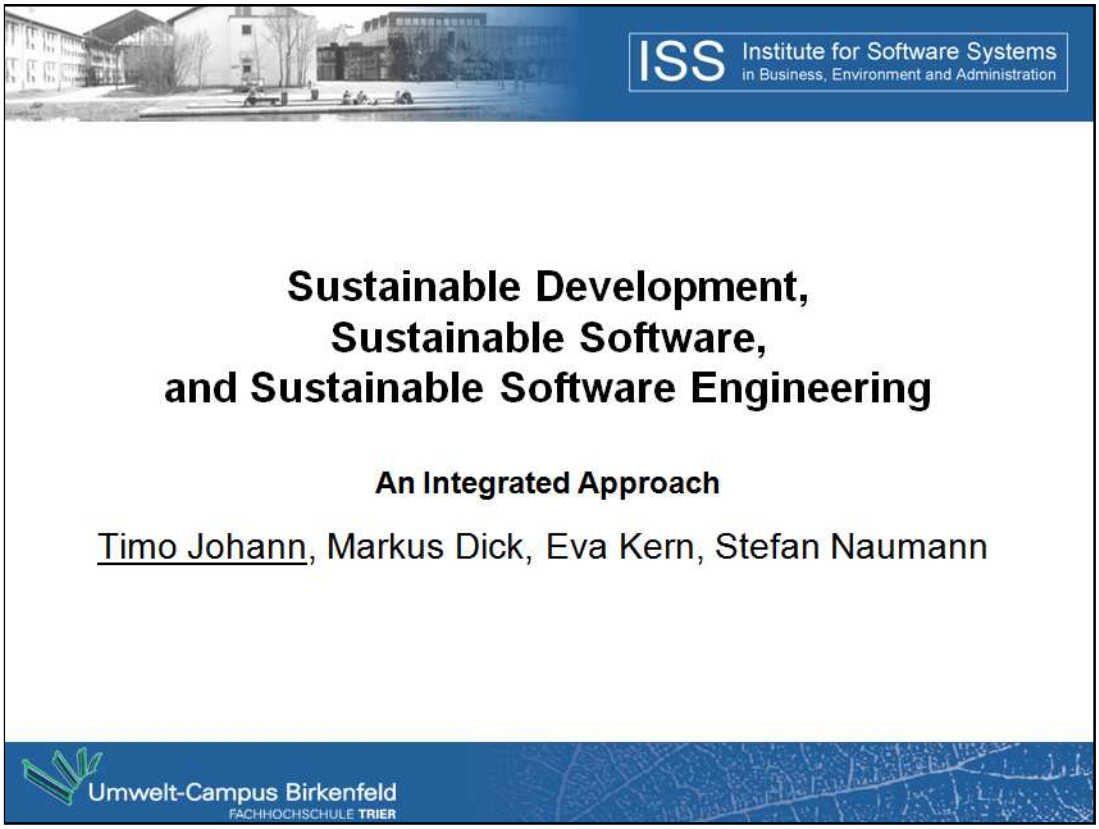


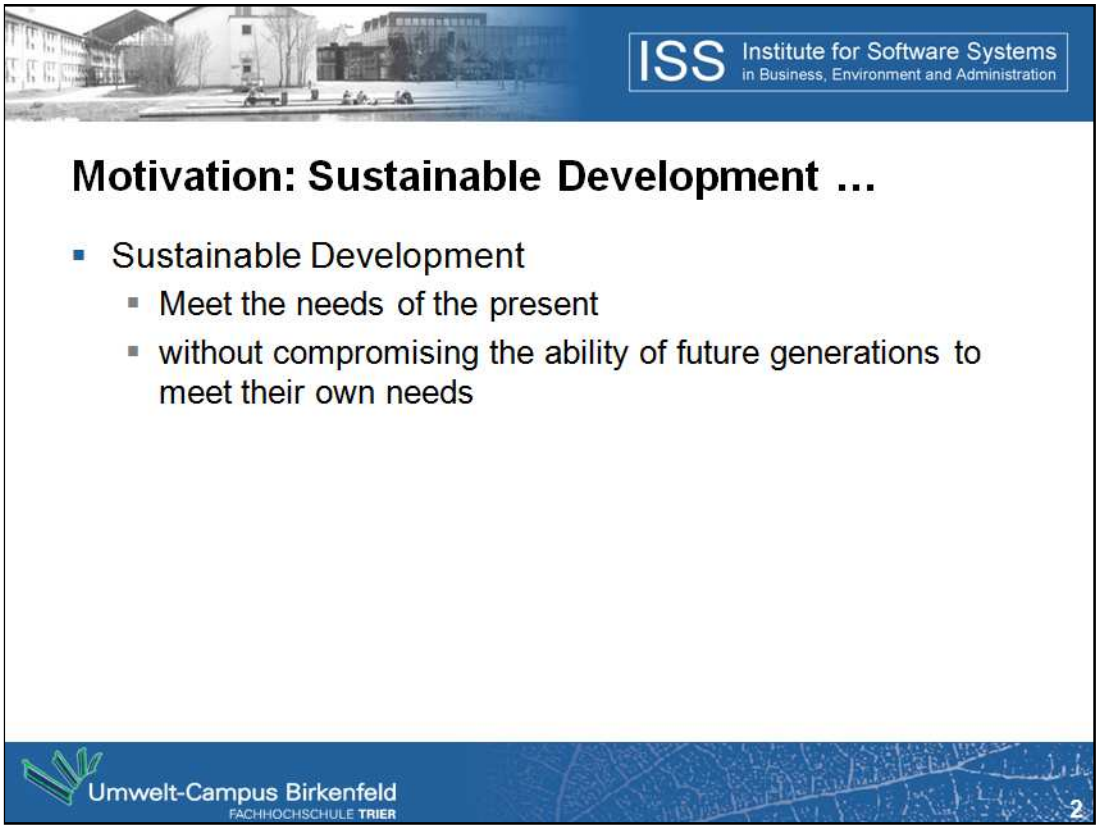
Timo Johann, Markus Dick, Eva Kern, Stefan Naumann
{t.johann, m.dick, e.kern, s.naumann}(at)umwelt-campus.de

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

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

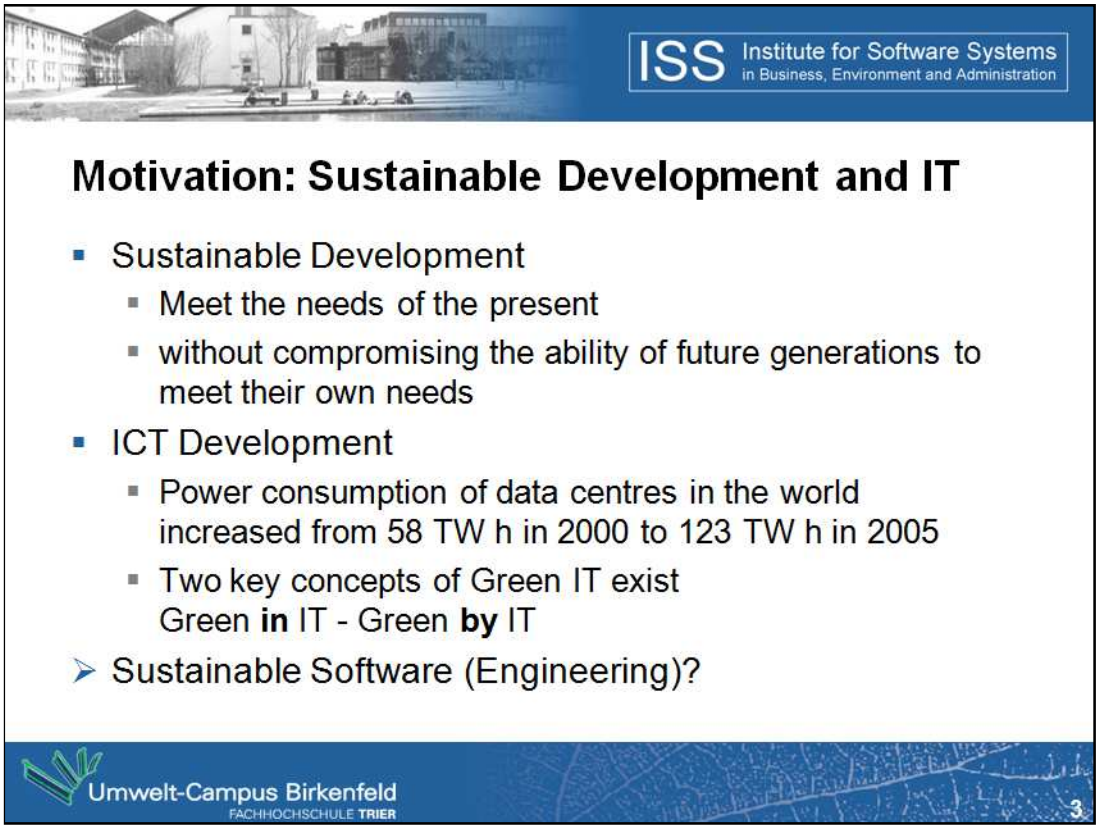
This presentation corresponds to the following paper:
Johann, Timo; Dick, Markus; Kern, Eva; Naumann, Stefan: ***Sustainable development, sustainable software, and sustainable software engineering: An integrated approach***. In: IEEE (Hrsg.): SHUSER 2011. 2011 International Symposium on Humanities, Science and Engingeering Research, June 6-7, 2011, Kuala Lumpur, Malaysia, 2011, S. 34–39.

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 most widely used definition of Sustainable Development is the definition of the Brundtland report, which says Sustainable Development is a development that meets the needs of the present without compromising the ability of future generations to meet their own needs.

Based on that, the question why IT is important for a Sustainable Development arises.



In Information and Computer Technology SD is basically discussed in fields of Green IT :

Where, for example, the energy consumption of computer hardware is considered or in the context of a sustainable acquisition, you talk about "Green **in** IT". There exist also software products which for example make processes more efficient which leads to lesser resource consumption. In these cases you talk about "Green **by** IT".

What we try to do is to broaden the sustainable aspects of ICT to other fields of computer science, in particular to software engineering and software, because efforts exist in the field of computer hardware but there is a lack of efforts in the field of computer software.

Therefore, one needs a holistic view which considers the whole lifecycle of a software and involves all actors in this lifecycle.

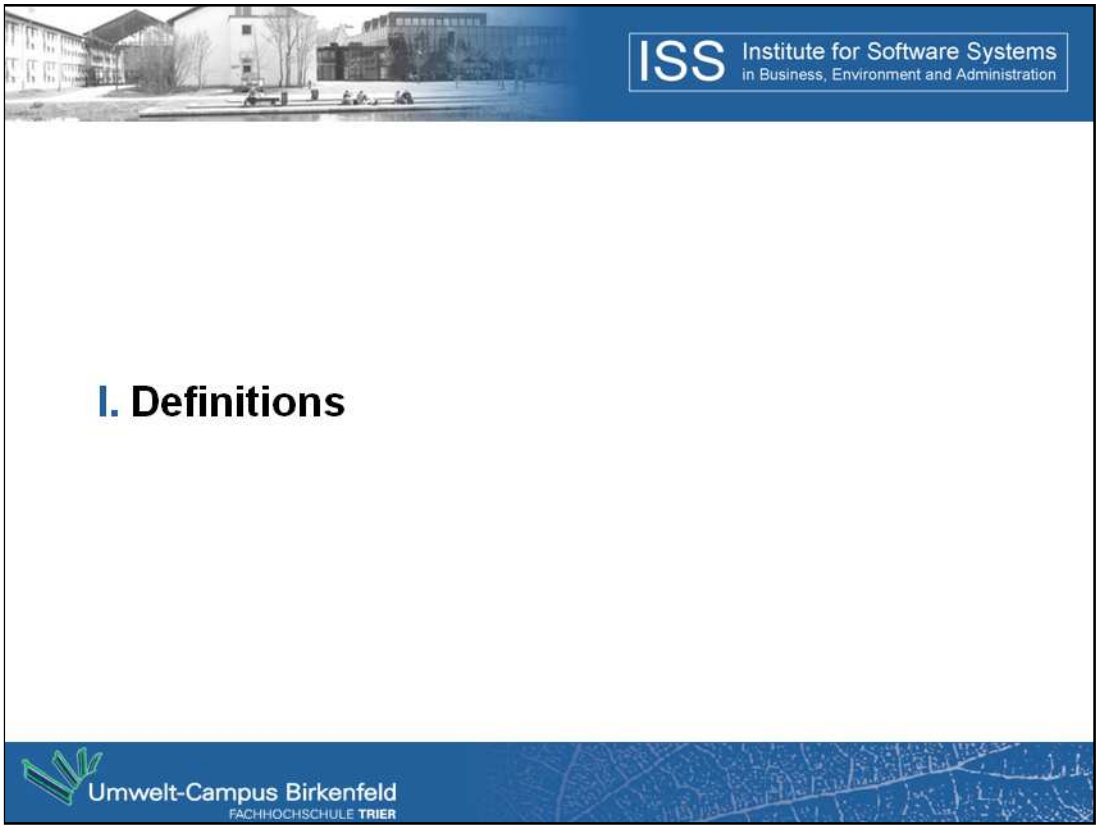
Outline

I. Definitions (Green and Sustainable Software and SE)

II. Prospects of Realization of Green and Sustainable Software

III. Recommendations for Actions and Tools

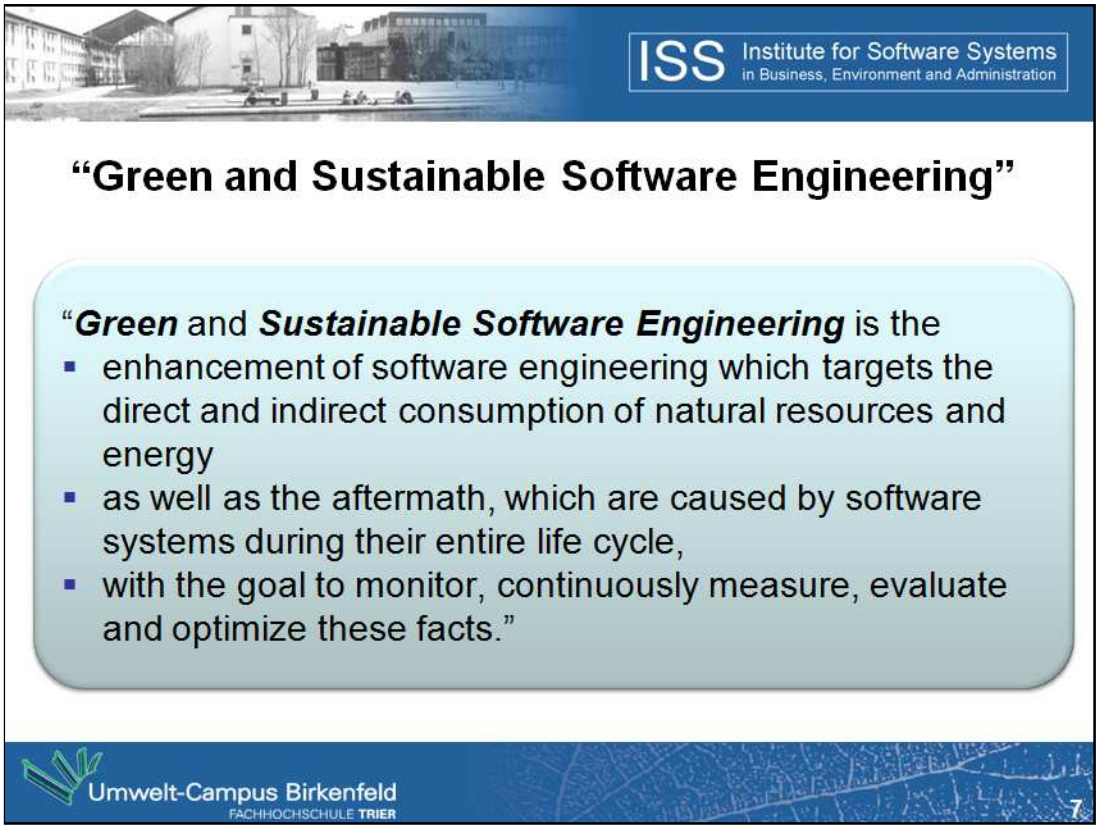
IV. Summary & Outlook


ISS
Institute for Software Systems
in Business, Environment and Administration
I. Definitions
Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

"Green and Sustainable Software"

"***Green*** and ***Sustainable Software*** is *software* whose

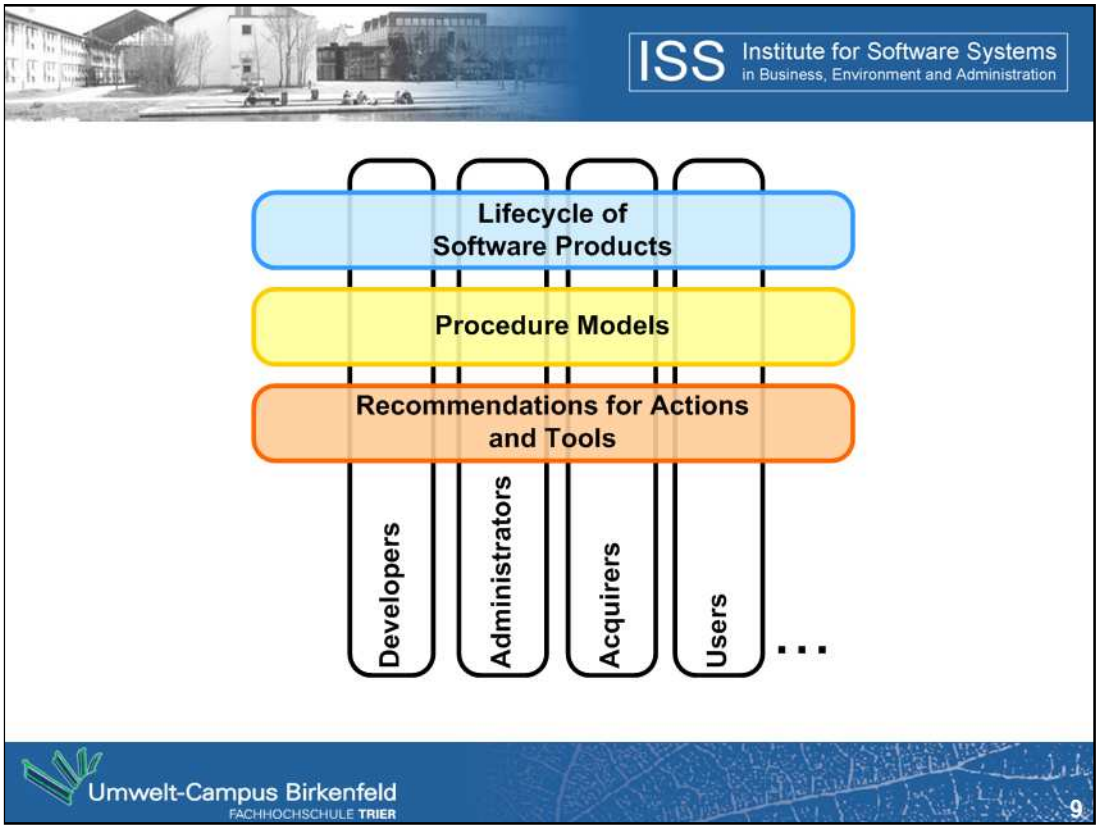
- direct and indirect consumption of natural resources, which arise out of deployment and utilization, are monitored, continuously measured, evaluated and optimized already in the development process,
- appropriation and utilization aftermath can be continuously evaluated and optimized,
- development and production processes cyclically evaluate and minimize their direct and indirect consumption of natural resources and energy"



In these definitions, we understand direct impacts as energy and resource demand that is necessary to produce, use and dispose the software product.
Indirect impacts are effects that result from using the software product on other processes, and long term systemic effects resulting from software usage.

Development, deployment, usage, and disposal address the whole lifecycle of a software product in analogy to ordinary "non-virtual" product lifecycles.
The goal to monitor, continuously measure, evaluate and optimize these facts.

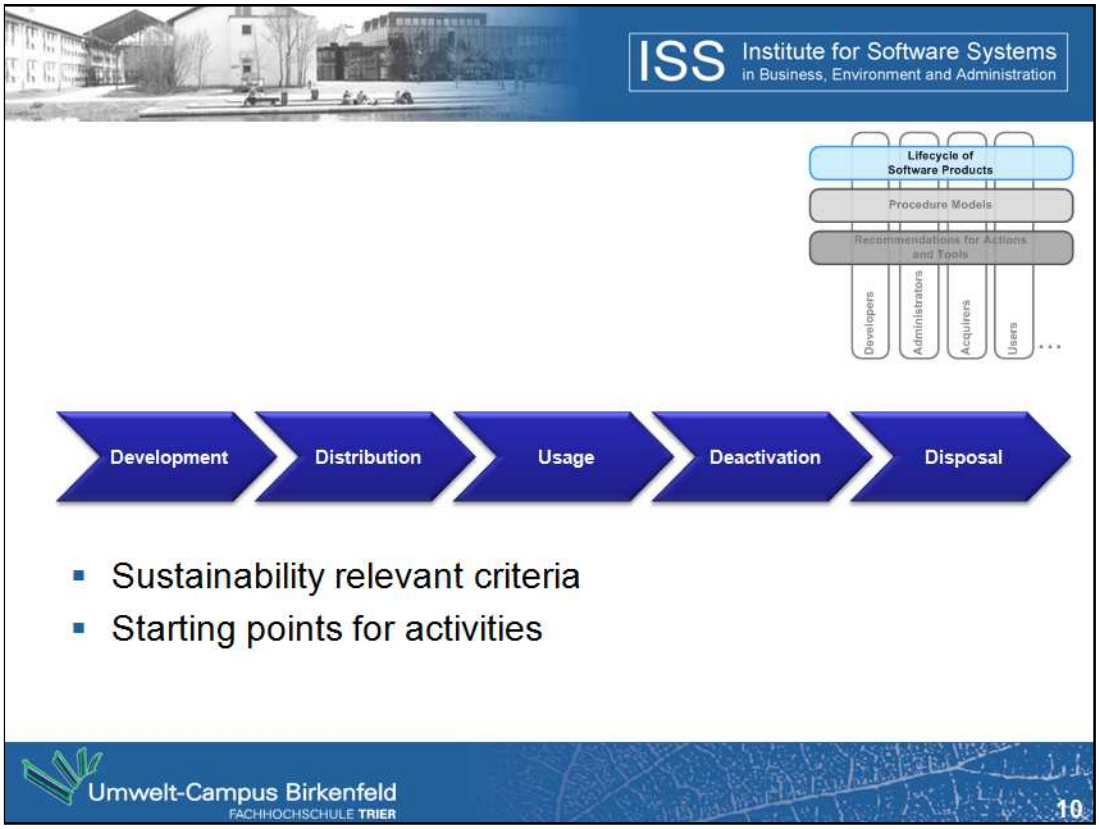

ISS
Institute for Software Systems
in Business, Environment and Administration
II. Prospects of Realization
of Green and Sustainable Software
Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER



Therefore we need processes and methods and especially involve all actors.

Here you can see our reference model, containing three levels, whereas different stakeholders are considered in every level.

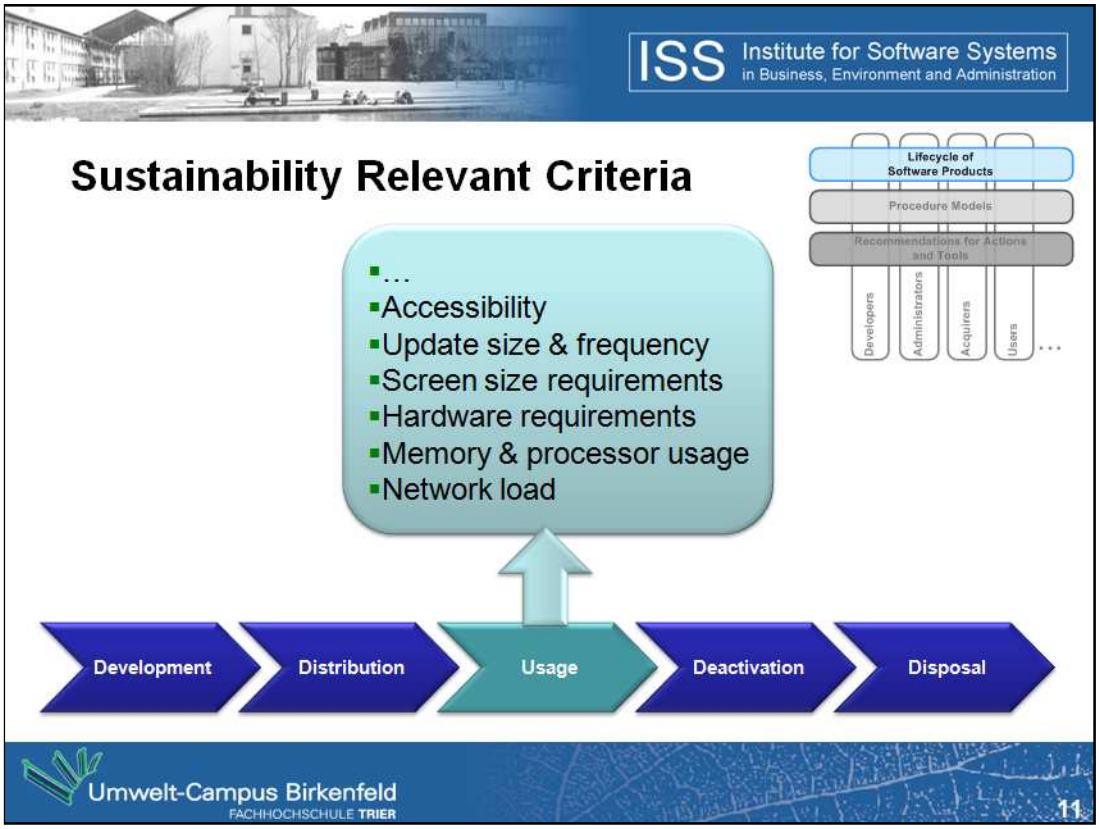
As you can see we consider all actors with their different roles on each level of our model because they all have the ability to act, so they need to be aware of impacts on sustainability in every stage.
The matrix shows the relation between the different user roles and the levels. As you can see the three different levels regard all of the user roles.



Starting with the **Lifecycle of Software Products**, I will go into the different levels.

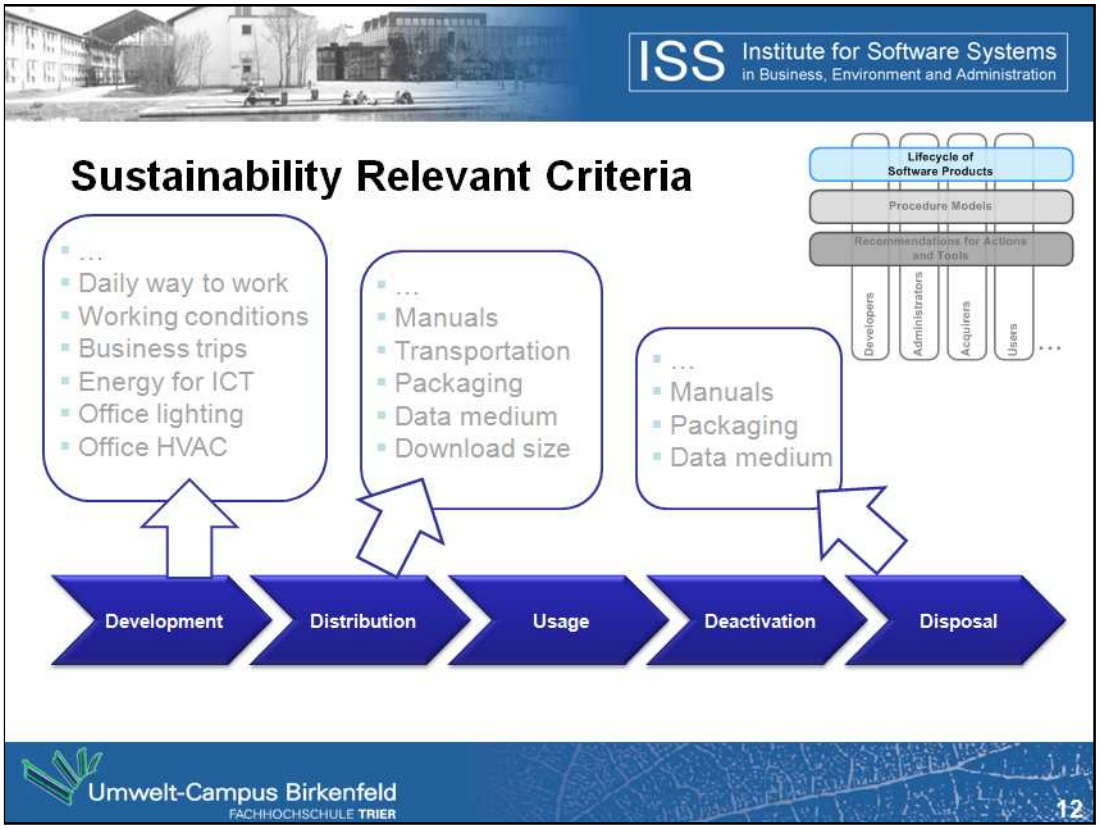
If you take a look at the phases of our lifecycle model, you will maybe know this from standard product lifecycles in the sense of Life Cycle Thinking (or "cradle-to-grave approach").
We tried to port this approach on software products and as you see this model considers more than an ordinary software lifecycle or software development process, which are usually focusing only on development phases and development activities.

This lifecycle model has two objectives:
• Its first objective is to assign criteria to the different lifecycle phases that lead to or result in sustainability relevant effects.
• Its second objective is to provide starting points for activities that hopefully lead to more sustainable software products.

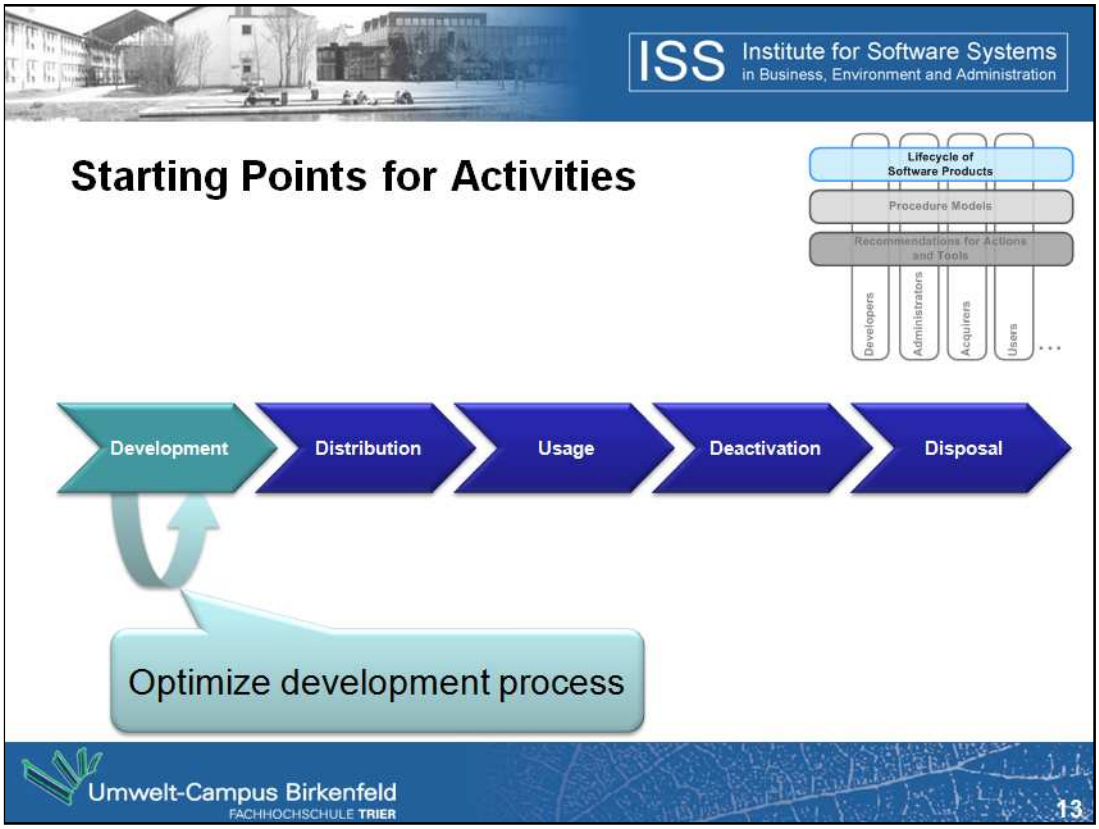


Examples of criteria for the usage phase are:

- Accessibility issues
- Screen size requirements
- Hardware requirements
- Memory and processor usage during program execution

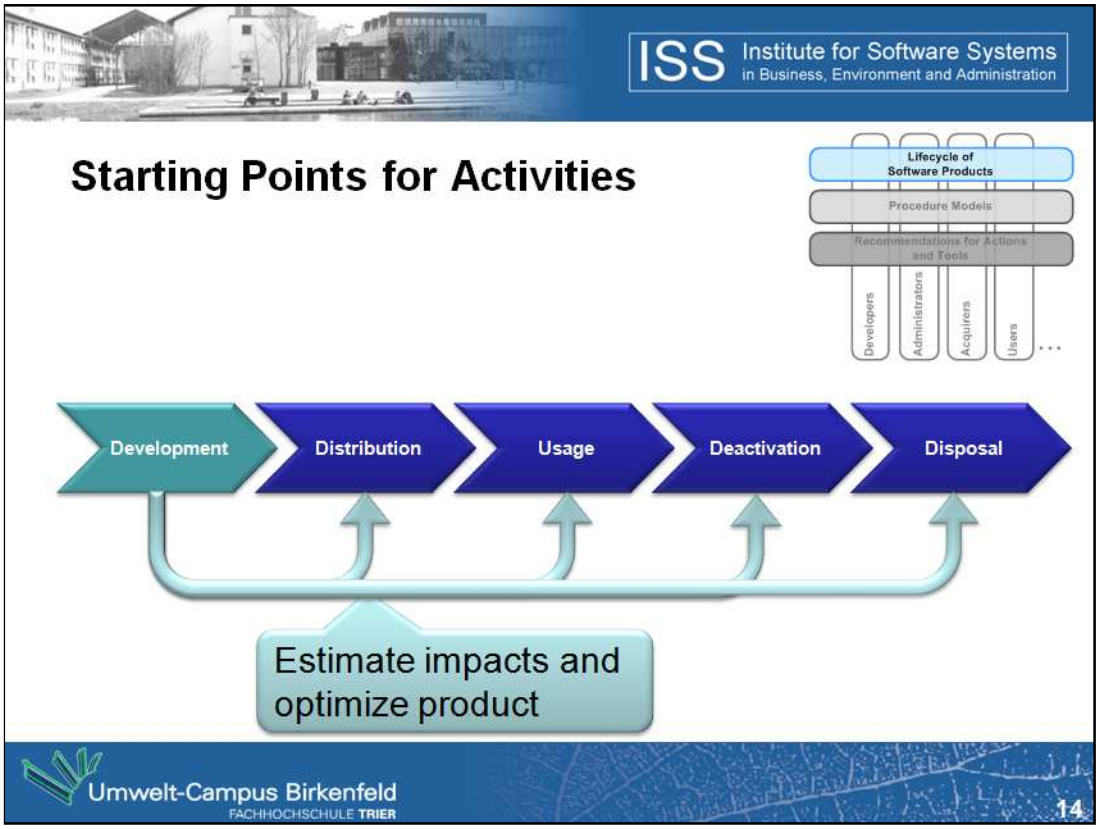


The shown image should just give you an impression on the possibilities for criteria that can occur in the several phases.
Please, note that these criteria are just examples and are far from complete.



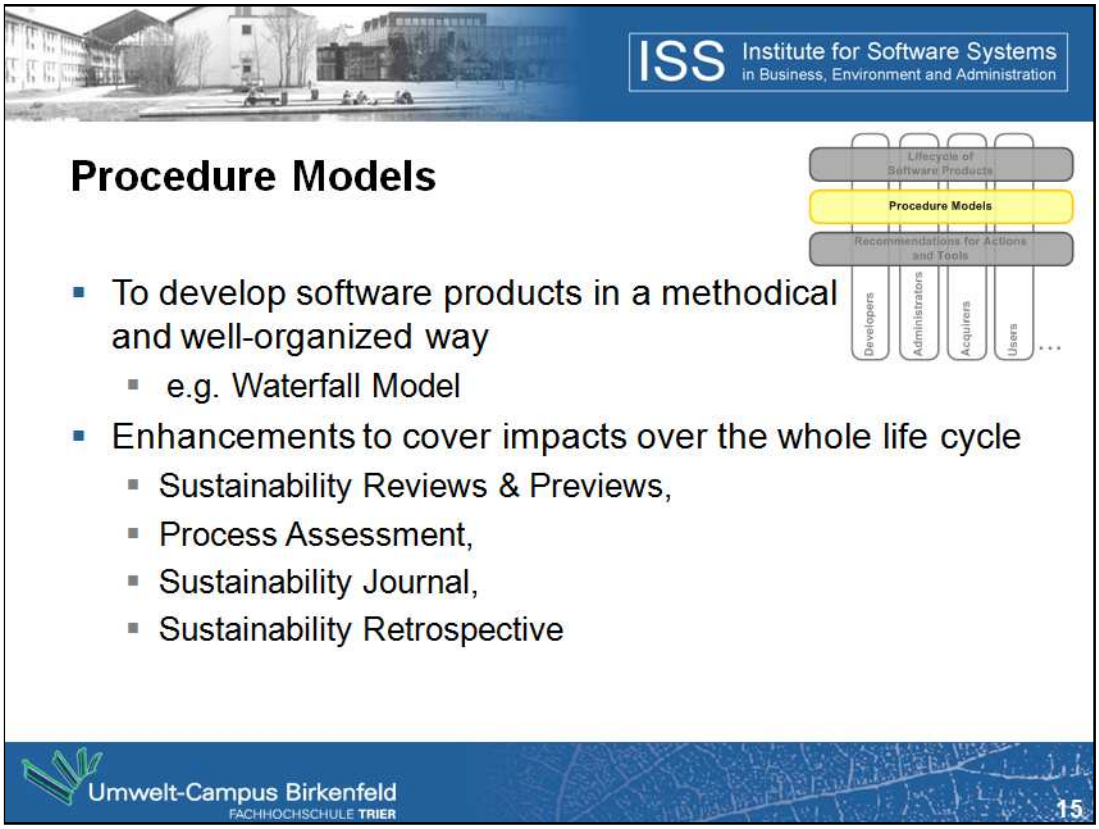
In this model we are focusing on two starting points:

First, according to our given definitions, the software development process itself should be optimized in order to mitigate negative impacts or to enforce positive impacts that result from it.



Second, during development, the positive or negative impacts that are expected to arise from distribution and future use of the software product should be continuously anticipated, assessed and reflected by involved actors.

The outcomes of these assessments and reflections should then be used to take action towards a more sustainable software product.



That leads us to the next level: **The Procedure Models**

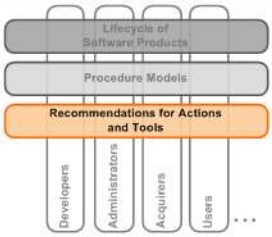
Here we extended a software development process.

Typical process phases are: requirements analysis, design, implementation, testing, operation and maintenance. A well-known model including these phases is the Waterfall Model. Even if other models exist, most of them contain these phases.

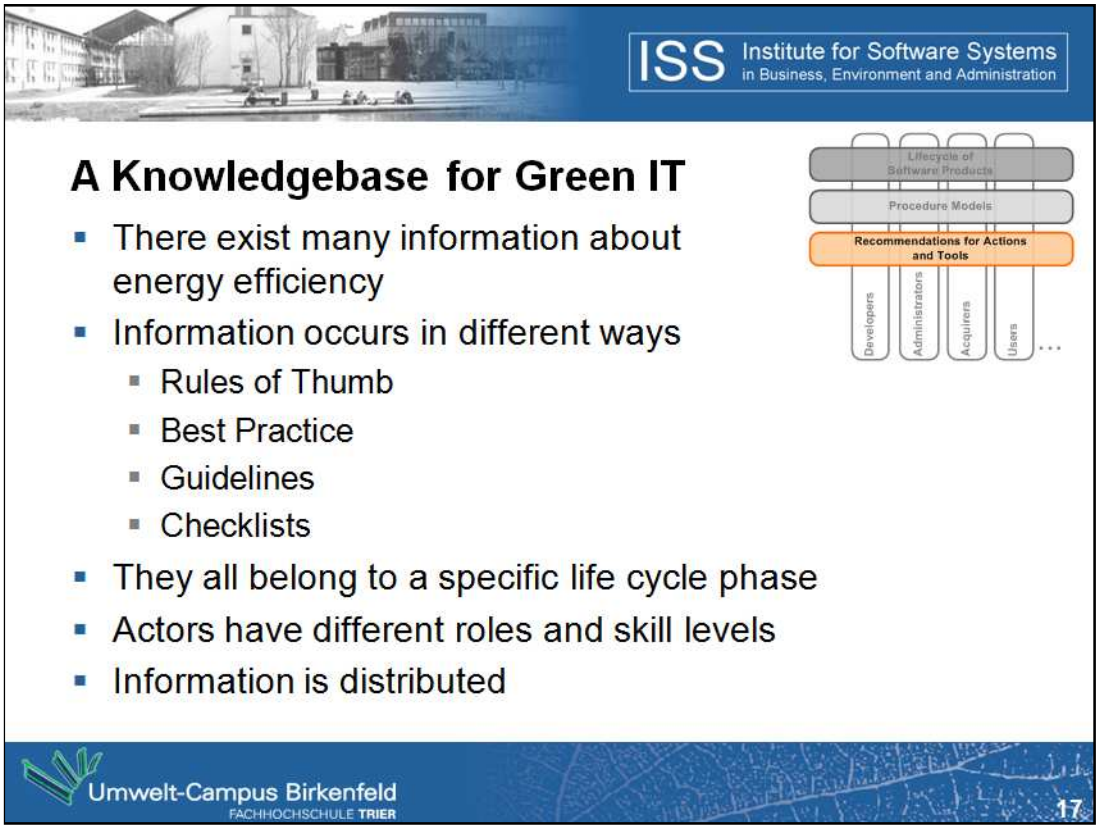
As an example for a procedure model we used a simplified "waterfall-like" software development process, which is generic enough to be tailored to other known process models like Scrum or Rational Unified Process

This general process is enhanced by several activities that have the objective to enable sustainable software engineering.
These activities are:
- Sustainability Reviews & Previews for every phase,
- Process Assessment, and
- the Sustainability Retrospective

Process Assessment covers the first starting point (the optimization of the process itself) , whereas Sustainability Reviews & Previews cover the second starting point (to estimate the products impacts). Finally, the Sustainability Retrospective combines them, in order to cover impacts over the whole lifecycle of the software product.



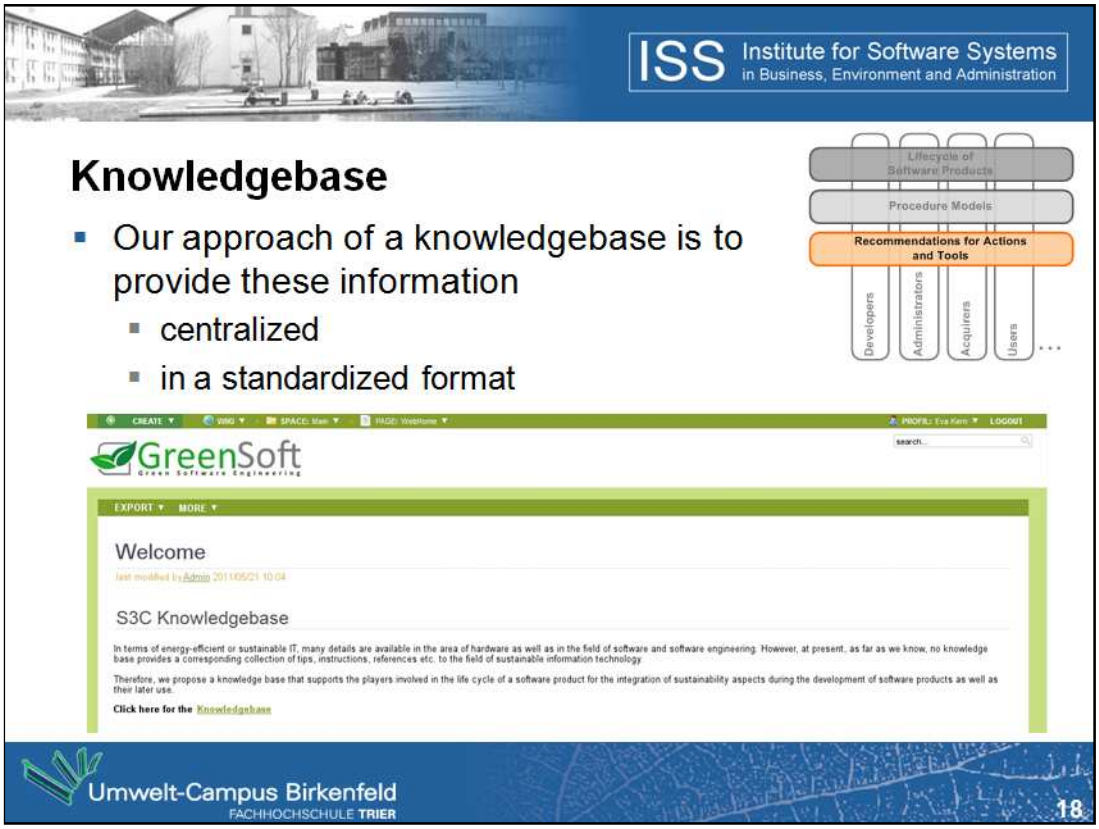
III. Recommendations for Actions & Tools



At the end lets have a look at the last part of our reference model:
Recommendations for Actions and Tools.

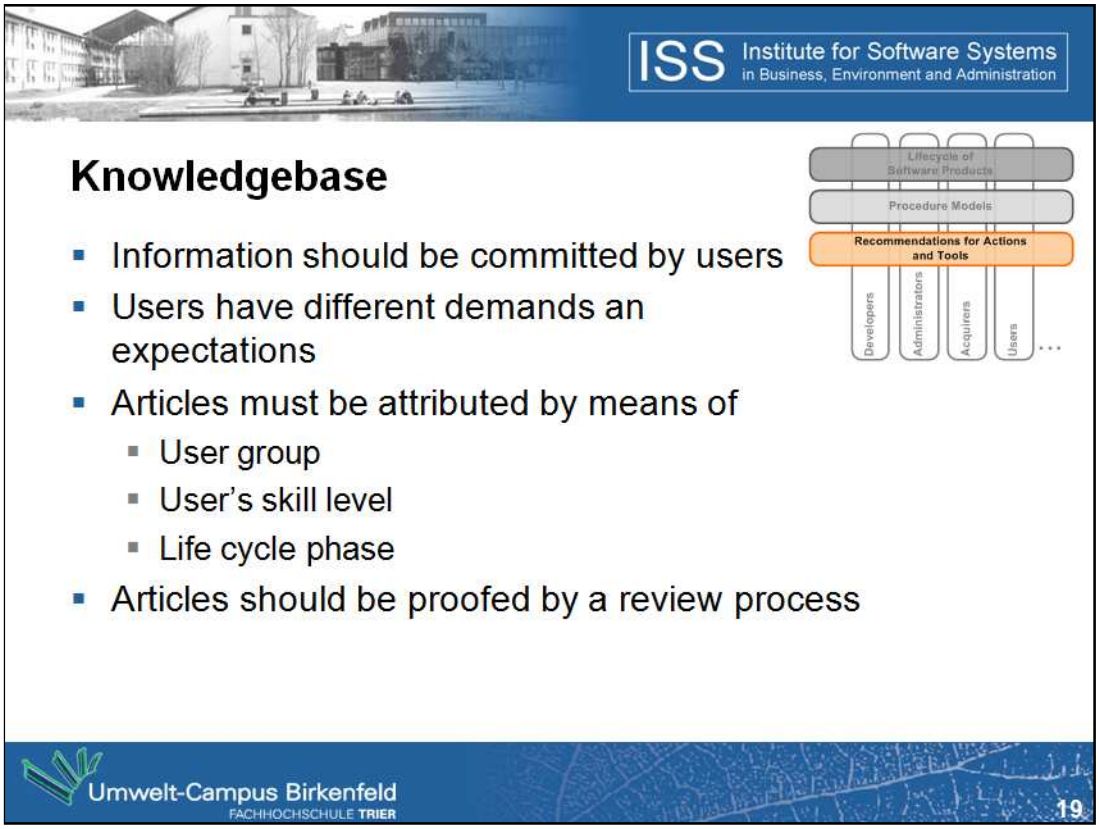
To realize the mentioned sustainable requirements one needs tools which support the actors.

Due to the fact that sustainable issues are not well known yet we are developing a knowledgebase to provide recommendations for actions.
There exist many different information about energy efficiency in form of rules of thumb, hint, best practices etc. These information are distributed for example in the web and not every actor has the same skill to handle these information.
They are neither assigned to roles nor to lifecycle phases



That is why we develop a knowledgebase which collects these information in a proper well defined manner:
Centralized and in a standardized format

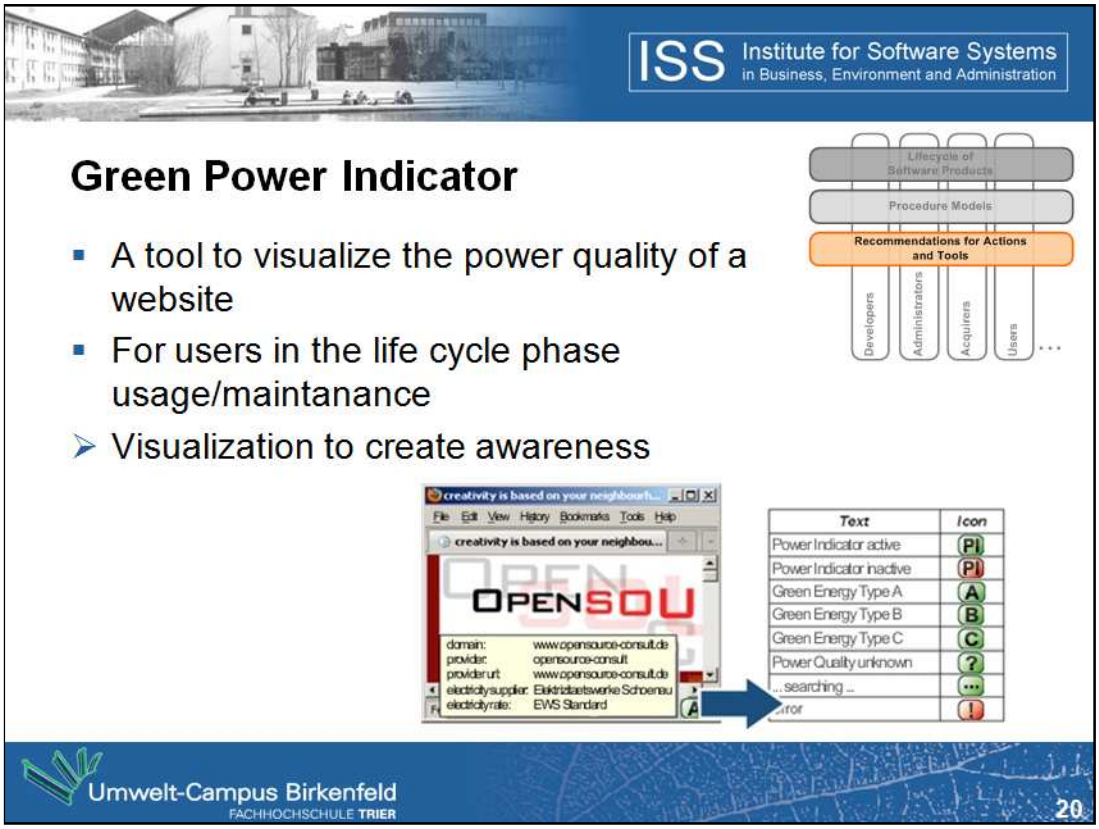
Here you can see a screenshot of the prototype.



Another important requirement is that the knowledgebase is open for everyone who wants to contribute to it.
Due to the reason that users have different demands and expectations articles must have meta information like

- for what user group and what skill level is it meant
- to what life cycle phase does it belong.

And article committed should be proofed to guarantee the quality and consistency of the knowledgebase.



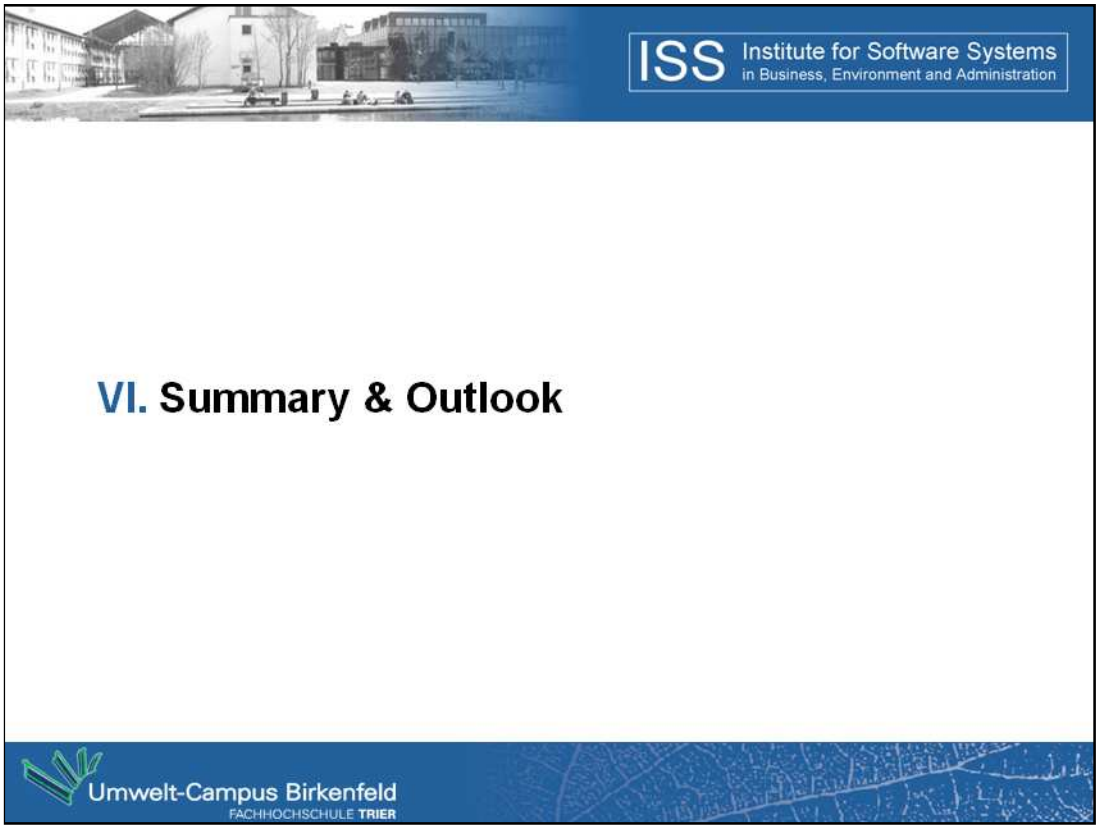
To create awareness it is important to visualize to users what is happening.

Our Firefox Add-on "Green Power Indicator" displays whether the called site is hosted on a server, which is operated with environment-friendly produced electricity. It is provided as a free download.

If a user is aware of the energy quality of the provider they will demand green energy. It might let the provider and administrator think about using environmental-friendly produced electricity.

A good example for this kind of user awareness is the campaign of Greenpeace "Facebook: Unfriend coal" where thousands of participants want to push Facebook to use clean energy.

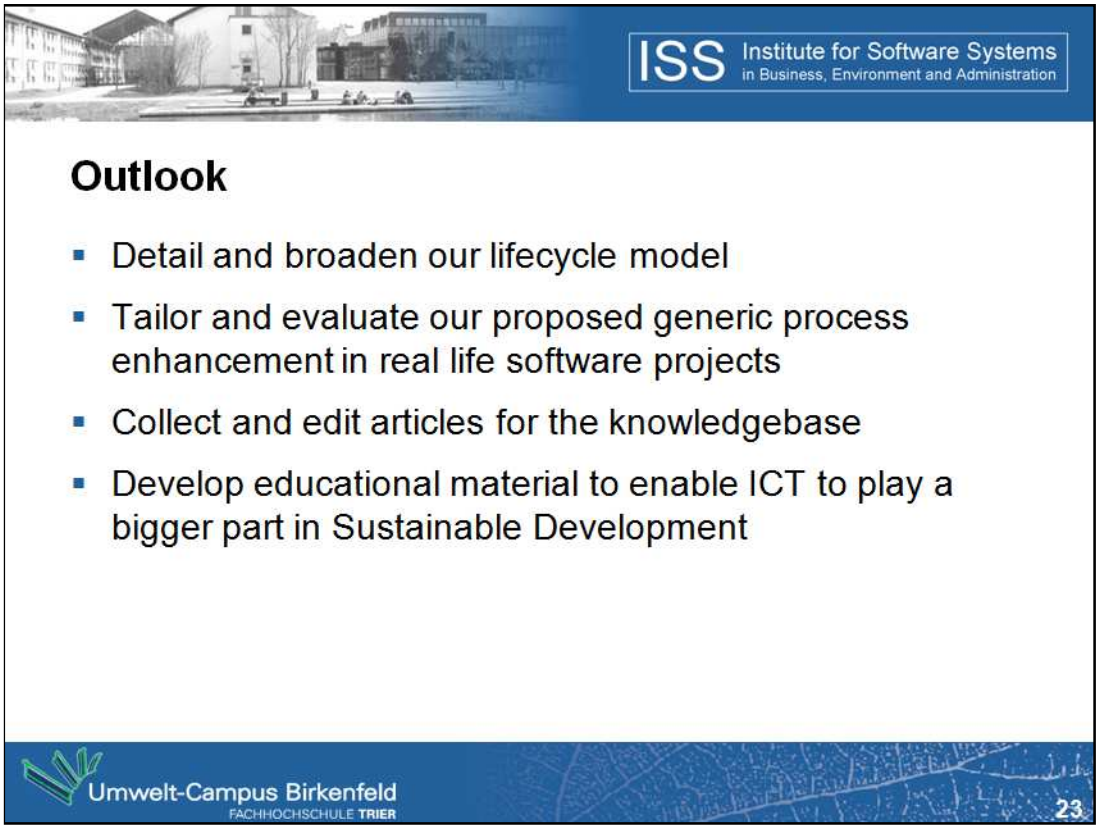
This shows how important it is to educate users about sustainable relevant facts.


ISS
Institute for Software Systems
in Business, Environment and Administration
VI. Summary & Outlook
Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

Summarizing, it is currently not clear whether energy savings through Information and Communication Technology (ICT) outbalance its energy consumption or not.
In either case it is rational to integrate sustainability aspects into software product design and development as it is already common today for material products, like cars, light bulbs or computer hardware.

It is very important to educate all actors in software lifecycles and support them with recommendations and tools

Hence, we presented
• a lifecycle thinking inspired lifecycle model for software products,
• a generic process model for Sustainable Software Engineering that can be tailored to fit arbitrary software development process models
• and two examples for tools to support different user groups with information about sustainable software



Our next steps are to detail and broaden our model with e.g. more criteria that addresses the social and economic dimensions of sustainability

Furthermore we plan to

• tailor and evaluate other procedure models with our sustainability enhancement in real software projects. Whereas procedure models also exist for other phases than only the development phase,

• collect and edit articles for the knowledgebase (of course everyone is invited to contribute with articles and ideas),

• develop educational material to enable ICT to play a bigger part in Sustainable Development because one of the most important fact for a Sustainable Development is the awareness of the people.

In general the question is how can ICT help to increase the sustainable aspects.

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