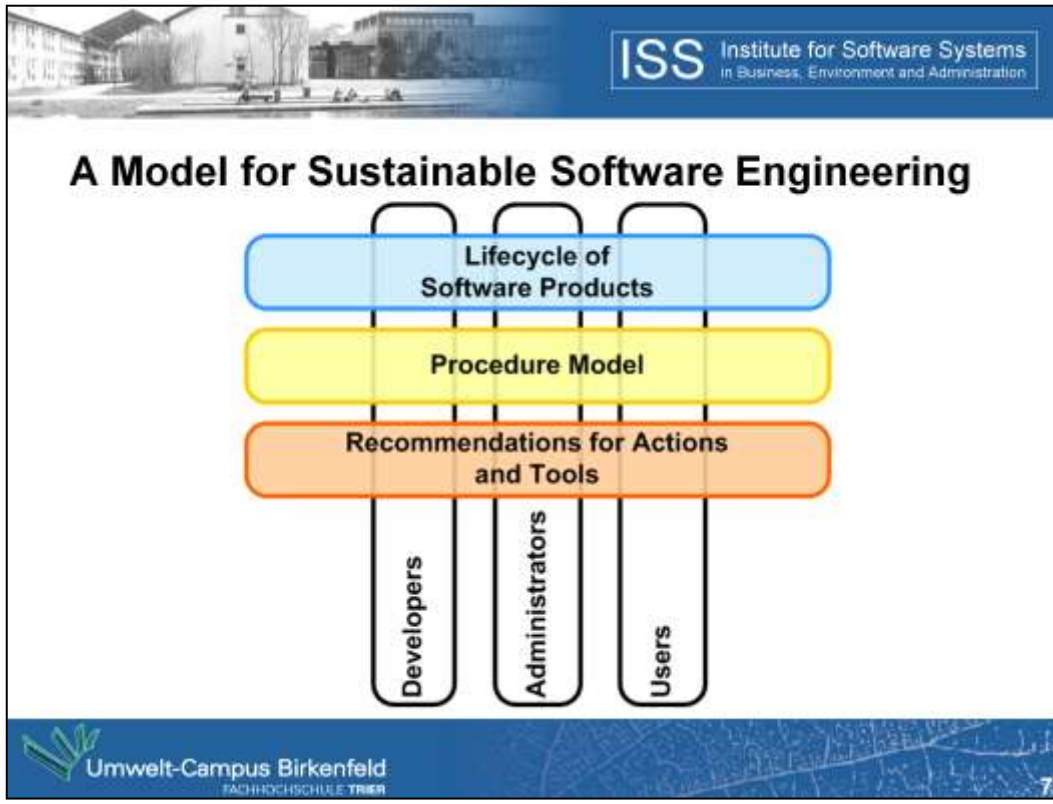
I. Definition of "Sustainable Software"

In this definition, direct impacts comprise energy and resource demand that is necessary to produce, use and dispose of the software product. Indirect impacts are effects that result from using the software product on other processes and long term systemic effects.

ISS Institute for Software Systems
in Business, Environment and Administration

II. A Model for Sustainable Software Engineering

Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

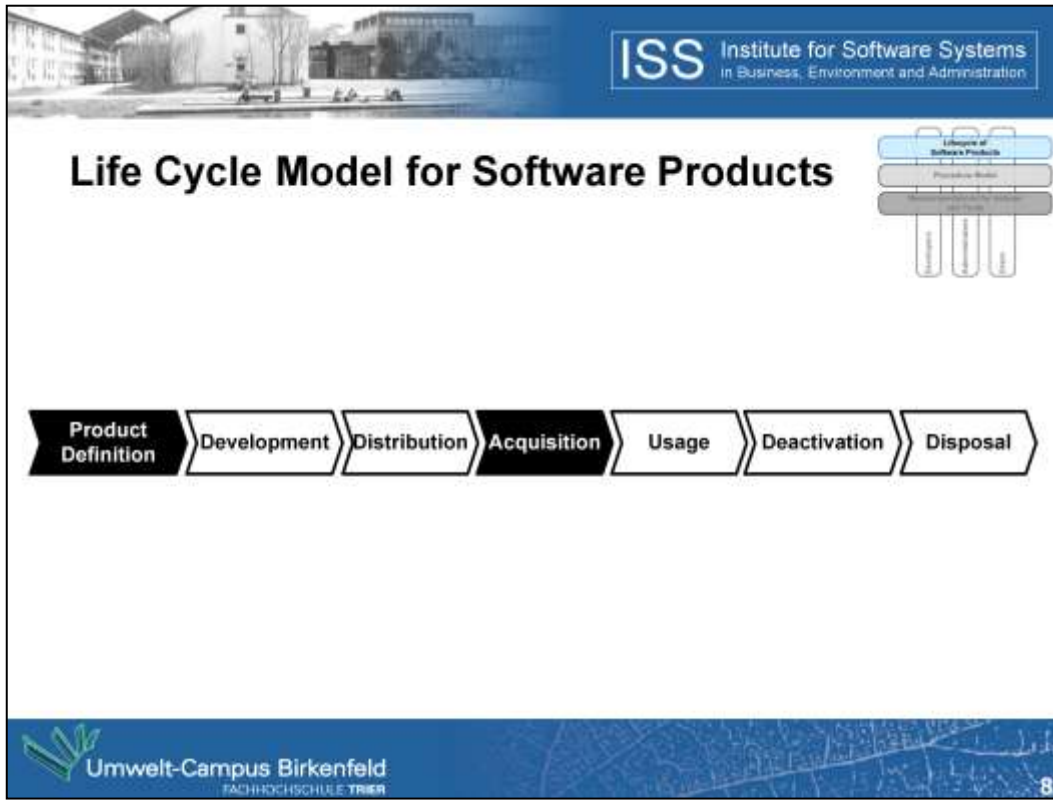


Our model comprises
–a lifecycle model for software products
–a procedure model
–and recommendations for actions and tools

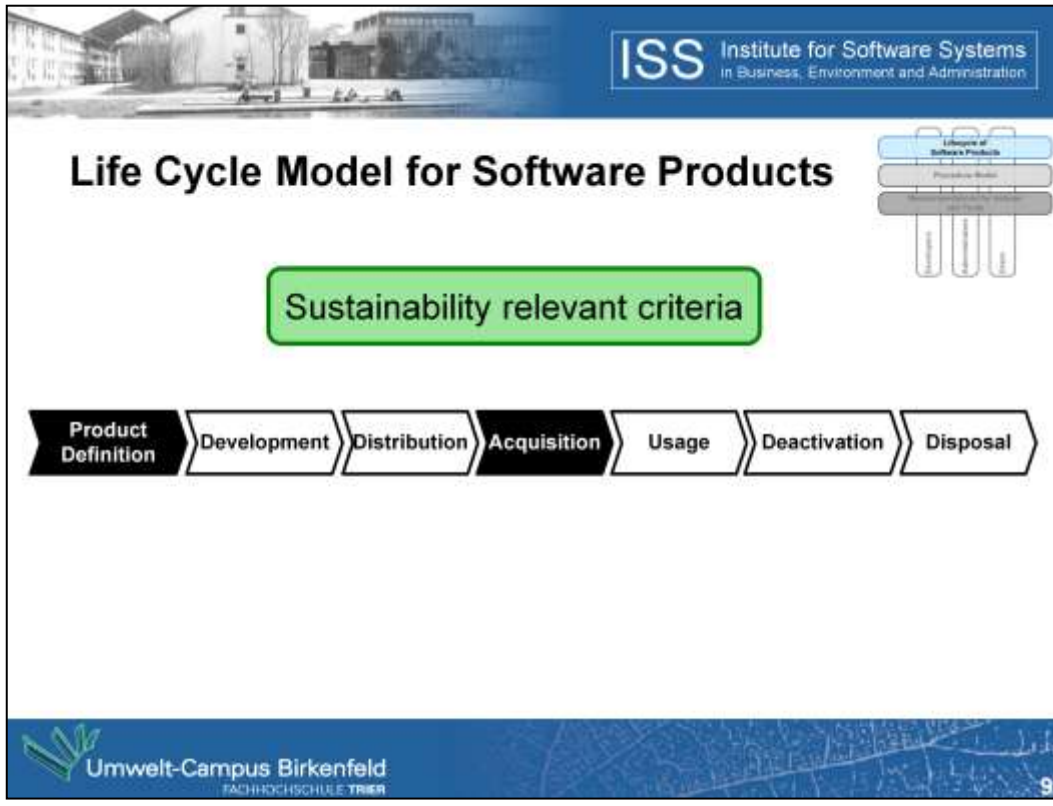
These are all provided for several roles, like Developers, Administrators, and Users.
Of course, there may be further and more specialized roles.

In principle we have two views in this model:
–the product view, which is given by the lifecycle model
–the organizational view, which is given by the procedure model

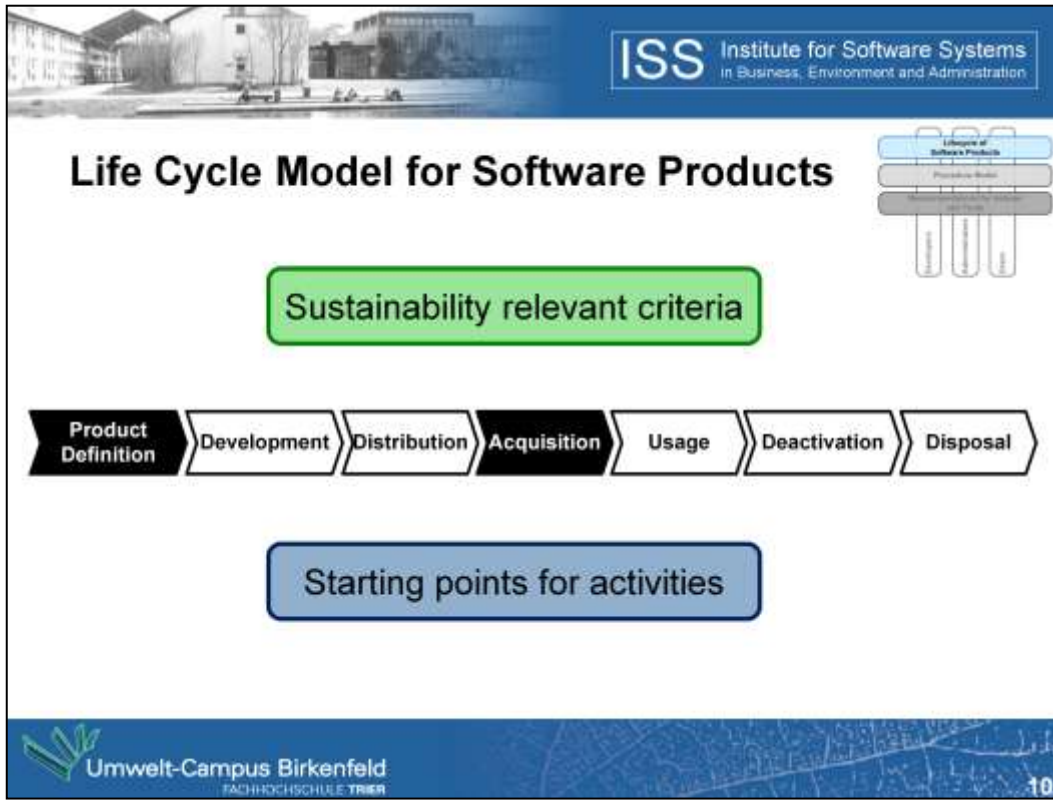
Both are accompanied by recommendations and tools, which support involved people in applying the procedure model and considering the whole lifecycle of a software product.



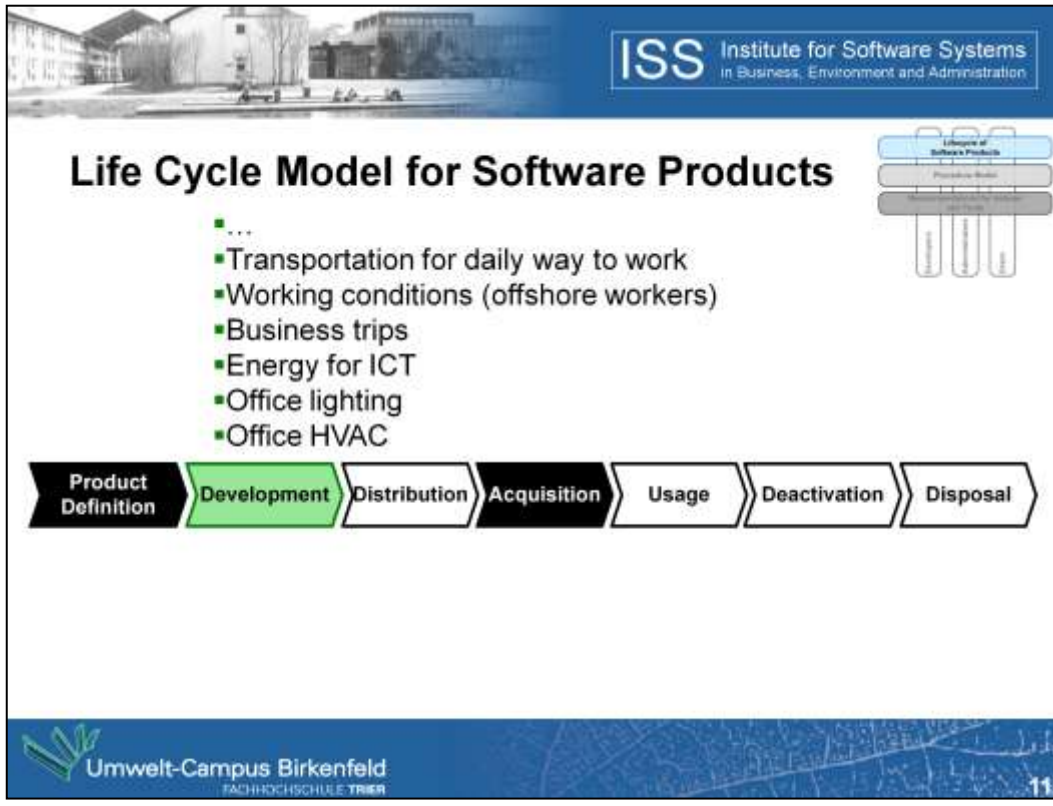
If you look at the phases of our lifecycle model, you will recognize, that it is more a product lifecycle in the sense of Life Cycle Thinking than an ordinary software lifecycle, because these are usually focusing on development phases and activities.
This model fits mainly to standard software products. For custom software products the phase "Acquisition" may follow directly after the phase "Product Definition".



The model has two objectives:
Its first objective is to assign criteria to the life cycle phases that lead to/result in sustainability relevant effects.

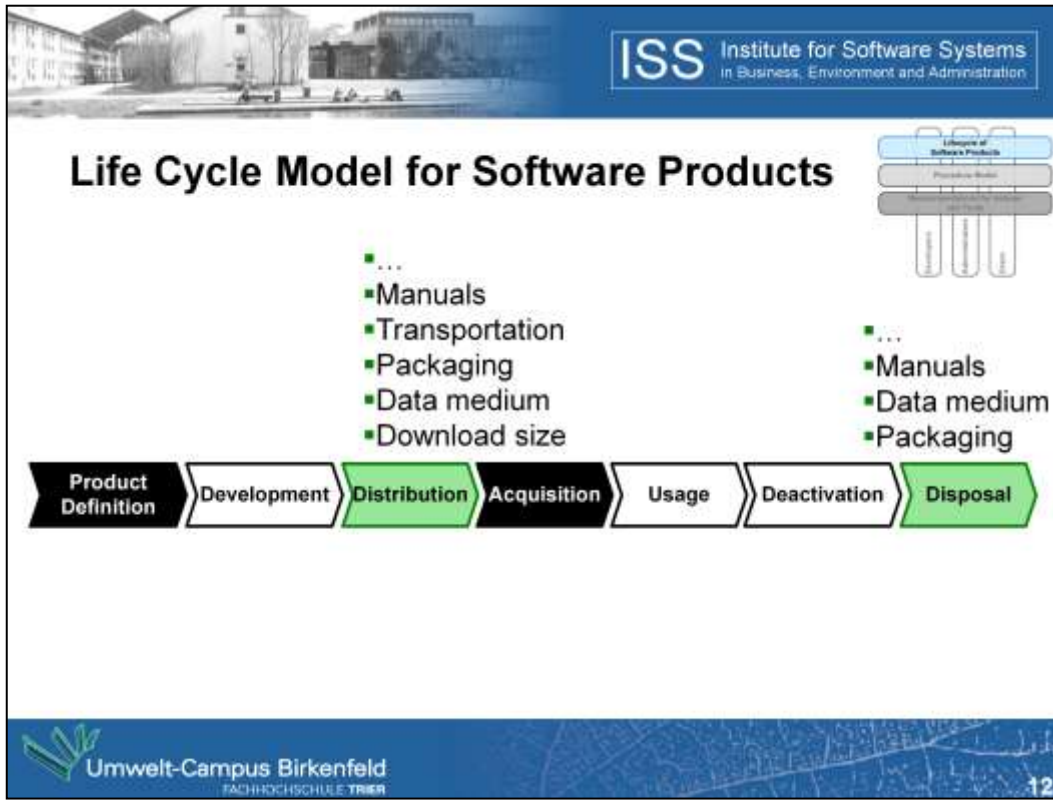


Its second objective is to provide starting points for activities that allow an assessment of the sustainability relevant effects that result from the software product over its whole life cycle.
Hopefully, these activities lead to more sustainable software products.



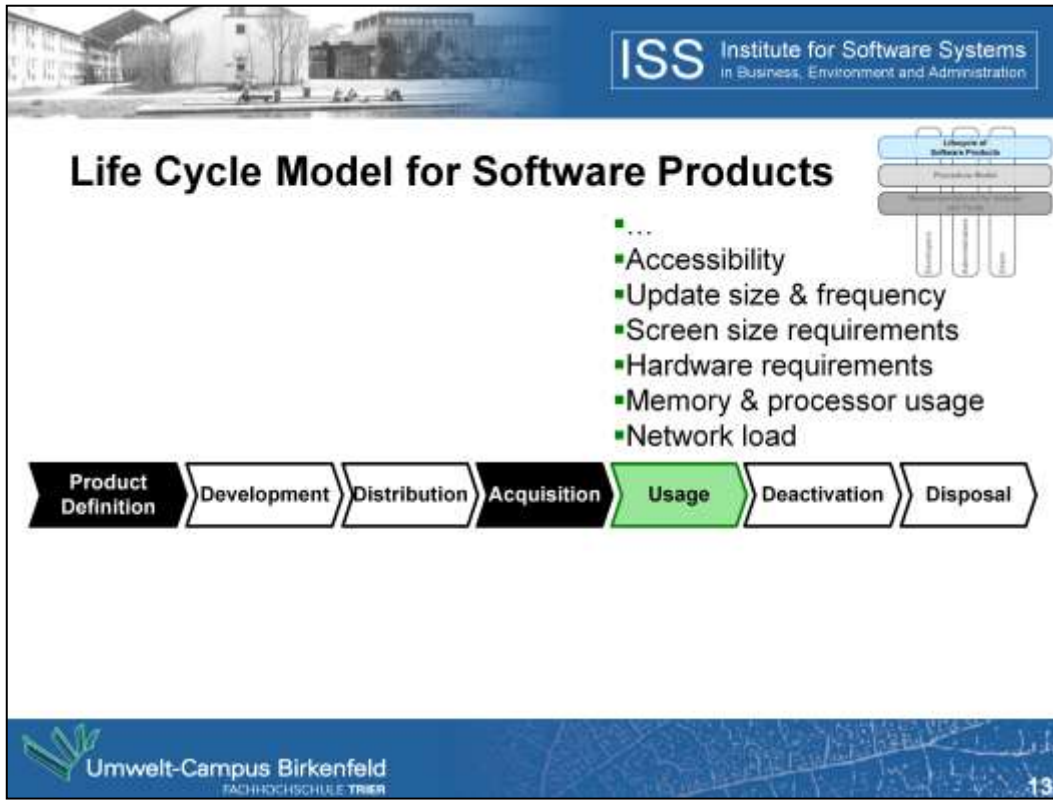
Now, let's have a look at some example criteria. Please, note that these examples are far from complete.

Examples of criteria for the development phase are:
–Working conditions
–Business trips
–Energy for IT-Infrastructure
–Office heating and air conditioning

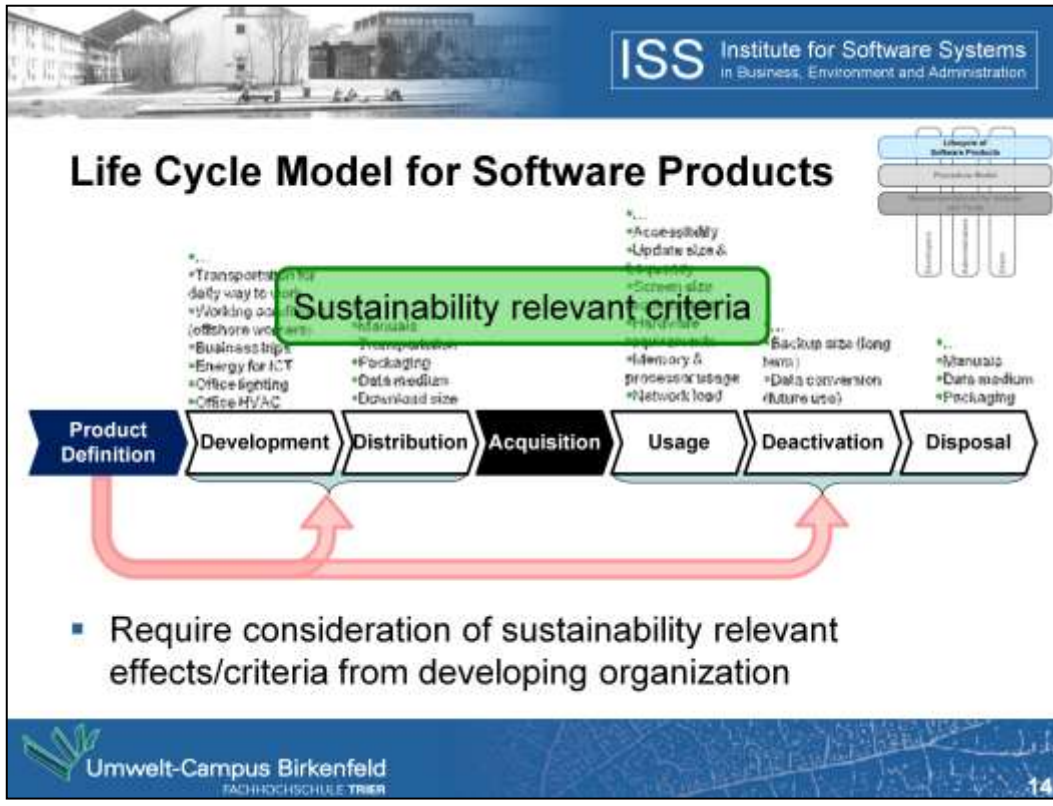


Examples of criteria for the distribution phase are
–Printed manuals
–Packaging
–Data medium
–Download size

These relate directly to criteria for the disposal phase, like
- the disposal of printed manuals
- the data medium, and
- the packaging

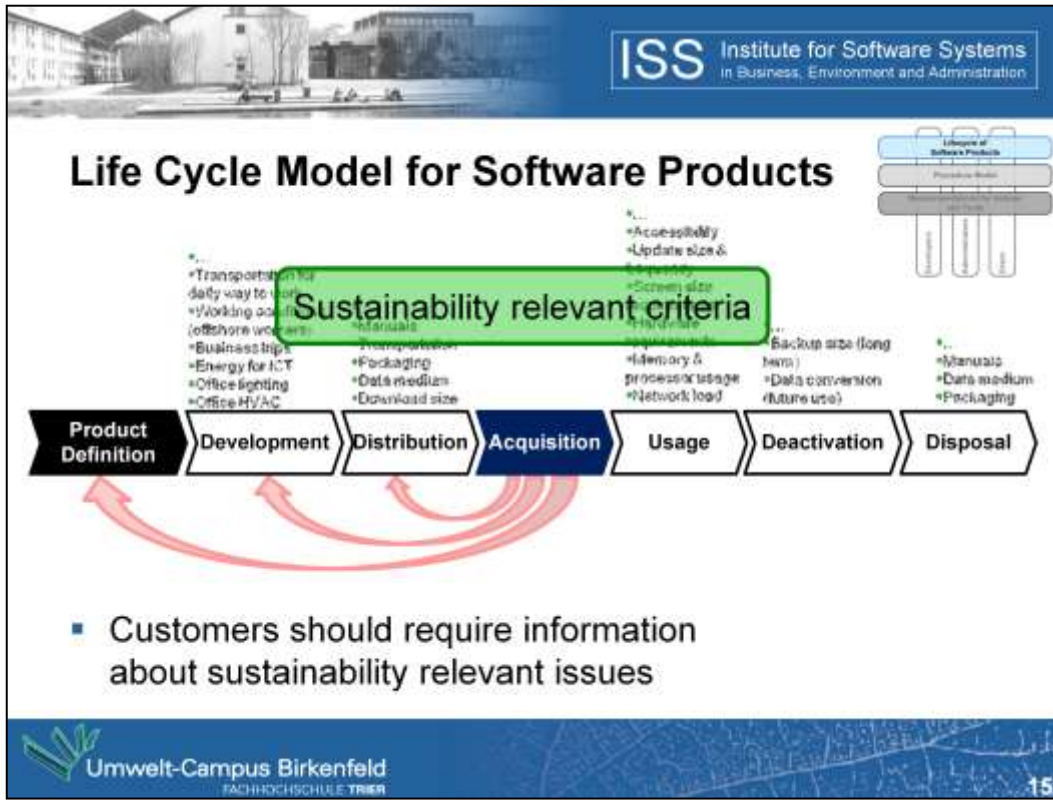


Examples of criteria for the usage phase are:
–Accessibility issues
–screen size requirements
–other hardware requirements
–memory and processor use during program execution



There are two preconditions:

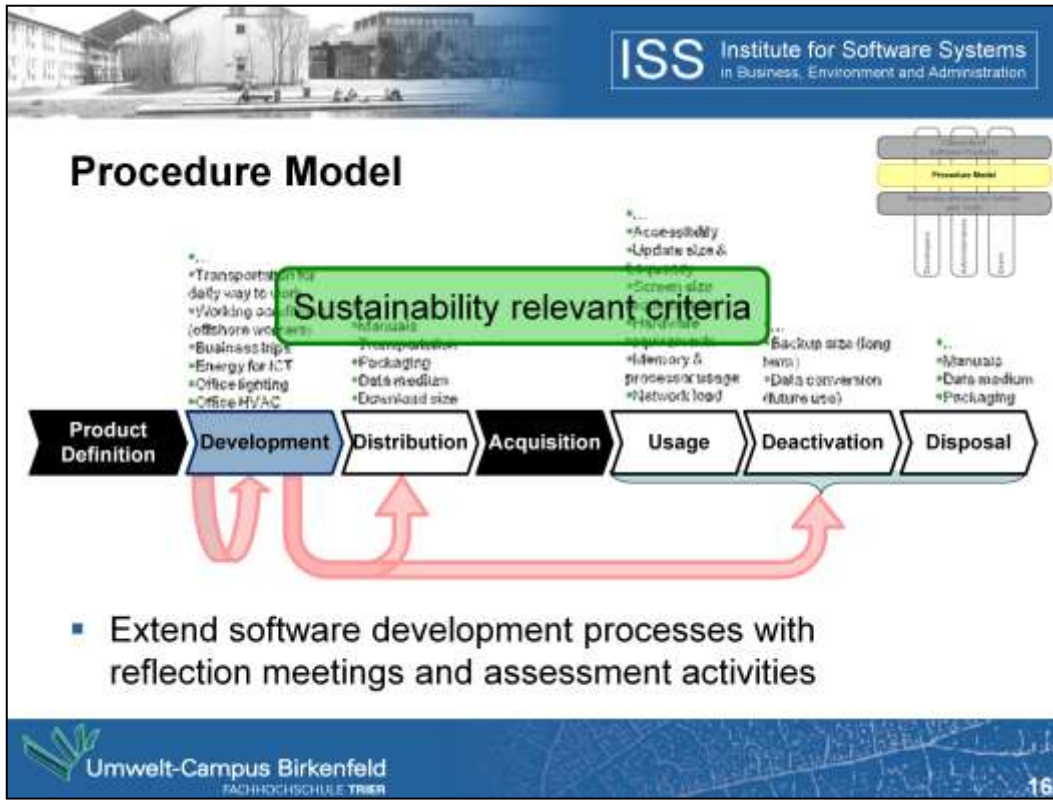
1. During the Product Definition phase, consideration of sustainability relevant effects/criteria must be required from the developing organization (e.g. as non-functional software requirements). This forces the developing organization to consider these criteria during development and to apply further activities in order to assess impacts that result from the software product over its whole life cycle.



2. Customers should require information about sustainability relevant issues.

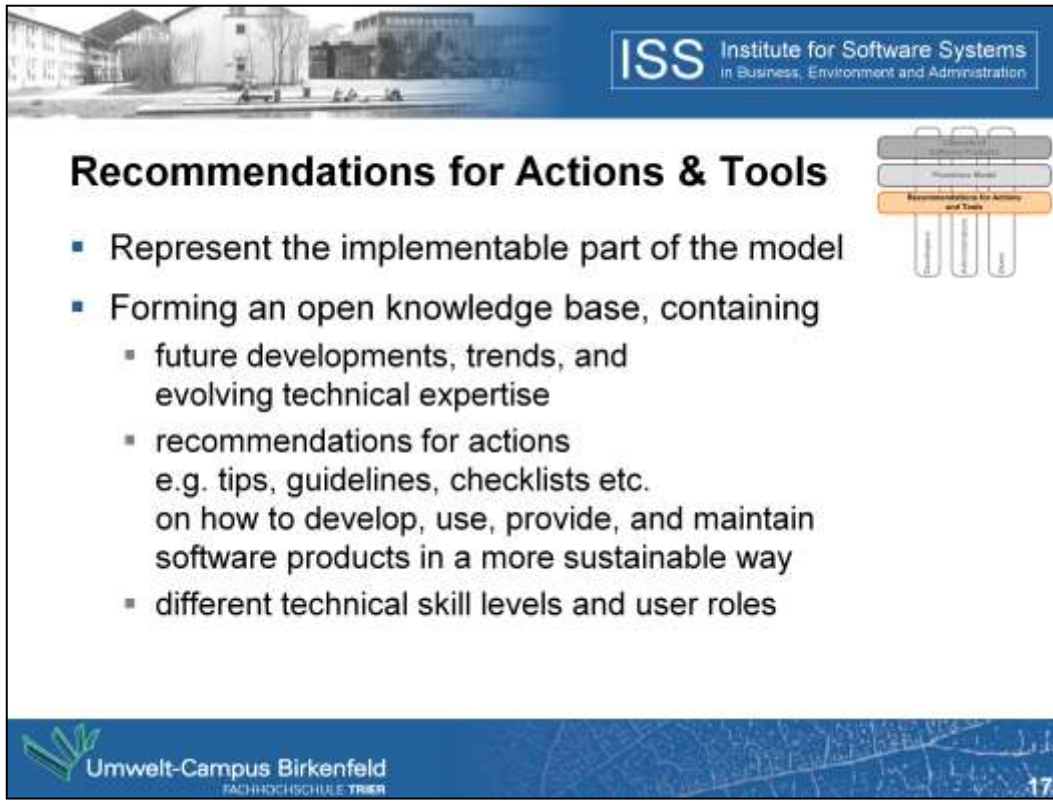
For custom software, this condition is met with the non-functional requirements
and with e.g. reports about assessments and applied activities.

For standard software, this may lead to some kind of a seal of quality or eco-label for software products in the future.



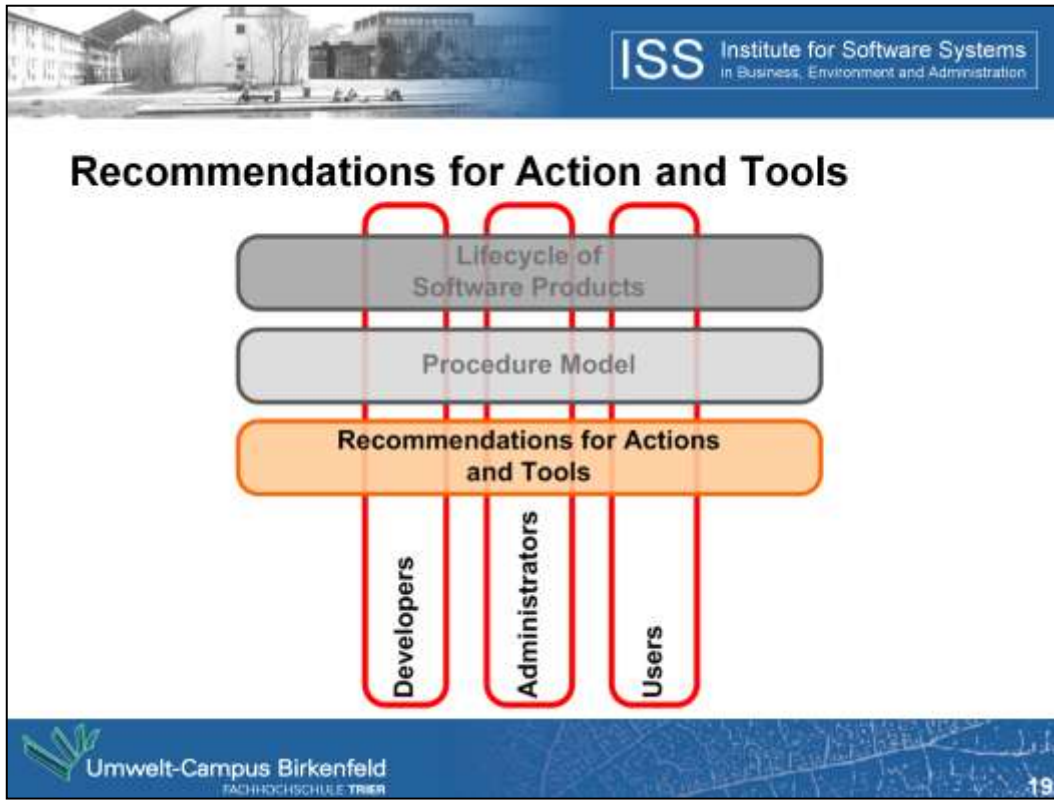
We propose a procedure model that extends arbitrary software development processes with continuous reflection meetings and assessment activities. These are targeting at assessing direct and indirect impacts,
–which result from the software development process itself and
–which are expected to arise from distribution and future use of the software product

The outcomes of the assessments and reflection meetings should be used to take action towards more sustainable software products.



Recommendations for Actions & Tools represent the implementable parts of our model.
In principle they form an open knowledge base, which enables the consideration of future developments, trends, and evolving technical expertise. It contains recommendations for actions (e.g. tips, guidelines, checklists) that provide information on how to develop, use, provide and maintain software products in a more sustainable way. These recommendations can be written for people with different technical skill levels and different user roles within the life cycle of a software product.

III. Selected Instances of the Model

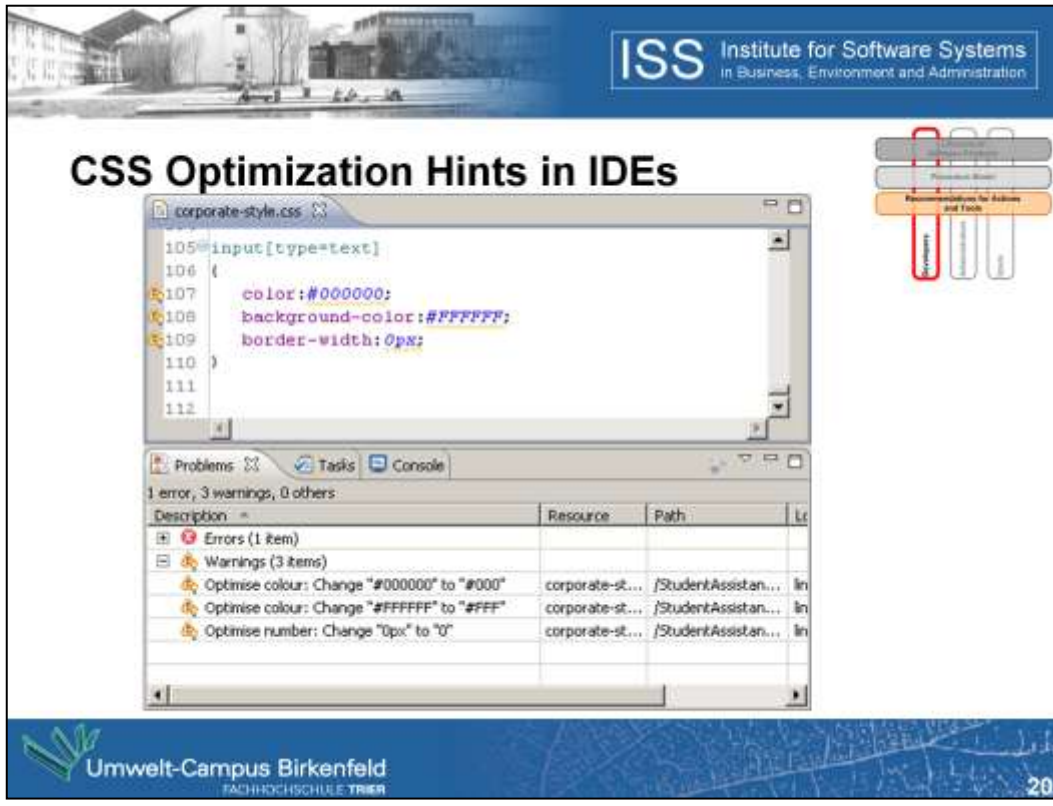


Now, to some selected instances of our model, which are realized as recommendations for actions and software tools for

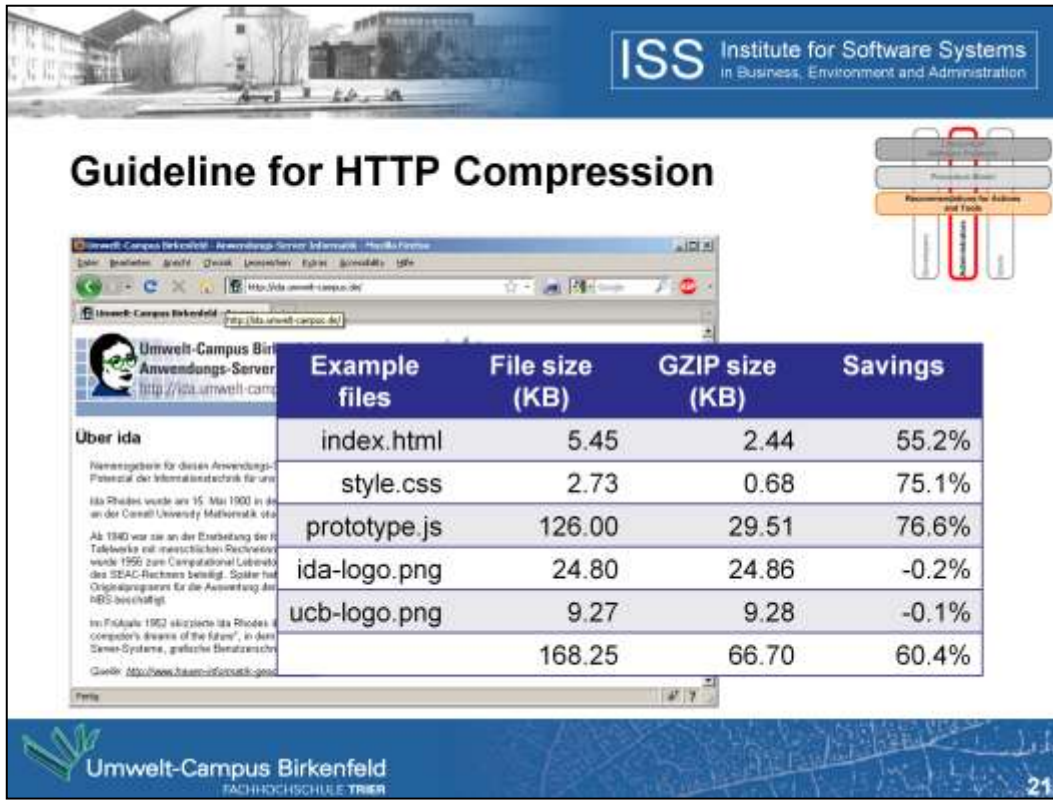
–Developers

–Administrators

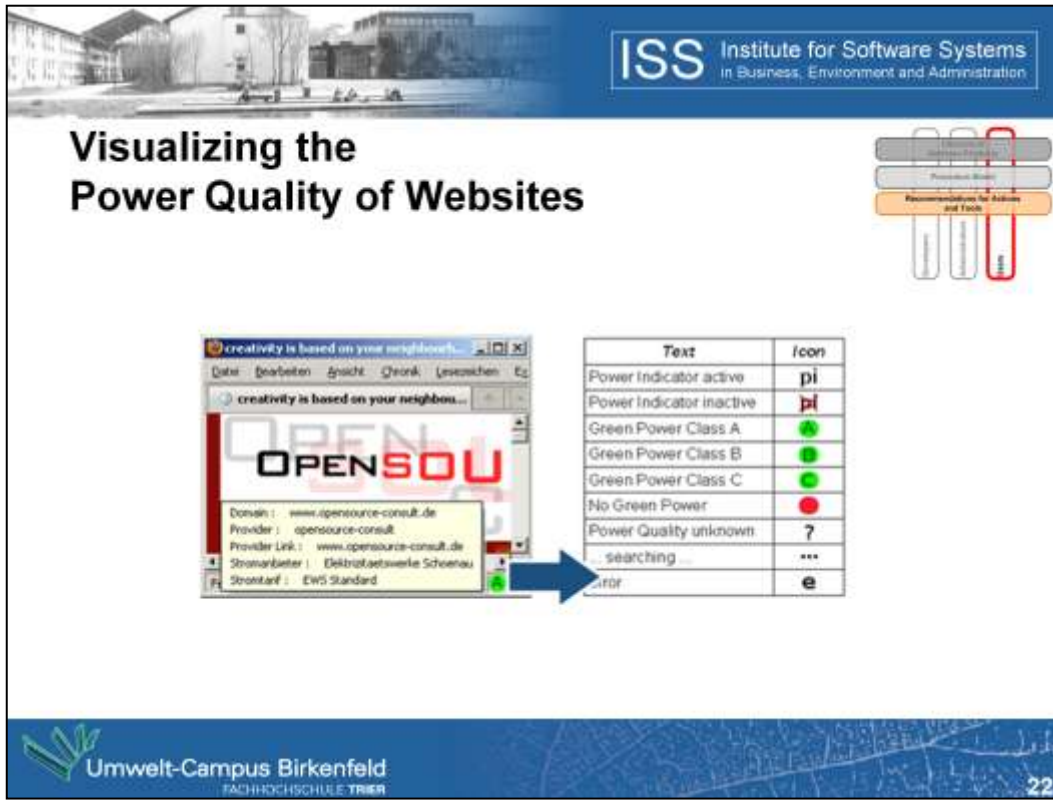
–Users



This screen-shot shows an example of a tool, which is integrated in the eclipse IDE. It analyzes CSS files and gives hints on how these files may be optimized in order to reduce their filesize. The tool can be applied by web developers during the development phase, which leads to a lower download size of the website during the usage phase.

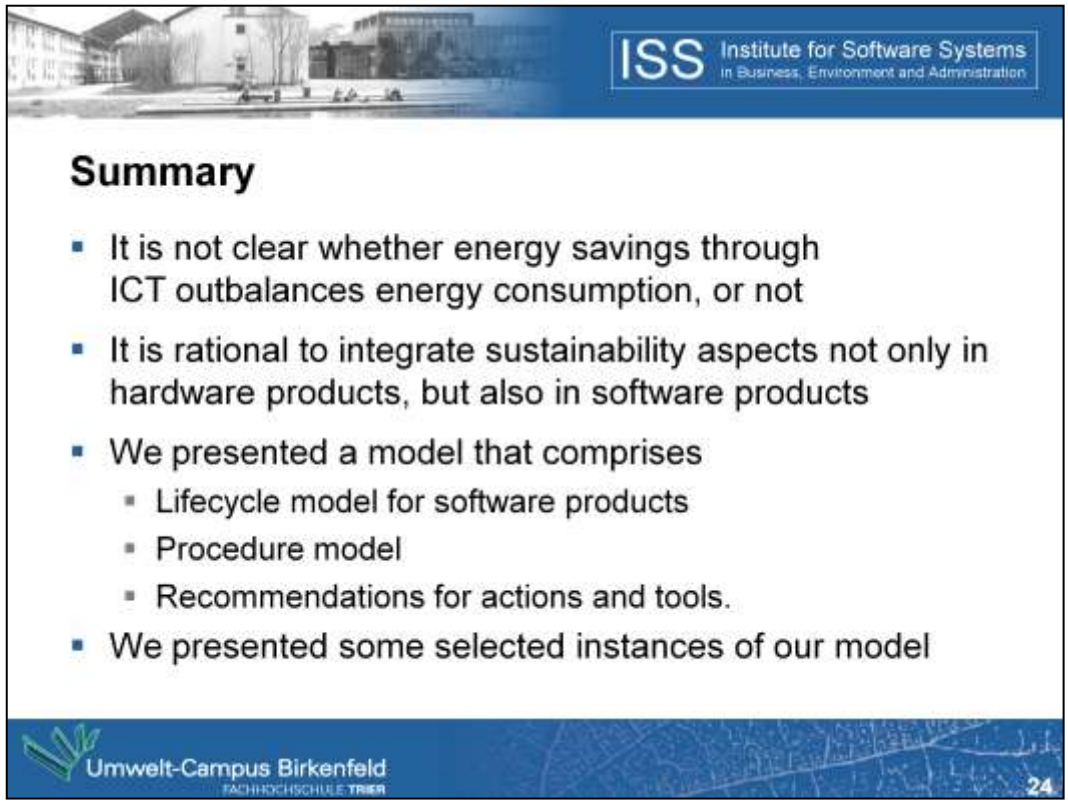


This slide motivates a guideline for HTTP compression. By enabling HTTP compression in web servers, it is possible to reduce network traffic. For this example web site, we achieved overall savings from 50 to 76 per cent for uncompressed file types. This guideline can be applied by administrators during the usage phase when they set up the web server for the website. It affects the download size for all users who view the website during the usage phase.



The Power Indicator is an add-on for the Firefox browser. It visualizes, whether a website is hosted with renewable energy or not. It belongs to the usage phase and it is intended to support web users in making informed decisions on which websites they want to or do not want to surf on.

IV. Summary & Outlook

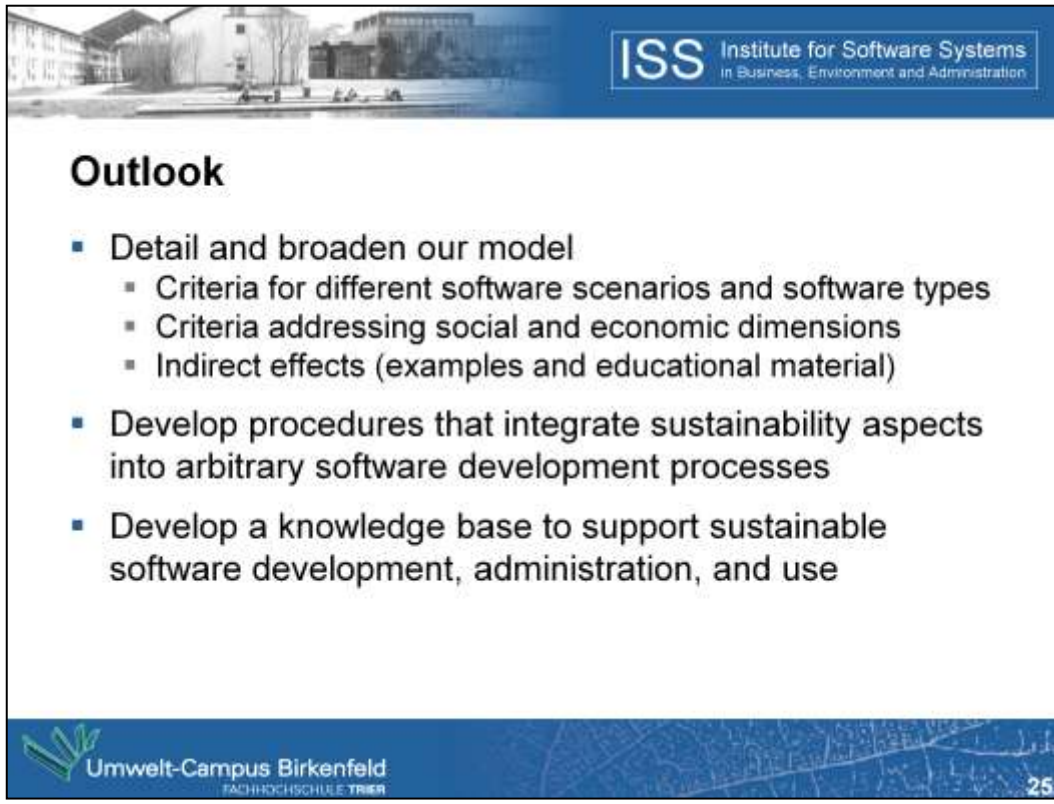


Summarizing, it is currently not clear whether energy savings through Information and Communication Technology outbalances its energy consumption or not. In either case it is rational to integrate sustainability aspects in software product design and development as it is already common for material products today.

Hence, we presented a model that comprises

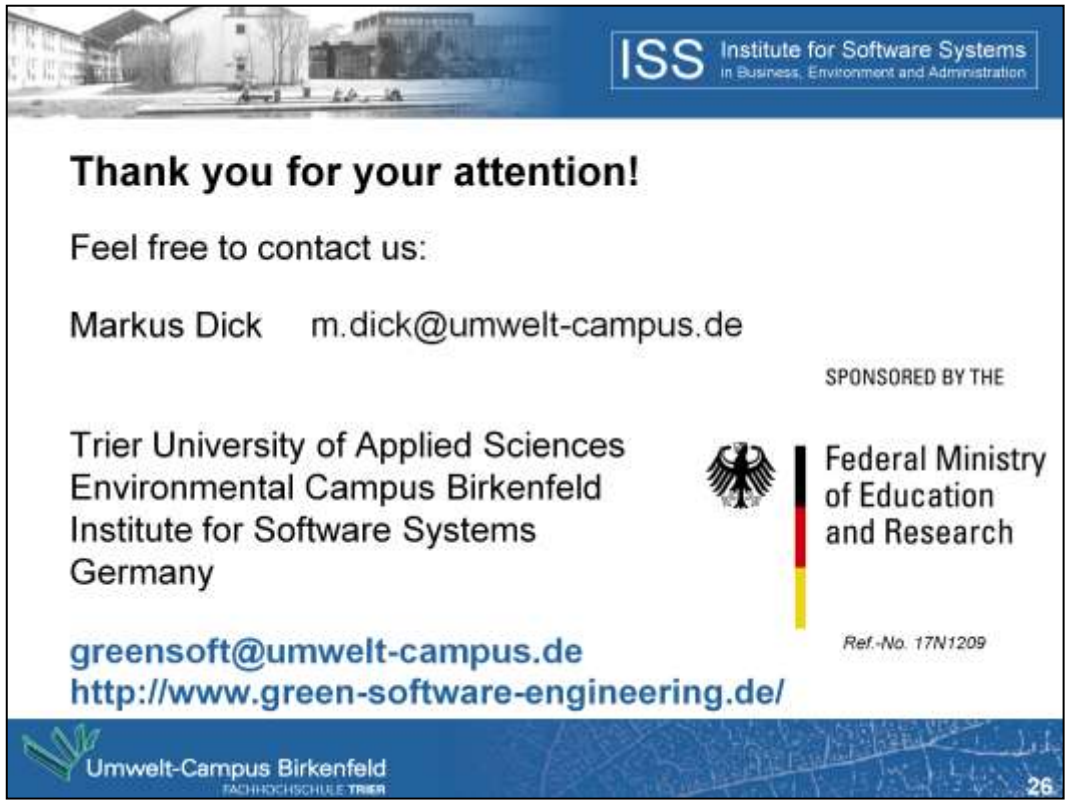
- A Life Cycle Thinking inspired Life Cycle of Software Products,
- A Procedure Model, and
- Recommendations for Actions and Tools.

As implementable instances of the model, we presented some examples of these recommendations and tools.



Our next steps are to detail and broaden our model with e.g.
–More criteria for different software scenarios and types of software
–Criteria that addresses the social and economic dimensions of sustainability
–Examples and Educational Material that address indirect effects of software use, because we do not expect software developers to recognize these intuitively.

Two main parts are currently missing: the procedure model and plenty of recommendations and tools. Hence, we plan to develop and evaluate procedures that integrate or help to integrate sustainability aspects into arbitrary software development processes and we plan to develop and operate the already mentioned knowledge base that provides recommendations for actions, which support development, administration, and use of software in a more sustainable way.



The project “Green Software Engineering” (GREENSOFT) is sponsored by the German Federal Ministry of Education and Research under reference 17N1209.