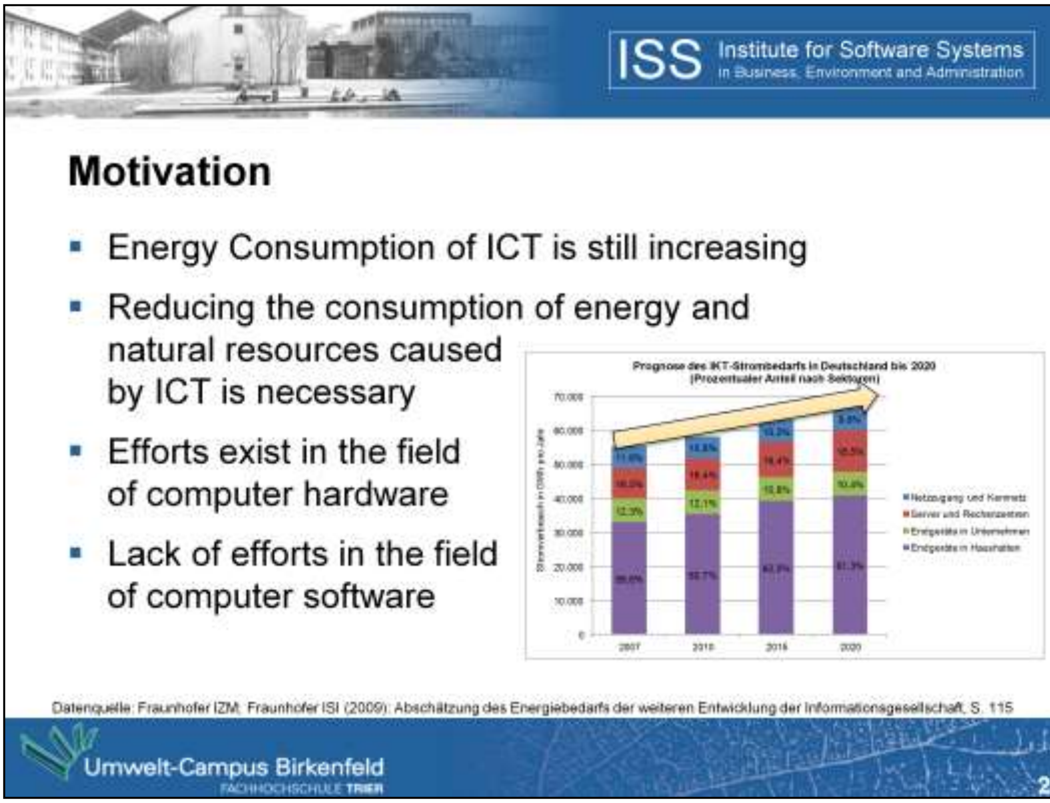
The project “Green Software Engineering” (GREENSOFT) is sponsored by the German Federal Ministry of Education and Research under reference 17N1209. The contents of this document are the sole responsibility of the authors and can under no circumstances be regarded as reflecting the position of the German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



The energy consumption of ICT 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 into software product design and development, as it’s common today for material products like cars, light bulbs or computer hardware.
Furthermore, well-known textbooks on software engineering like Sommerville or Balzert that are used in lectures don’t even mention sustainability aspects of software.
Additionally, up to now it is not clear what the terms “sustainable software” and “sustainable software engineering” mean or what they are.

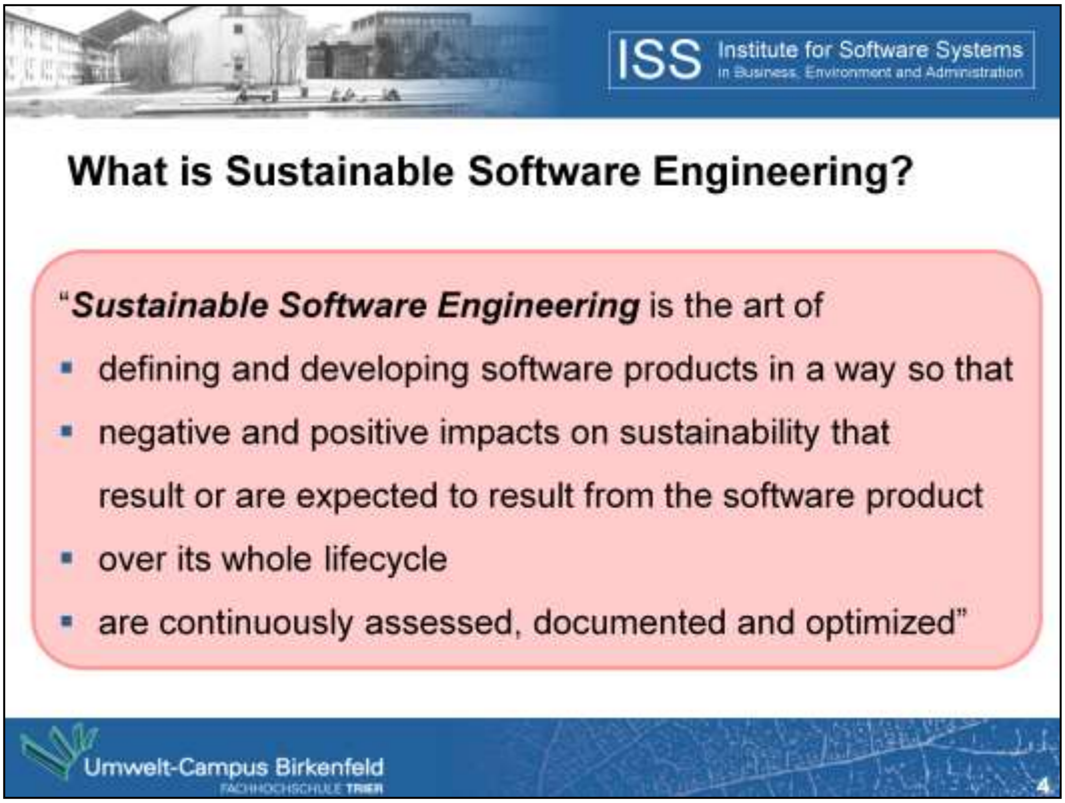
Outline

I. What is Green and Sustainable Software Engineering?

II. A Generic Model for Sustainable Software Engineering

III. Influences of Software on Energy Consumption

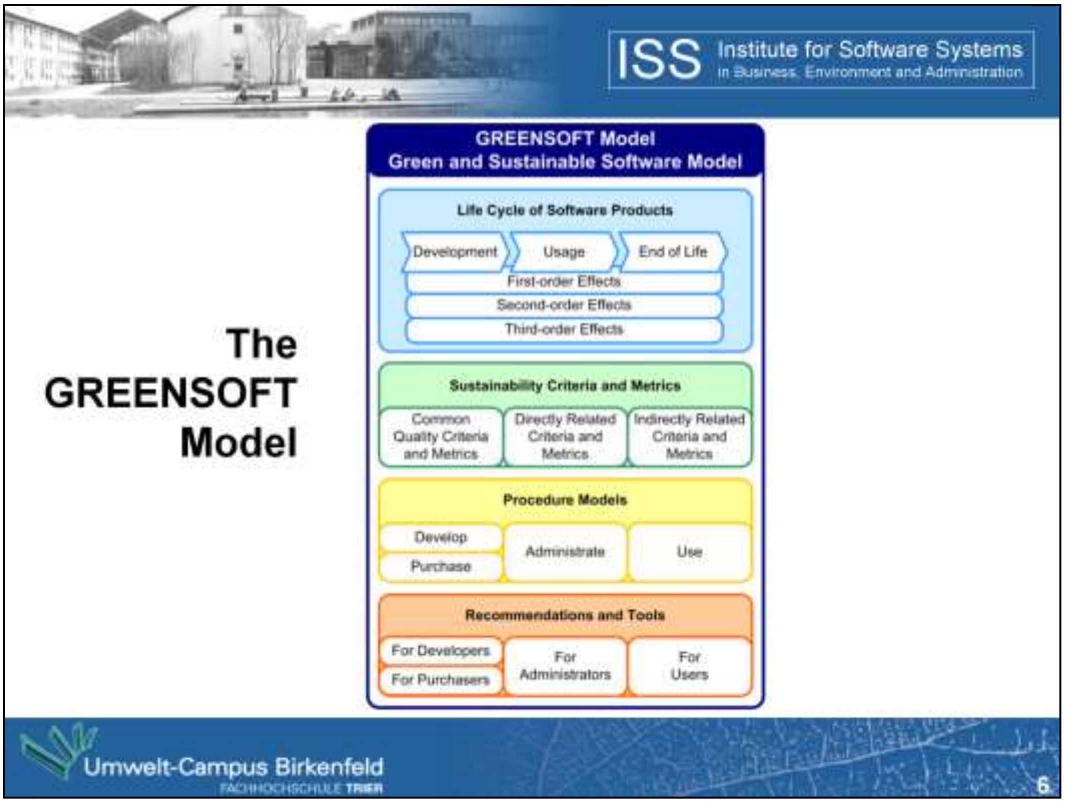
IV. Summary



In this definition we understand direct impacts as energy and resource demand that is necessary to produce, use and dispose the software product.

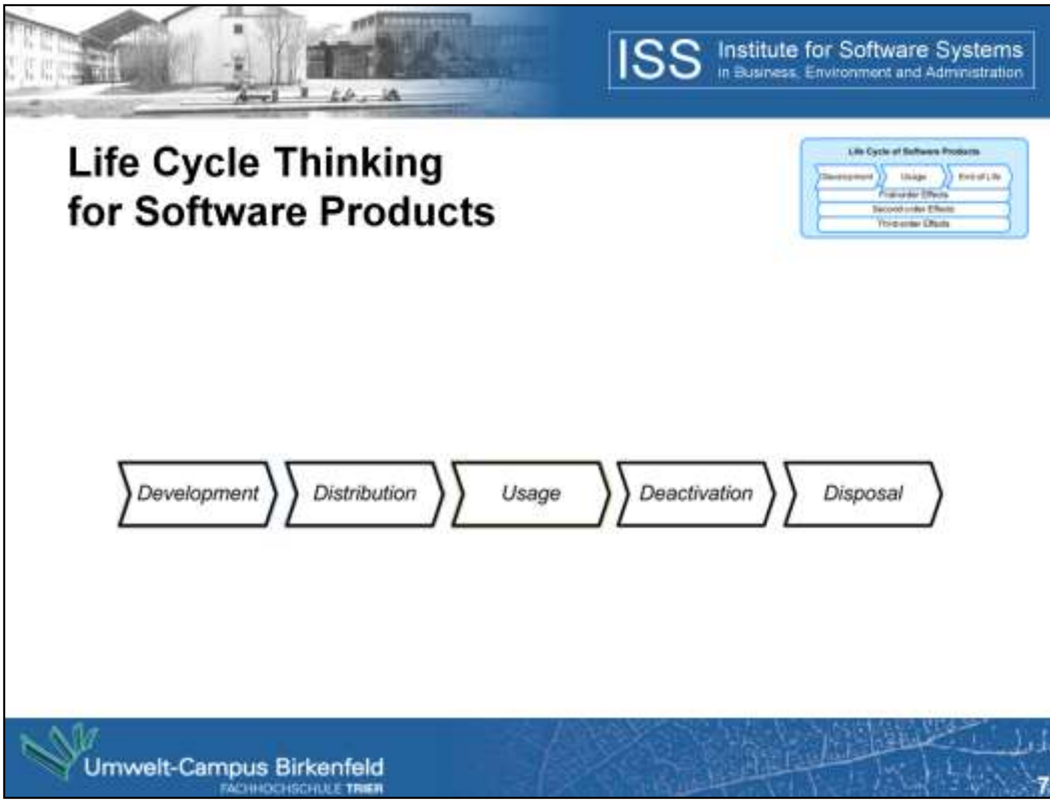
Indirect impacts are effects that result from using the software on other processes…

II. A Generic Model for Green and Sustainable Software Engineering



The GREENSOFT Model consists of four parts:

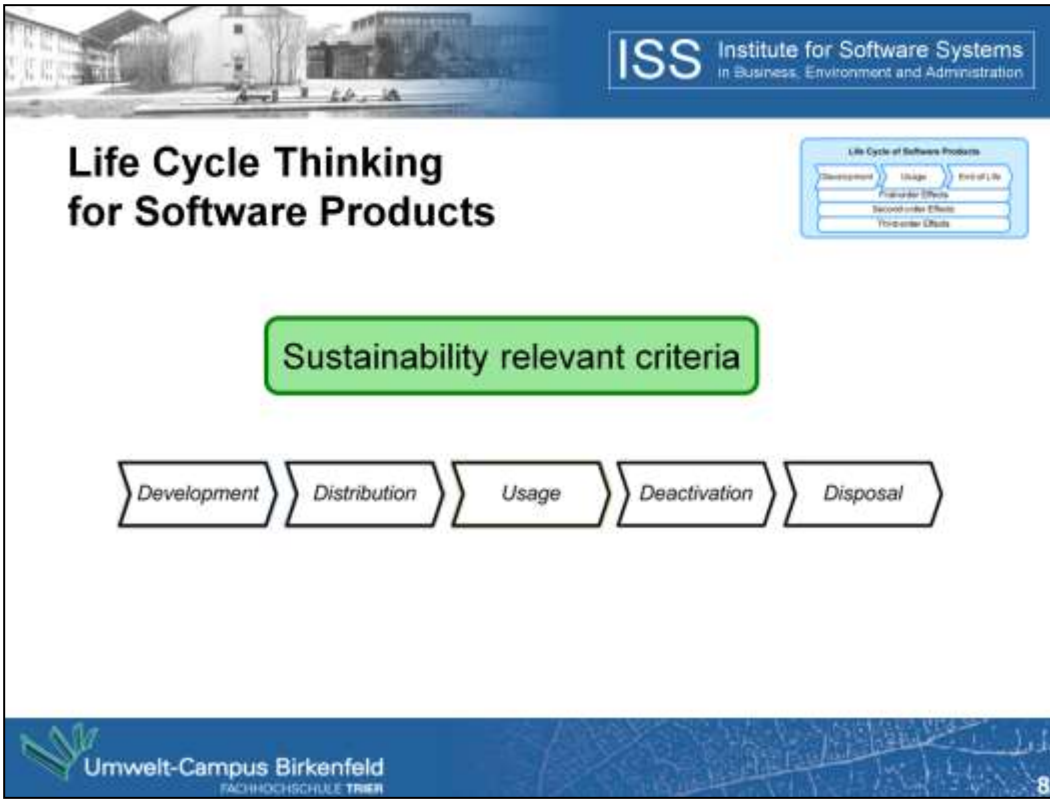
This is just an overview of the different aspects, let's see how we do understand these four sub models:



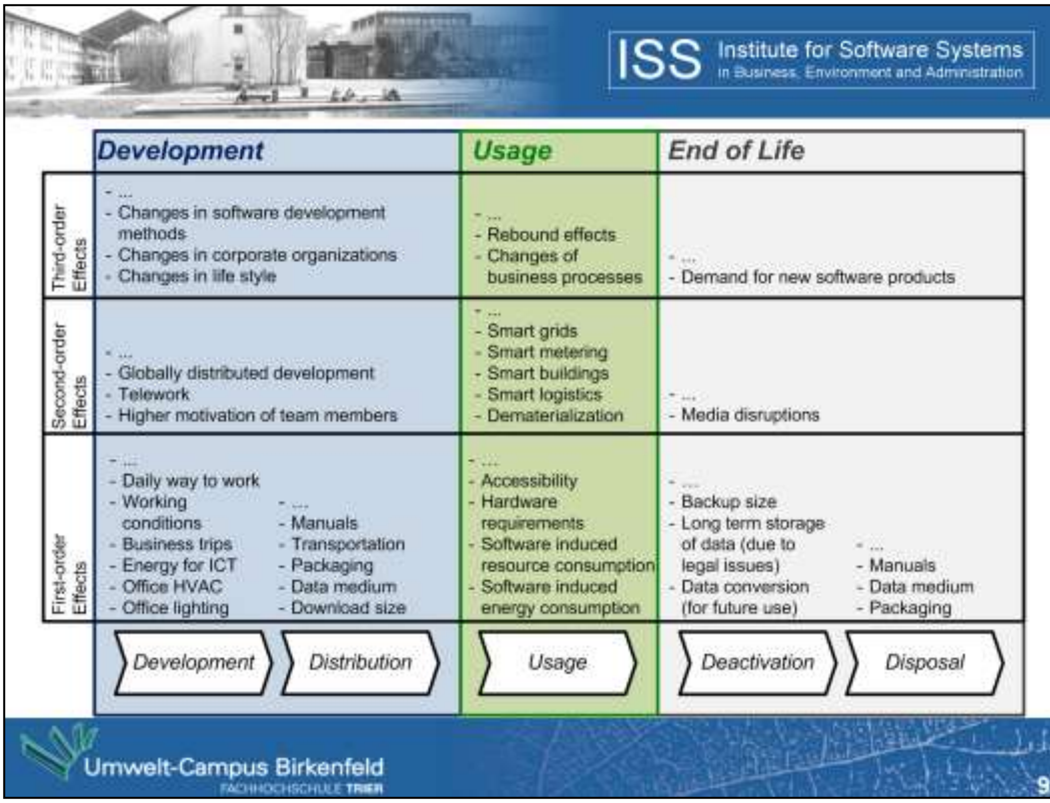
First, we want to point out that also software products do have a life cycle.

If you glimpse at the phases of our life cycle model, you will recognize that it's more a product life cycle in the sense of Life Cycle Thinking (or "cradle-to-grave approach") than an ordinary software life cycle or software development process, because these are usually focusing on development phases and development activities.

This model mainly fits standard software products.
For custom software products the phase "Acquisition" follows close upon the phase "Product Definition".



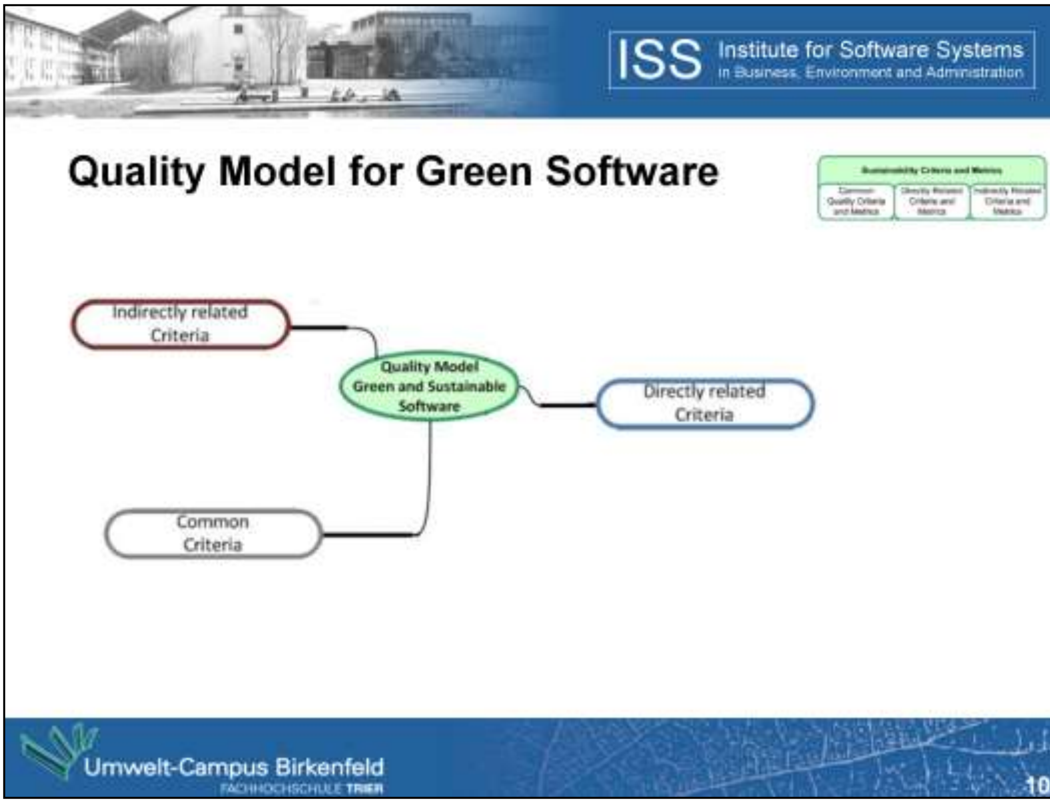
The objective of this life cycle is to assign criteria to the different life cycle phases that lead to or result in sustainability relevant effects.



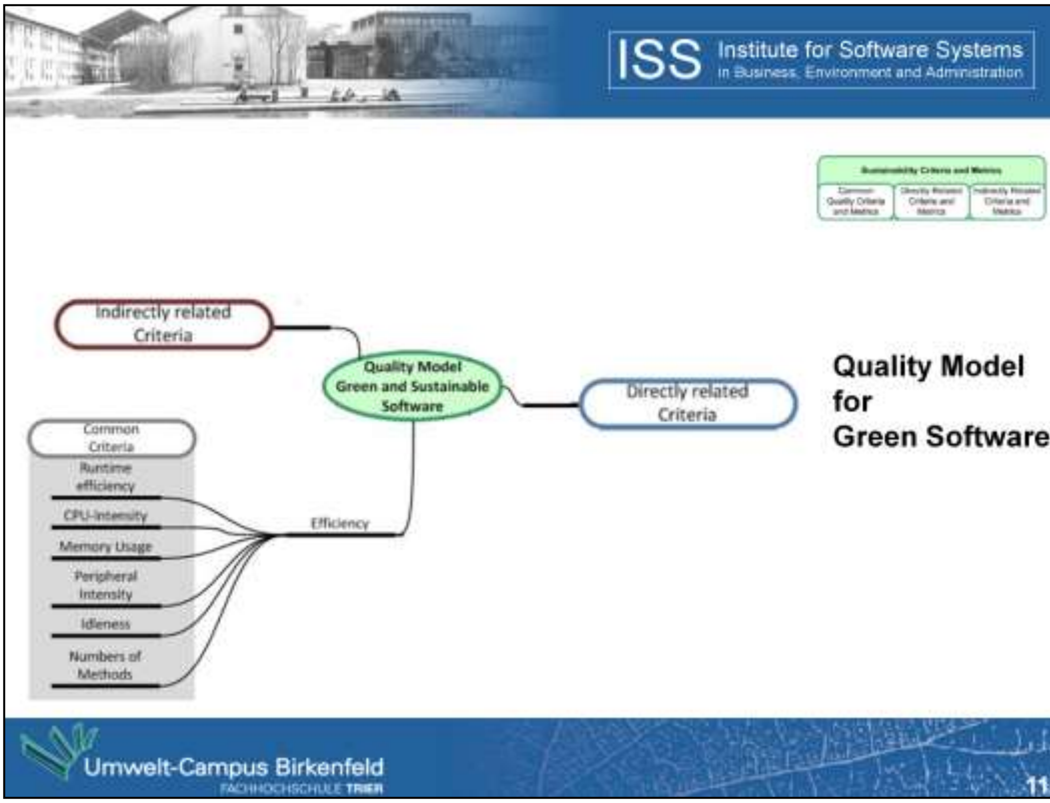
Which effects can software have during its life cycle?

- First order effects can be directly assessed. There are for example the lighting of the office during the development, the download size of the product in context of the distribution and the data conversion for future use in the deactivate phase.
- The second order effects are caused by the usage of the software within the life cycle of other services or products. This is getting more and more important since software is to be find in nearly every products of the daily life. Typical are so called smart grids, smart metering, …
- Very hard to assess are third order effects. One example are Rebound effects. That means there are more resources available caused optimization activities but they will be over consumed so that the actually positive effect is gone

These are effects and impacts of software during its life cycle – but how can criteria and metrics for sustainable software look like?

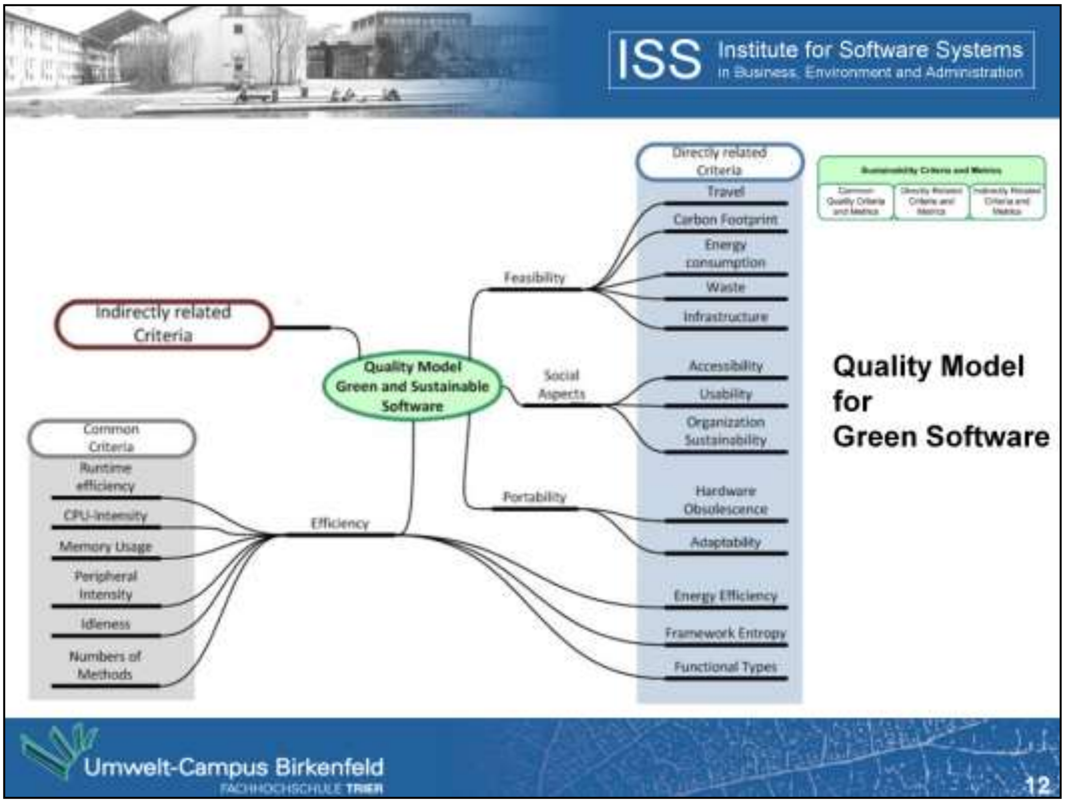


A first approach is the Quality Model I will introduce to you know:
In order to classify and develop sustainable software, criteria are required.
They can be categorized in common, directly and indirectly related criteria:



The common quality criteria arise out of the well-known quality criteria for Software. The model takes aspects into account like Efficiency, Memory Usage, Idleness and Number of Methods.

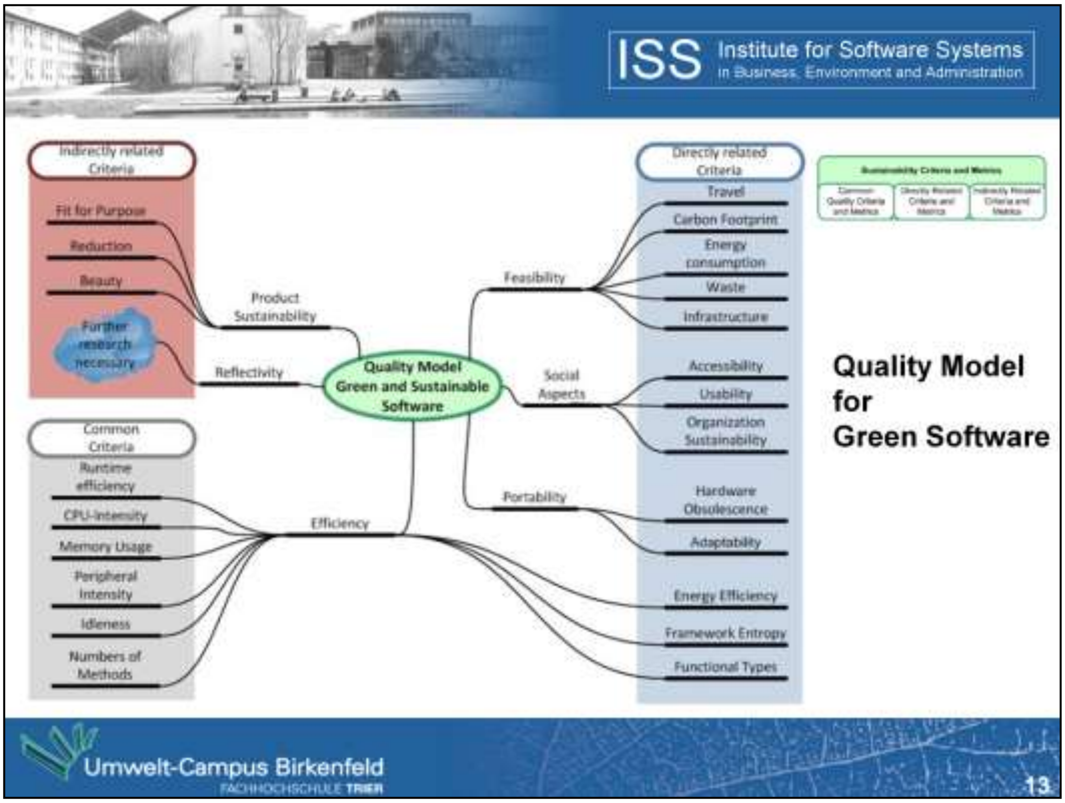
- Idleness: describes how often the system is idle
- Number of Methods: reflects size of application



In contrast to that, Energy Efficiency can be ranked against a reference system and goes to directly related criteria. It is the same for Framework Entropy and Functional Types

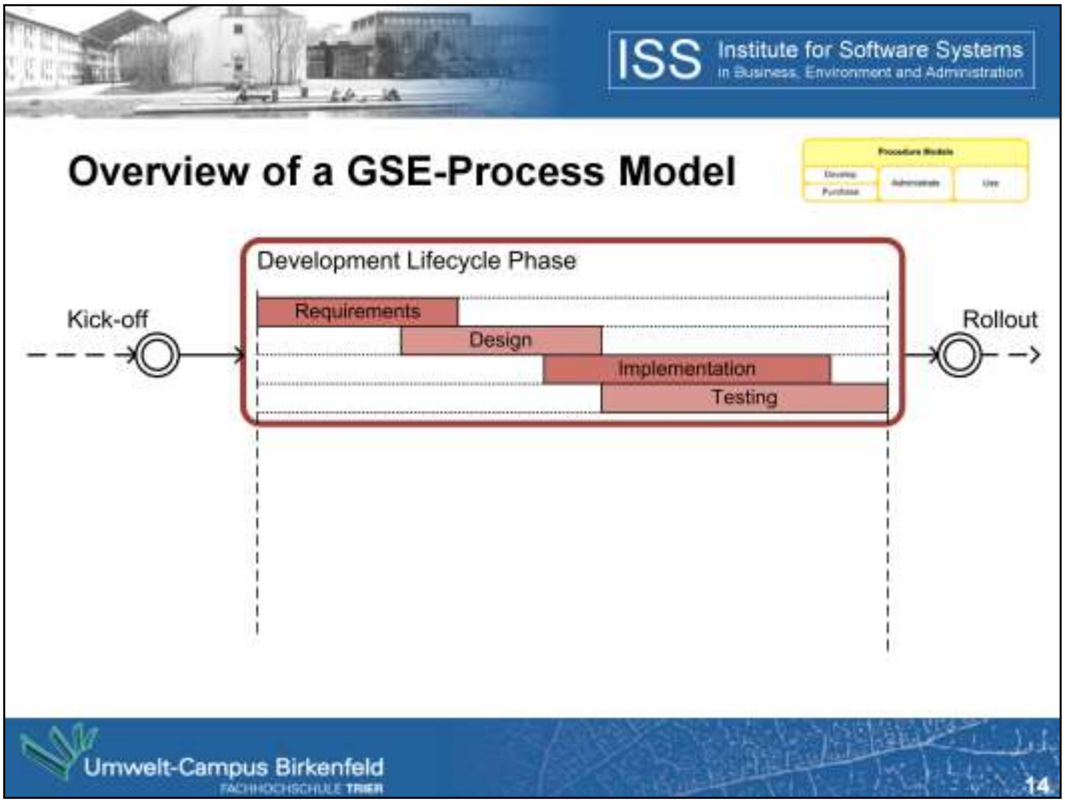
- Framework Entropy: grade of usage of external libraries
- Functional Types have different energy consumption.

Additionally, aspects like Hardware Obsolescence, the Carbon Footprint and Energy consumption go into this category



The indirectly related criteria are the third group. These apply to software which aims to support sustainable development. Hence, Product Sustainability with ist different criteria sums up the effect of software on other products.

Here you can see the parallel to the categories of the life cycle effects: Whereas, first-order effects are directly related to the software product, second and third order effects are mainly indireclty related criteria.

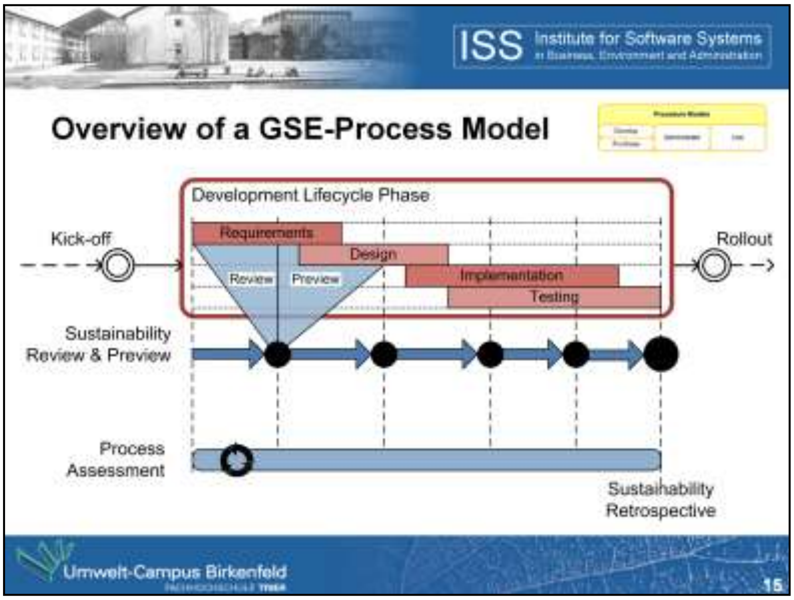


The next part of the GREENSOFT Model are procedure models. In our case we integrated models for the development, purchasing process, administrating and the usage.

Indeed, I will just go into one of these today: the Green Software Engineering process, belonging to the development phase.

For this introductory overview, we're using a simple "waterfall-like" software development process with the exception that the different development phases are executed in parallel rather than strictly sequential.

This is the more general case and enables us to subsume processes like the IBM Rational Unified Process with this model.



This general process is enhanced by several activities that have the objective to enable sustainable software engineering.

– Sustainability Reviews & Previews mainly consider impacts on sustainability which are expected to arise from distribution and future use of the software product.

In this activity, actors take a look at the work done and assess outcomes according to sustainability criteria. This is the review part.

As the preview part, actors develop and realize measures until the next Review & Preview in order to optimize the sustainability of the software product.

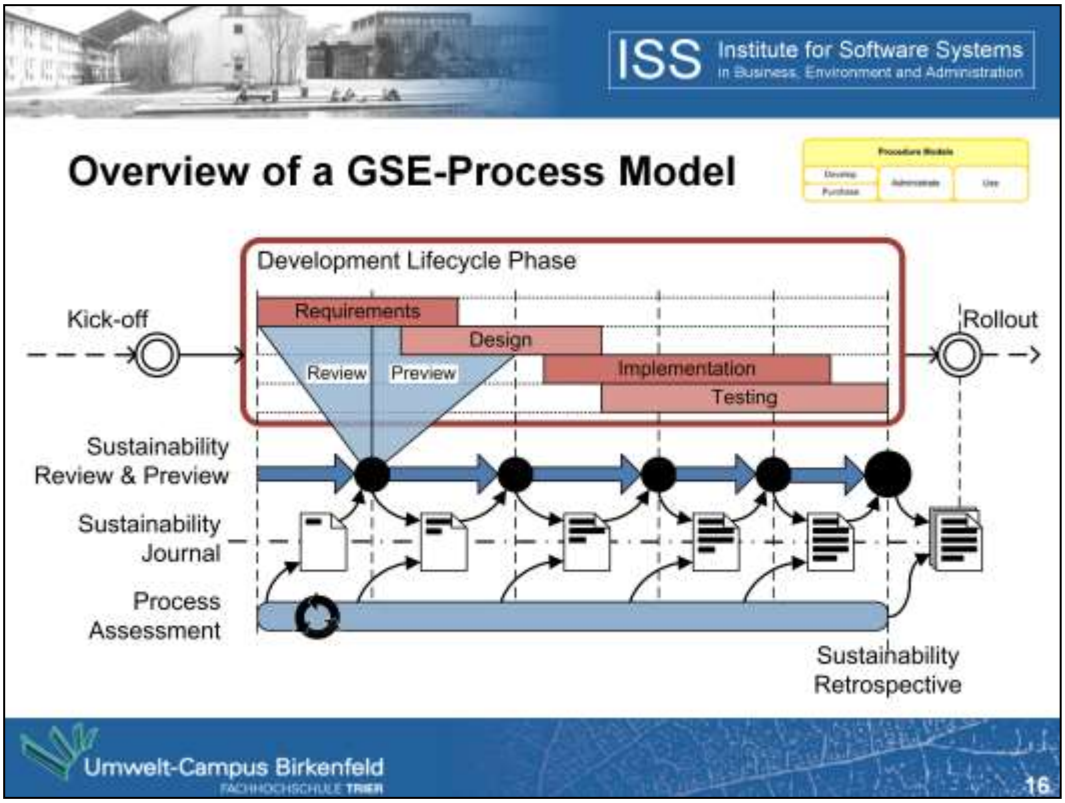
Reviews & Previews are a team facilitation approach. Involved actors review and assess their artefacts (e.g. requirements, architecture, coding) and develop and assess alternative solutions in order to choose the better “more sustainable” solution.

- The continuous Process Assessment activity quantifies and assesses impacts on sustainability, which result from the software development process itself.
- At the end of the development process, the Sustainability Retrospective combines the results of both activities:
    - Reviews & Previews (mainly impacts that are expected from the usage phase) and
    - Process Assessments (mainly impacts that result from the development phase).

  Thus it covers impacts over the whole lifecycle of the software product. The outcomes should be reported to stakeholders of the software product.

Additionally, it looks for ways to improve upcoming software development projects

The expected outcomes of this learning approach can be e.g. decisions for future projects, lessons learned, best practices regarding sustainability issues of software products or development processes.

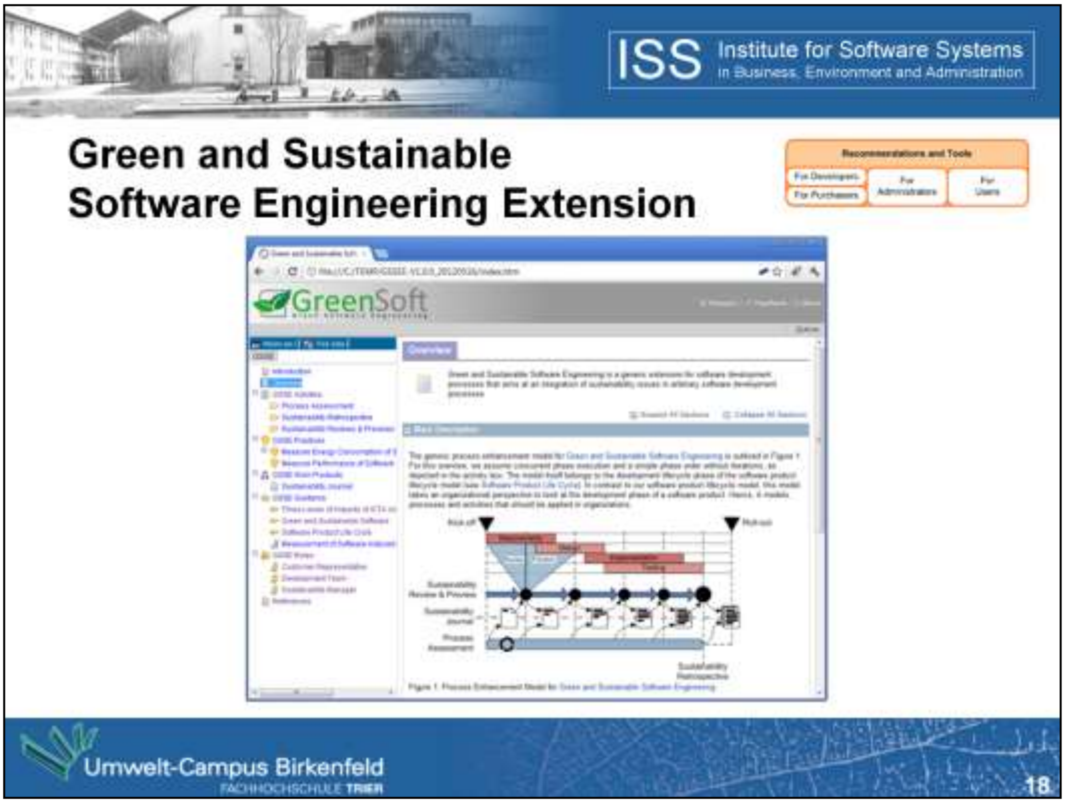


The Sustainability Journal is the information hub of our process enhancements.
It’s a well structured report, which evolves simultaneously with the software project.
Its purpose is to document Sustainability Reviews & Previews, Process Assessment and the Sustainability Retrospective.
Finally, after the project has finished, it reports the assessed impacts on sustainability.

But… how can we use all this? How can we bring the ideas and approaches to practical?

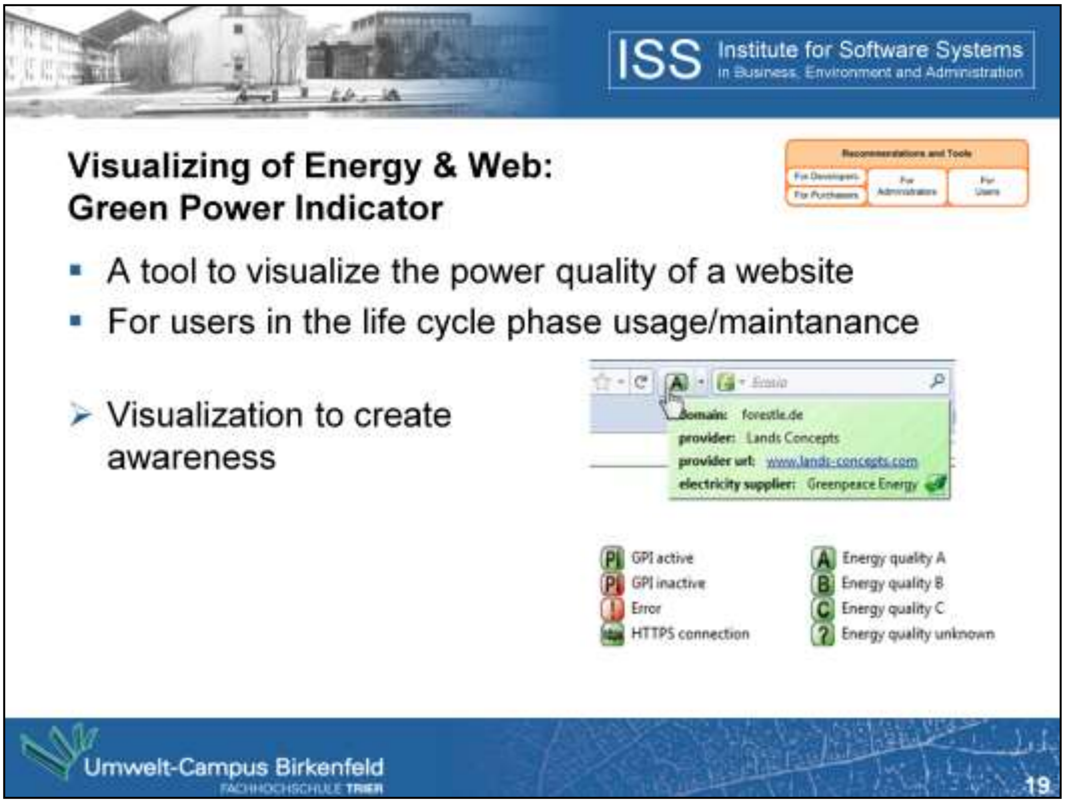
Our first step was to create tools and collect recommendations.

According to the process model, I introduced to you, we used the Eclipse Process Framework Composer and created the Green and Sustainable Software Engineering Extension



It is a summary of the procedure model and should support developers. Next to the description of the software engineering process it offers recommendations as well as templates, for example for the Sustainability Journal.

It can be published as a web application as you can see by this screenshot (is not published so far)



Since this tool is mainly for developers, the next one is for users in the usage phase. In order to create awareness for the energy consumption of the web (not only on the client, but also on the server side!) it is important to get more transparency in this context.

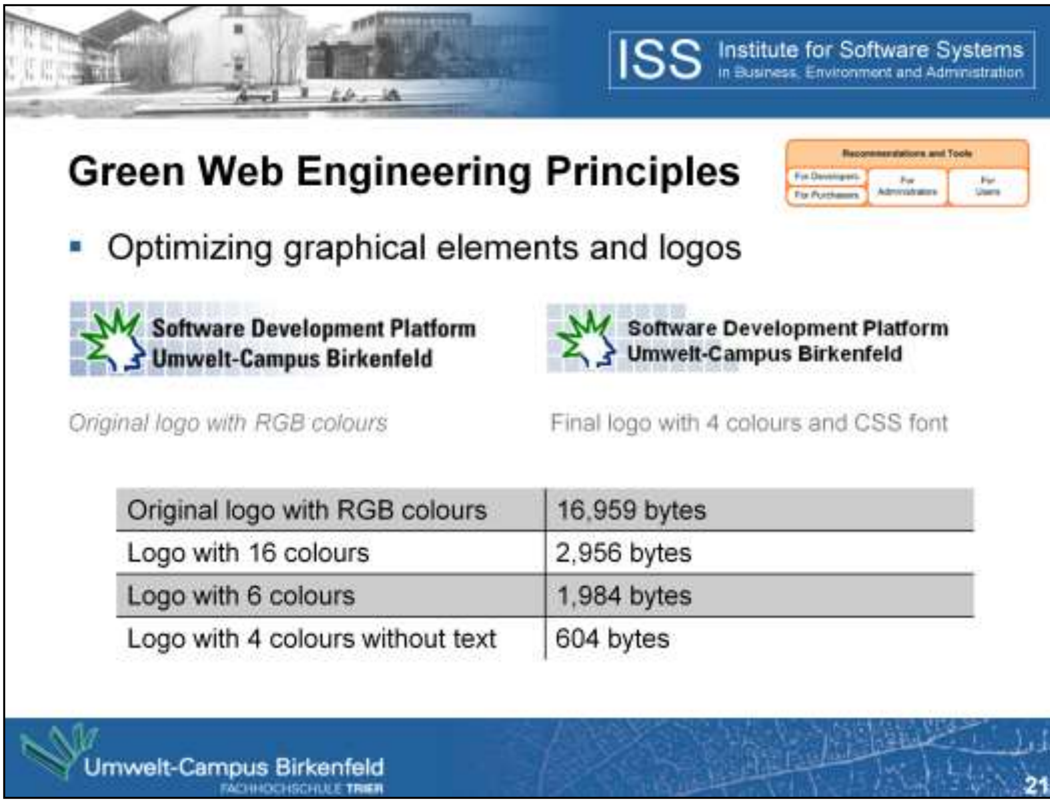
In order to create awareness it is important to visualize to users what is happening.

Our Firefox addon “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.

We think, if a user is aware of the energy quality of the provider they might demand green energy and brings the electricity and internet provider and administrator to act.
Main: awareness!

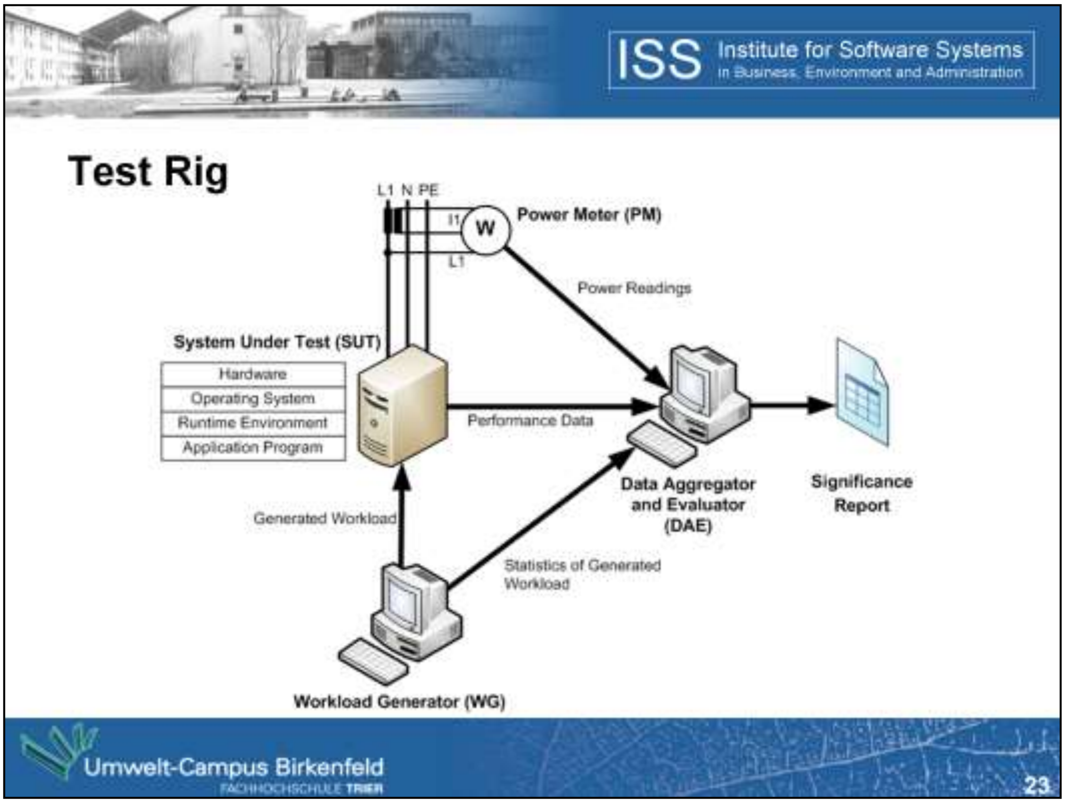
Regarding the same context, we sum up some principles for a Green Web Engineering in order to support administrators, developers and all the other stakeholders to move to a green(er) web.

For example, administrators should configure caching support and use compression.
Above that, developers should minimize and optimize JavaScript and CSS as well as graphical elements, logos and photos to reduce the data that has to be downloaded (sent by the server) to open a website.



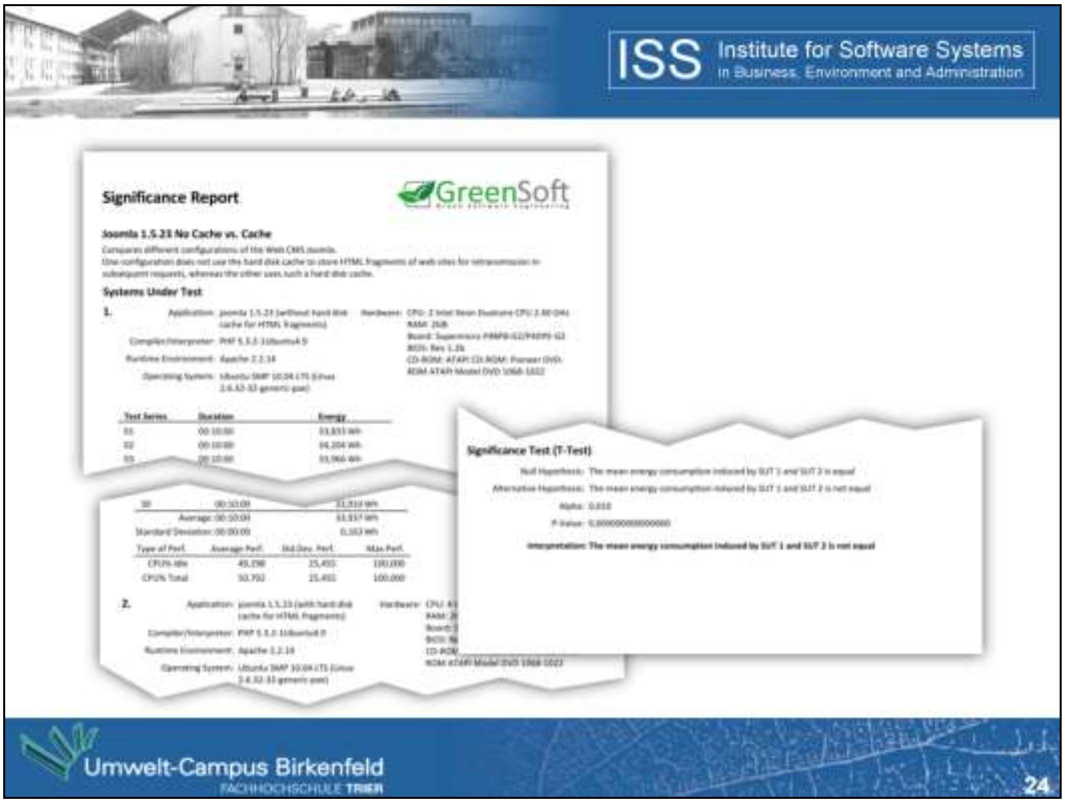
Here you can see an example in comparing the filesizes after optimization. As you can see there is just a small difference in this case of a small logo but you can imagine that the effect grows by applying the principles to all of the elements of a website.

III. Influences of Software on Energy Consumption



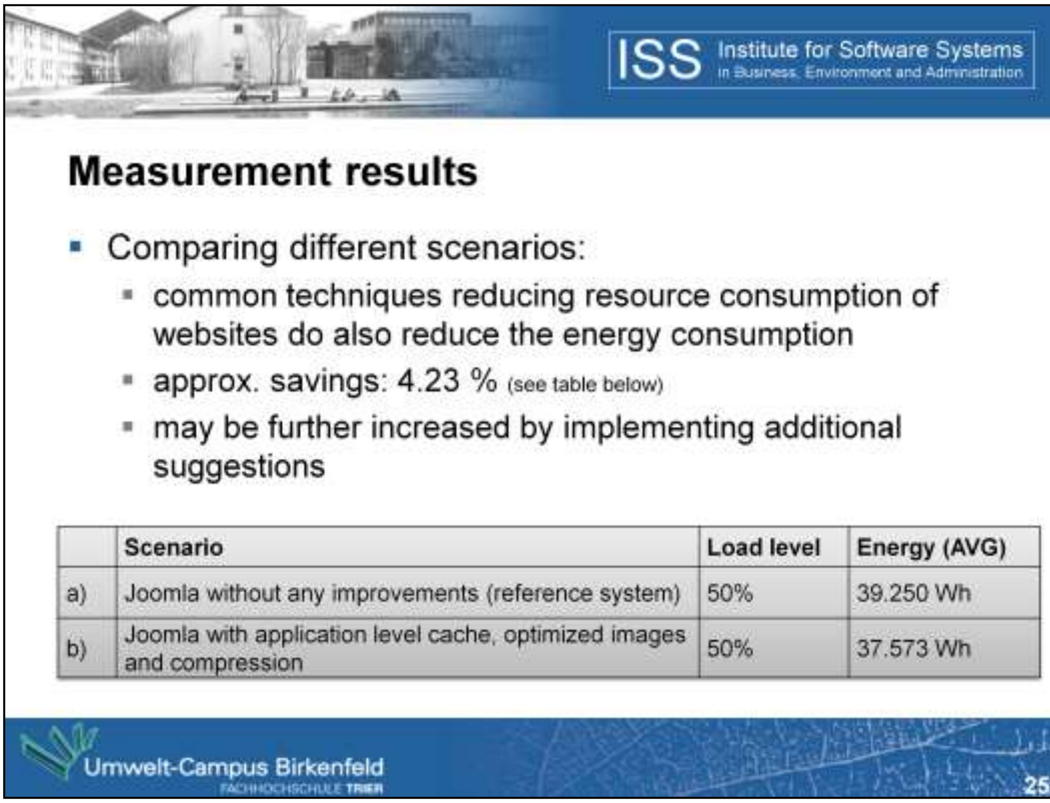
Basically, software has no energy consumption. Indeed, we measured the energy consumption of a specific combination of hardware components. Since software brings hardware to consume energy by execution software components.
This combination of the application program, the operating system and so on is called the System Under Test (SUT)

- The SUT is connected to a Power Meter (PM) which measured the consumed energy.
- The Workload Generator (WG) applies the reproducible workload to the SUT
    - Directly on the SUT in case of desctop software
    - Indirectly executed in case of server software
- The Data aggregator and evaluator (DAE) collects the different readings from the SUT (performance data), the PM (energy readings) and the WG (workload statistics) and generates the so called significance report.



This report states, which one of two compared systems consumes less energy and is therefore more energy efficient.

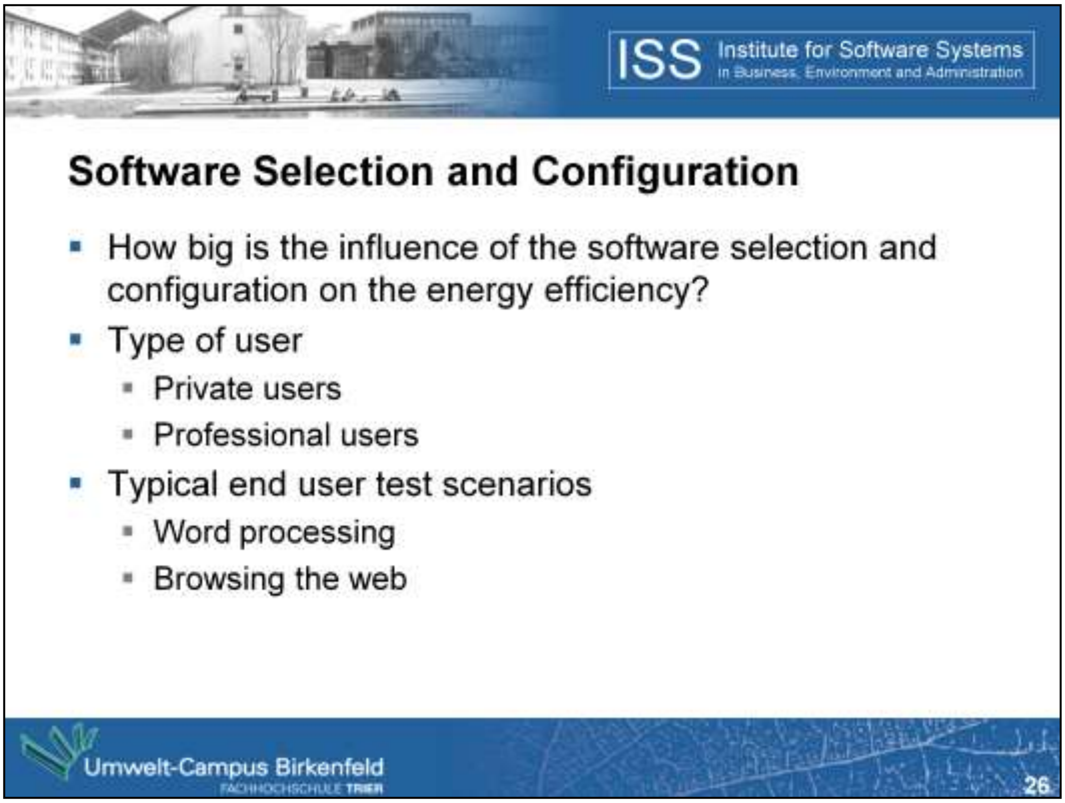
Here you can see an example of a “Significance Report” generated with our prototypical DAE software “S3C Power Analyzer“



Just to give you an example of our measurements:

We compared different scenarios of the Joomla Web Content Management System. The results are shown in the table. The difference was that Scenario a) did not have any improvements
And in case of b) we applied some suggestions of the green web engineering principles

The different energy consumption is not that big in this case (just 4,23%) but the important result is that there is a different!
And as I already said before, if one will do it, it might not be that effective, if everyone would reduce the data, it might be possible to switch of a server in a data centre and comsume less energy in that way.

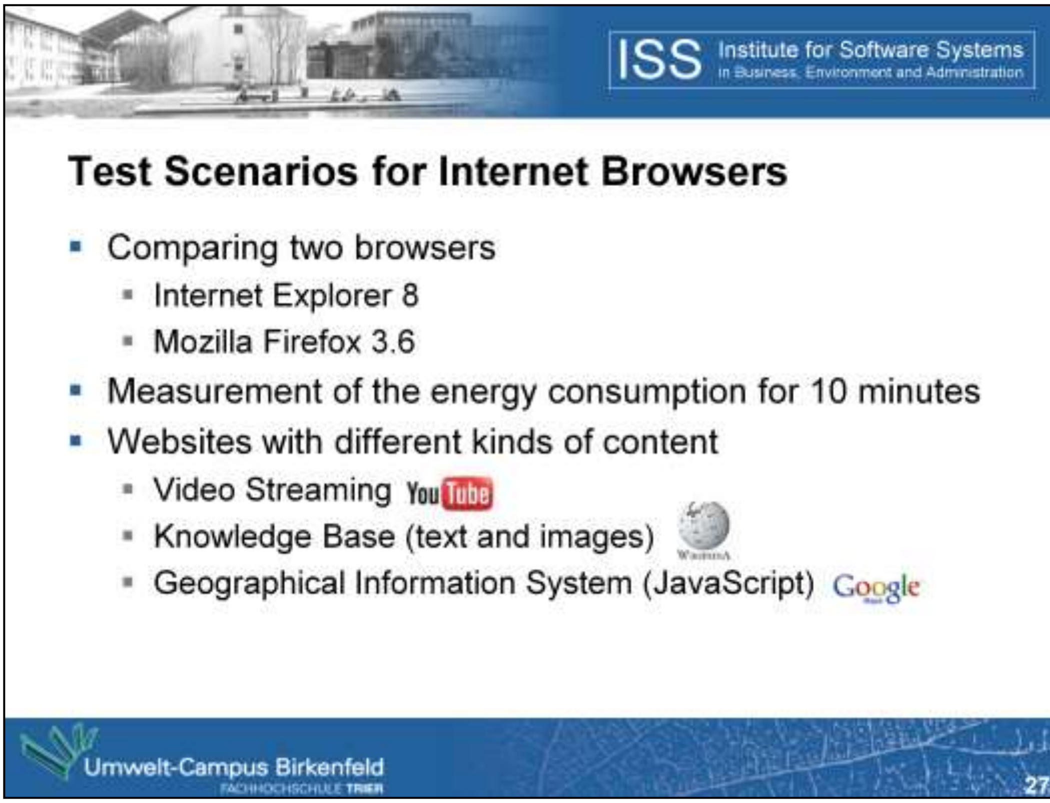


By testing the end user scenarios we wanted to find out how much users can directly reduce their energy consumed by IT.

In this context “end user” does not only include the private user at home but also business users, since the topic is also interesting for companies.
The agreement of Green IT influences the environmental report and carbon footprint of a company, which is getting more and more important in economics.

Statistics show that the private user (but also the business user) uses the PC mainly for word processing, browsing the web and communication.
Hence we set up test scenarios for two word processor application, namely Microsoft Office Word and Open Office Writer.
We defined actions which were performed while the measurements are being taken. These actions include typing text, formatting the document, adding an image, saving and printing the document.
The results show differences in the resulting power consumption of the two software tools.

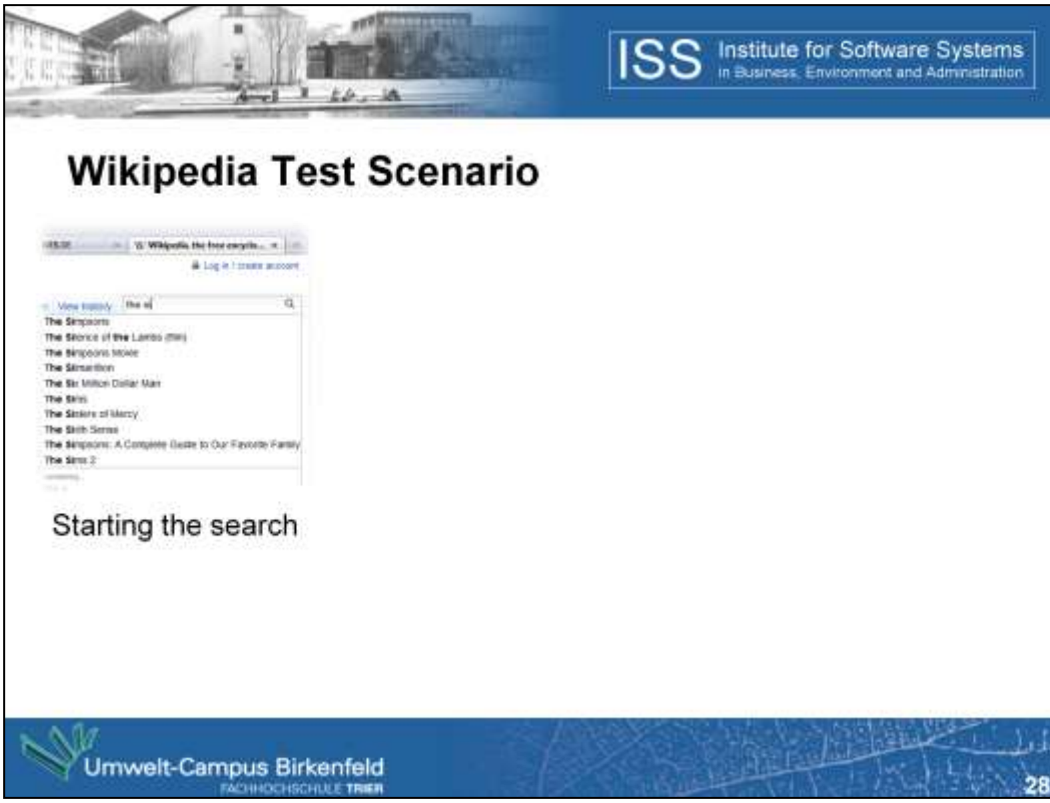
Furthermore we tested, two different internet browsers …



We compared the Internet Explorer 8 and Mozilla Firefox 3.6.

We set up test scenarios to take measures of the energy consumption of different end user’s applications.
All scenarios are laid out for a 10 minutes test run.
We used an automation tool to be able to repeat the tests as often as needed.

Since the energy consumption is related to the shown content of a website we chose websites with three different kinds of content:
- one video streaming website
- one knowledge base with simple text and images as content
- and one geographical information system which is realized with JavaScript



To test the energy consumption of the knowledge base website we set up the following test scenario:

During the measurements a search of a specific article is simulated.

As you know, a search of an article on Wikipedia is started by typing in the keyword.

ISS Institute for Software Systems in Business, Environment and Administration

Wikipedia Test Scenario

„Reading“ for 2 minutes

Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

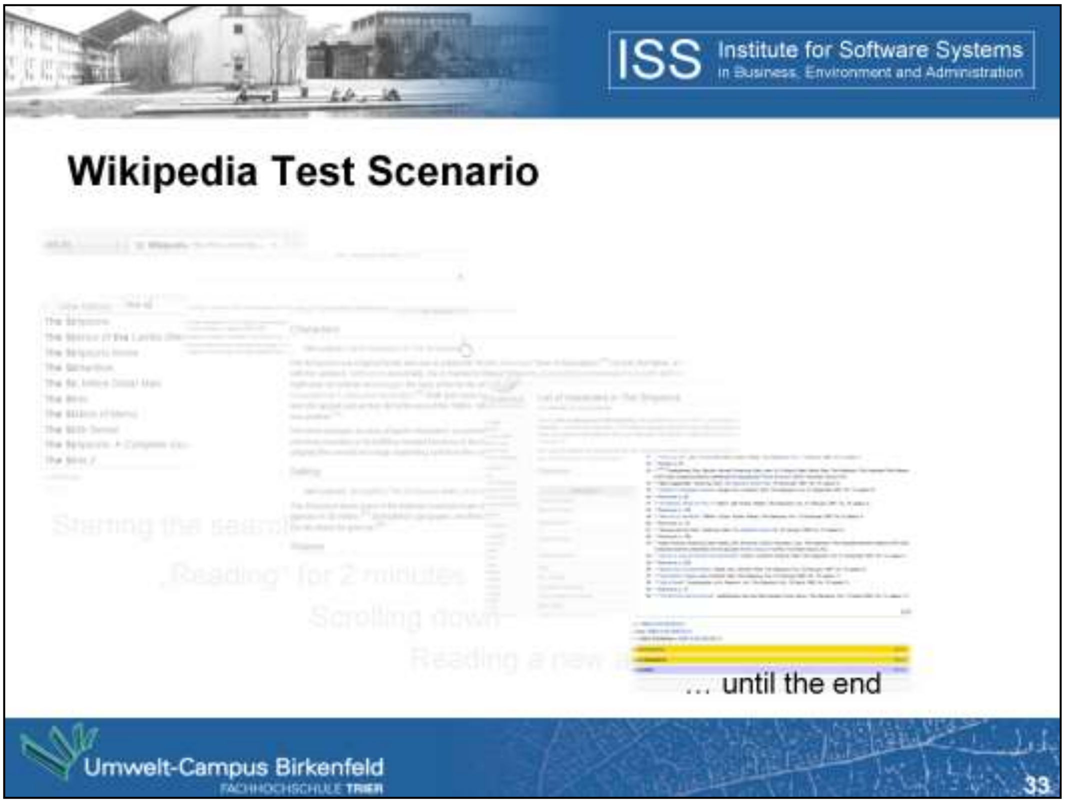
29

The next step was to simulate the reading operation….

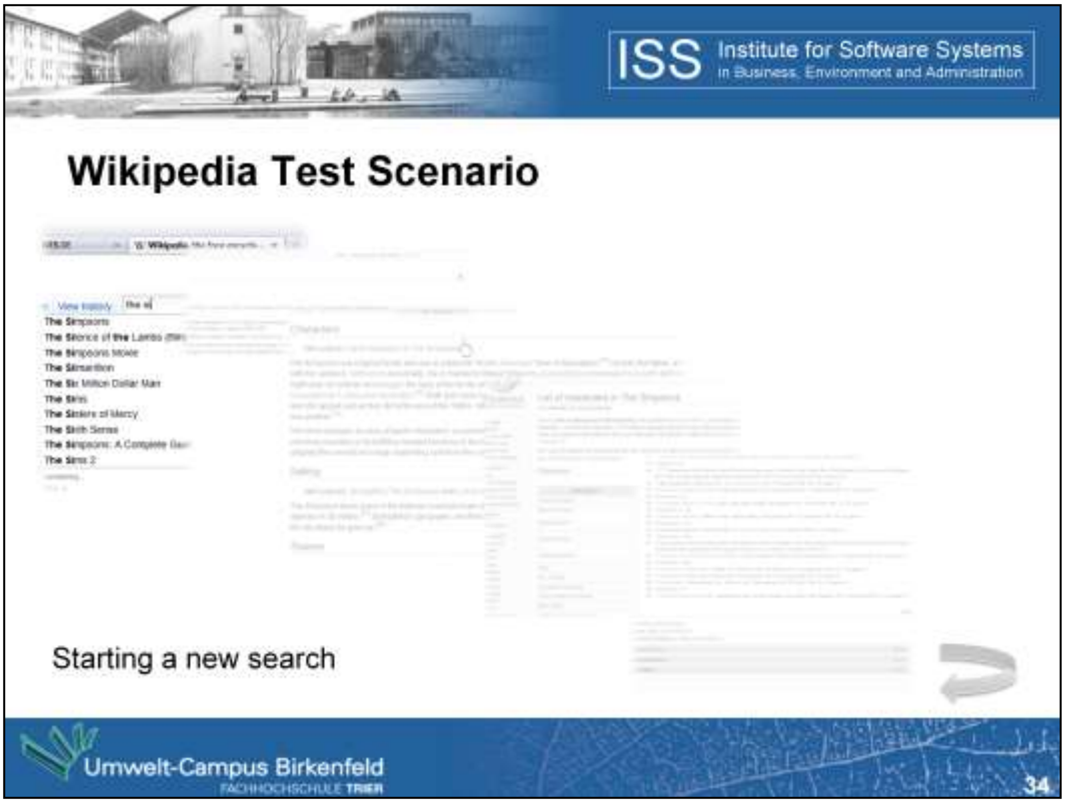
We did so by scrolling through the article and doing nothing for two minutes…

The user or rather our automation tool followed a link on the page…

And a new article is loaded

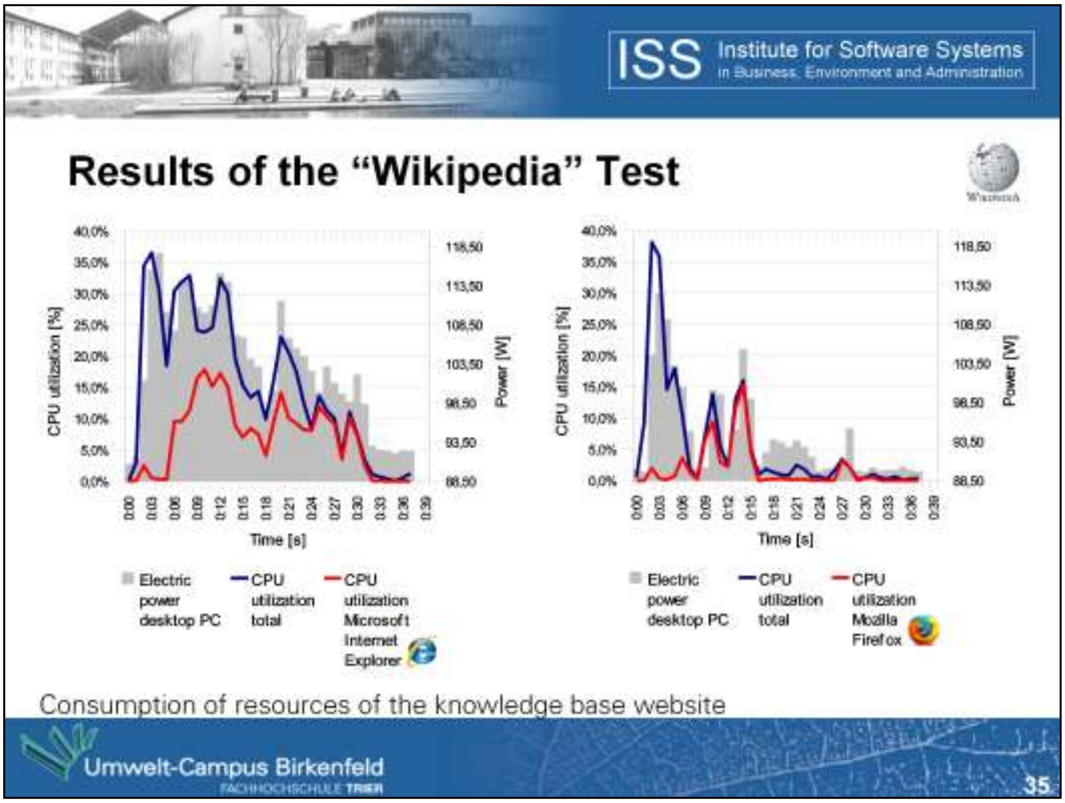


The user read this new article and scrolled until the end of it, means until the end of the shown website.



Afterwards a new search is

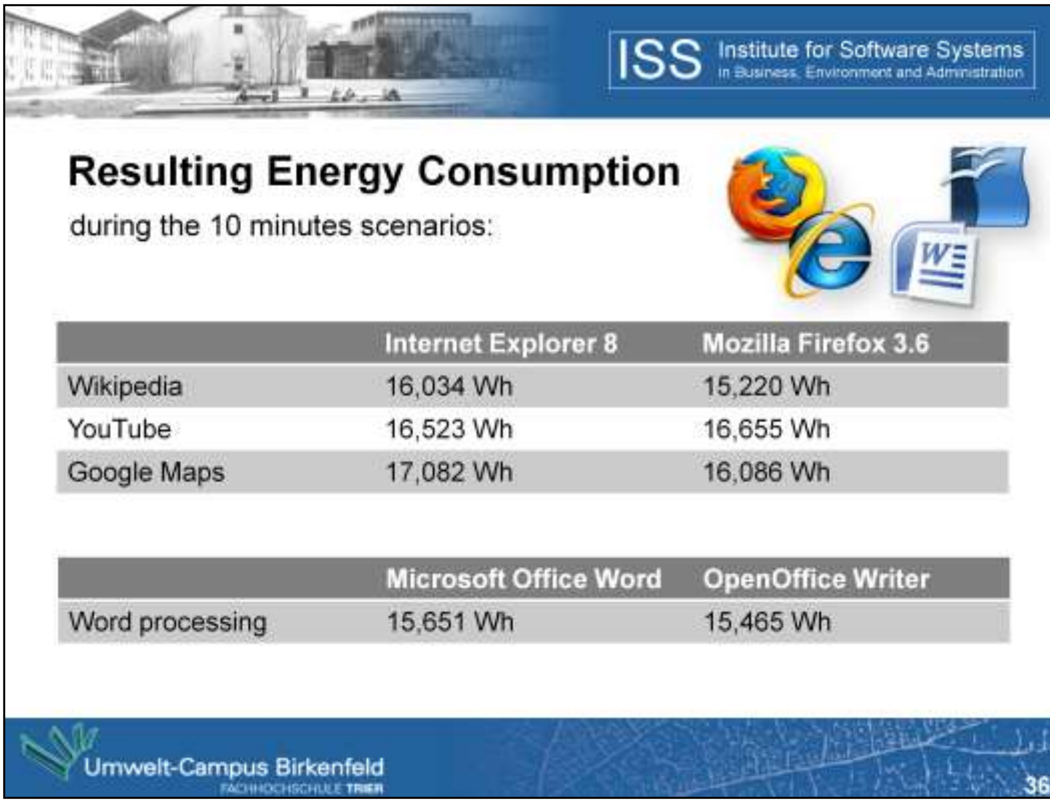
… and all the steps are repeated
So that the test runs for 10 minutes overall in the end.



Comparing the results of testing the two internet browsers different values of energy consumption can be detected.

Here you can see the results of testing the Internet Explorer (shown on the left)
And of the Firefox (shown on the right)

We chose the start of the search to show the resulting graph since this part is the most interesting part of the whole scenario. There you can see the biggest differences in the resulting values.
Even though the user is inactive the graphs show that the process is consuming energy.



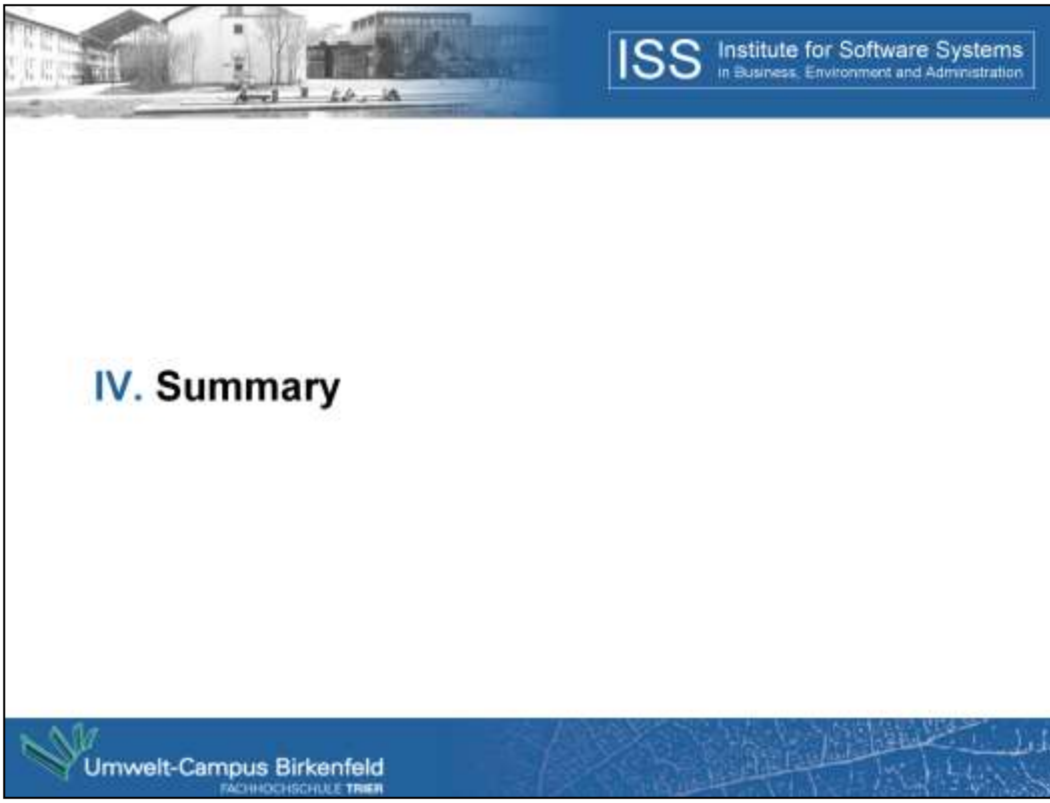
	Internet Explorer 8	Mozilla Firefox 3.6
Wikipedia	16,034 Wh	15,220 Wh
YouTube	16,523 Wh	16,655 Wh
Google Maps	17,082 Wh	16,086 Wh

	Microsoft Office Word	OpenOffice Writer
Word processing	15,651 Wh	15,465 Wh

Overall, means over the testing period of 10 minutes the Internet Explorer consumed about 16 Watt hours.

In contrast the Firefox needed only 15 and a half Watt hours while the actions were the same.

As you can see different values for the resulting energy consumption comes into notice for the internet browser as well as for the word processing applications.
Overall it is shown that there is a relation between the selected software tool and the resulting electrical power consumption.



So, let me just some up our research results / the presentation…

Summary

- The GREENSOFT Model
  - sums up different aspects of Green and Sustainable Software Engineering
  - is a theoretical approach
- Measuring software as a first step to get an impression of the influence of software to energy consumption
- Creating awareness is important!

➢ Theoretical model has to be evaluated in practice

If you have further ideas or suggestions on how to improve the measurement and rating method,

feel free to contact us at the Environmental Campus Birkenfeld of the Trier University of Applied Sciences in Germany.

Thank you very much for your time and attention.