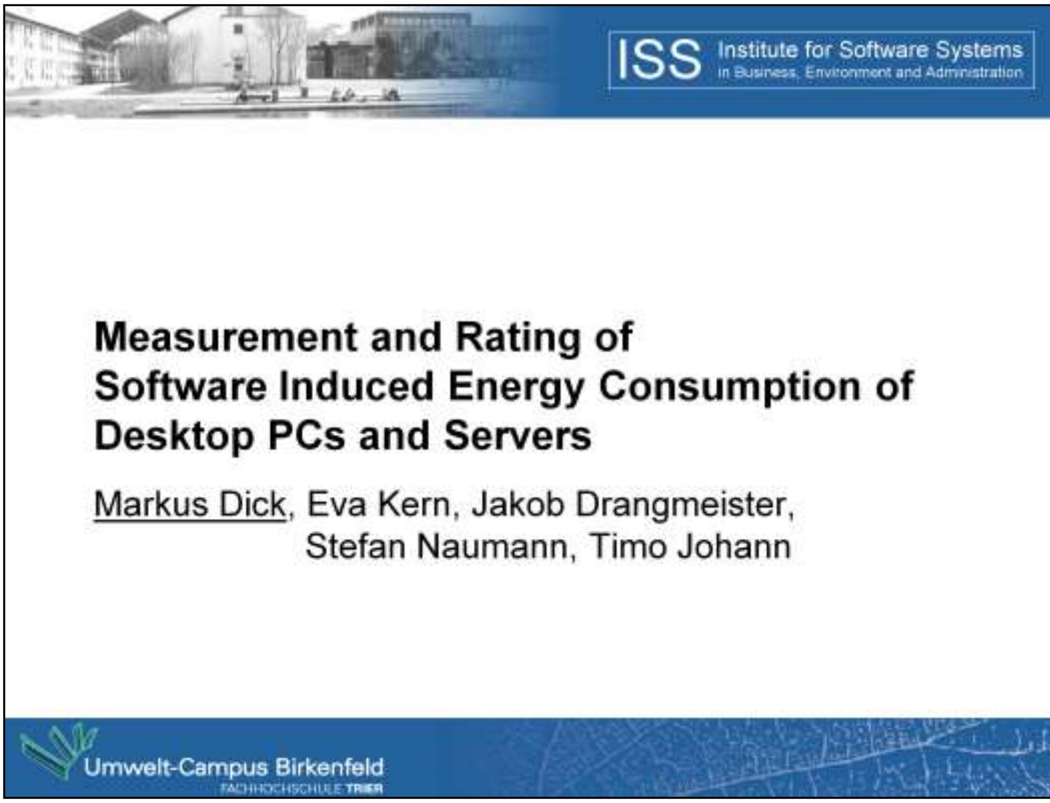


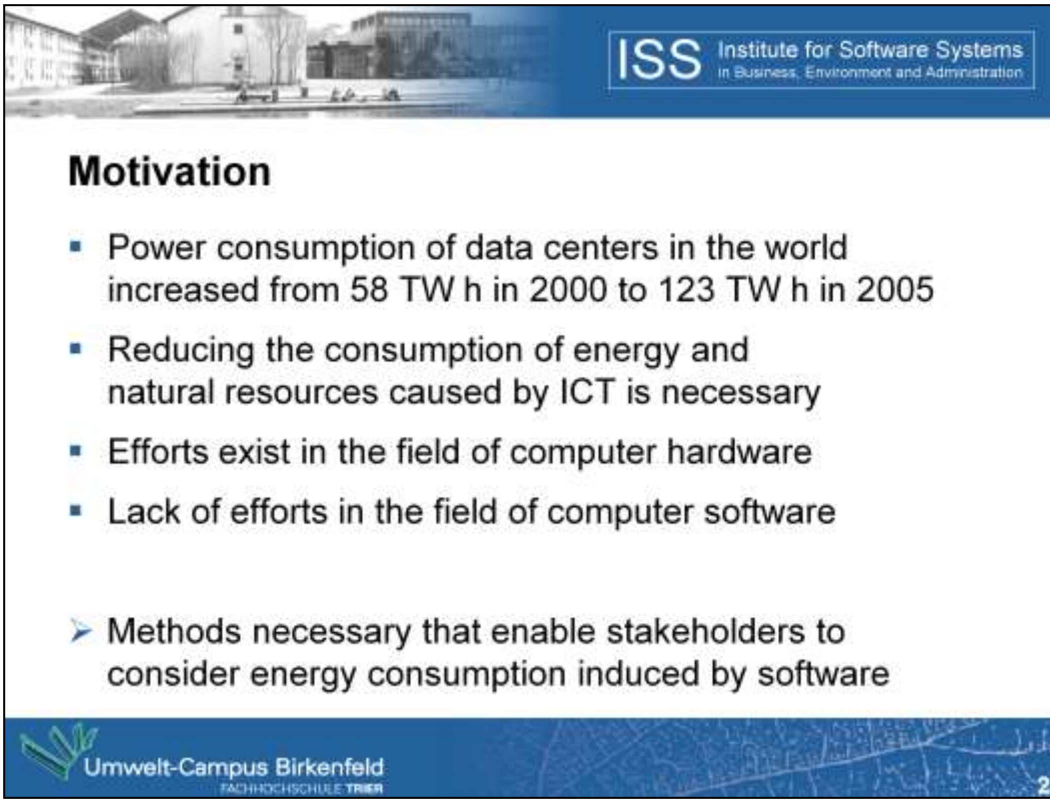
Markus Dick, Eva Kern, Jakob Drangmeister, Stefan Naumann, Timo Johann
{m.dick, e.kern, -/-, s.naumann, t.johann}(at)umwelt-campus.de

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; Kern, Eva; Drangmeister, Jakob; Naumann, Stefan; Johann, Timo: **Measurement and Rating of Software Induced Energy Consumption of Desktop PCs and Servers**. In: Pillmann, Werner; Schade, Sven; Smits, Paul (eds.): Innovations in Sharing Environmental Observation and Information. Proceedings of the 25th EnviroInfo Conference "Environmental Informatics" October 5-7, 2011, Ispra, Italy. Part 1, 2011, pp. 290 - 299.

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 to 123 TW h in 2005[1], 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 efforts in the field of computer software.
Therefore, methods are necessary that enable different stakeholders like developers, purchasers, administrators or even users to consider energy consumption induced by software in their decisions on software products.

[1] Koomey, J.G., 2007. Estimating total Power Consumption by Servers in the U.S. and the World. Final report, February 15, 2007. [Online] Analytics Press: Oakland.
Available: https://files.me.com/jgkoomey/98ygy0 [Accessed: 13 Oct. 2011].

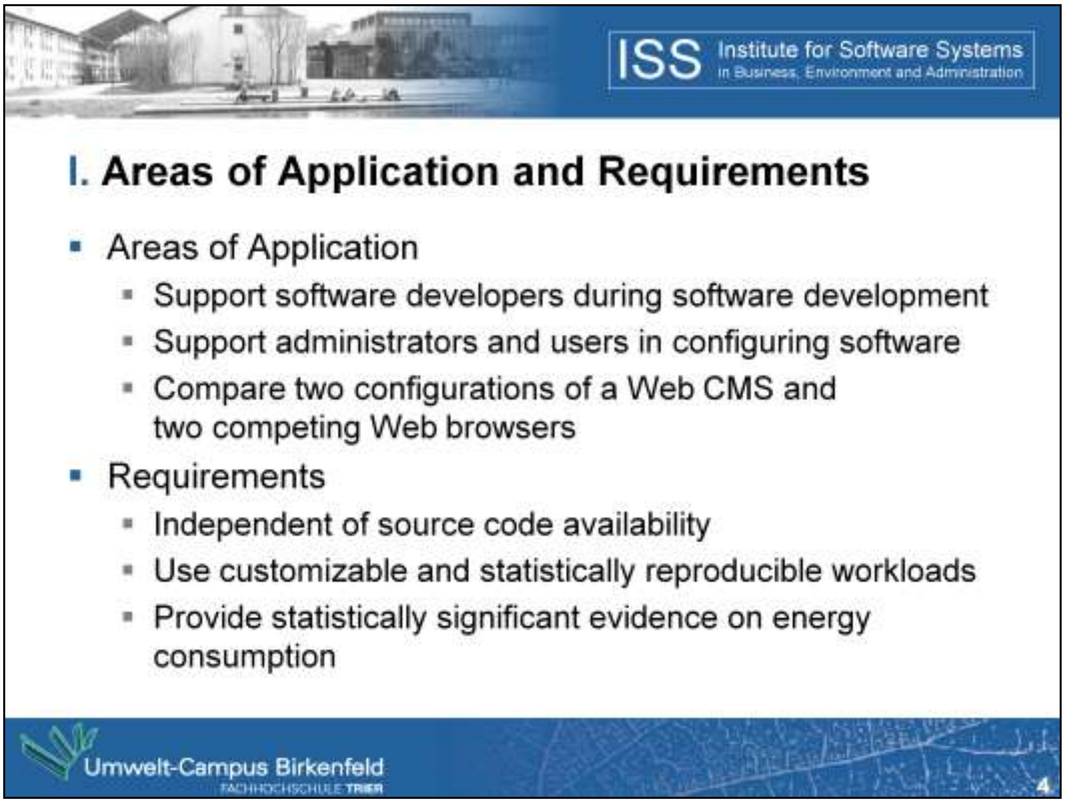
Outline

I. Areas of Application and Requirements

II. Test Rig and Measurement Method

III. Example Measurements

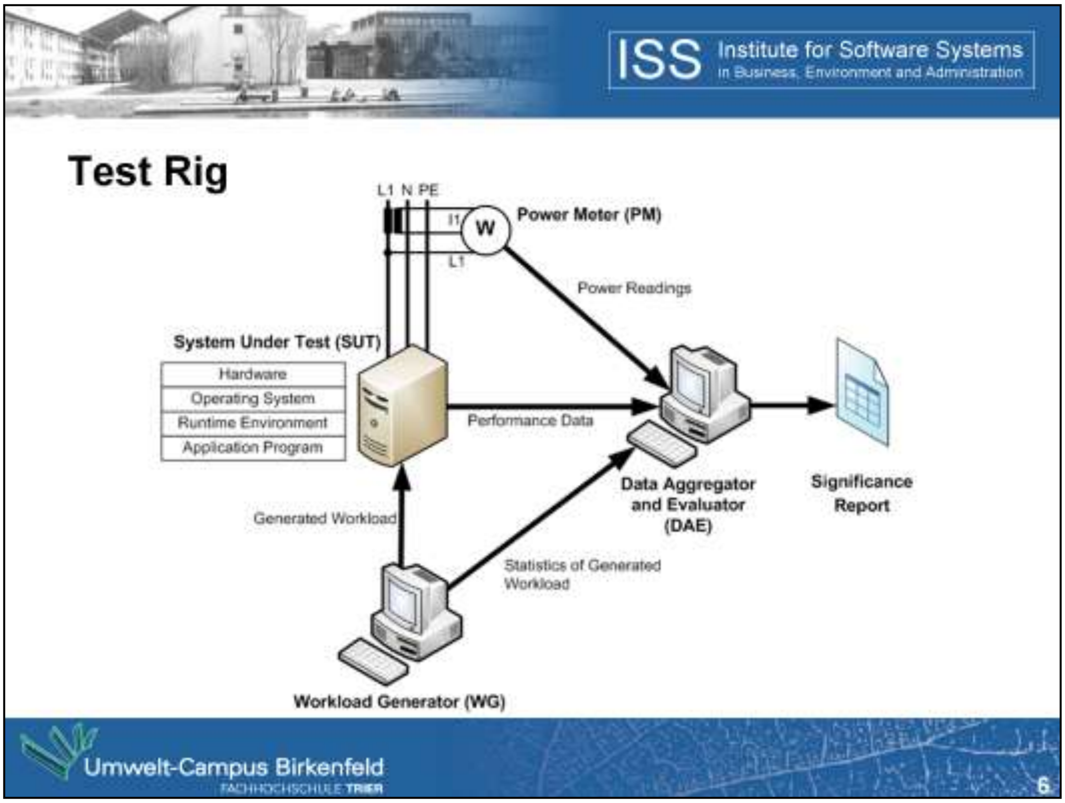
IV. Summary & Outlook



For our measurement method, there are several areas of application: Basically, it is intended to support software developers during software development but also administrators and users when configuring software or when deciding on software that they currently use or operate or plan to use or operate in the future.
We applied the method to compare the mean energy consumption of
•two configurations of a Web Content Management System (Web CMS) and
•two competing web browsers
These two measurements are later on shown as examples (as a kind of proof of concept) how the measurement method is applied to desktop PCs and servers.

The basic requirements are:
•It should be independent of source code availability, because administrators and users usually do not have the source code in order to inject special measurement code
•It should use customizable workloads so that it can be principally applied to any kind of software
•It should use statistically reproducible workloads so that workloads of different measurement experiments (the samples) are comparable
•Finally, it should provide statistically significant evidence on mean energy consumption of two compared software products

II. Test Rig and Measurement Method



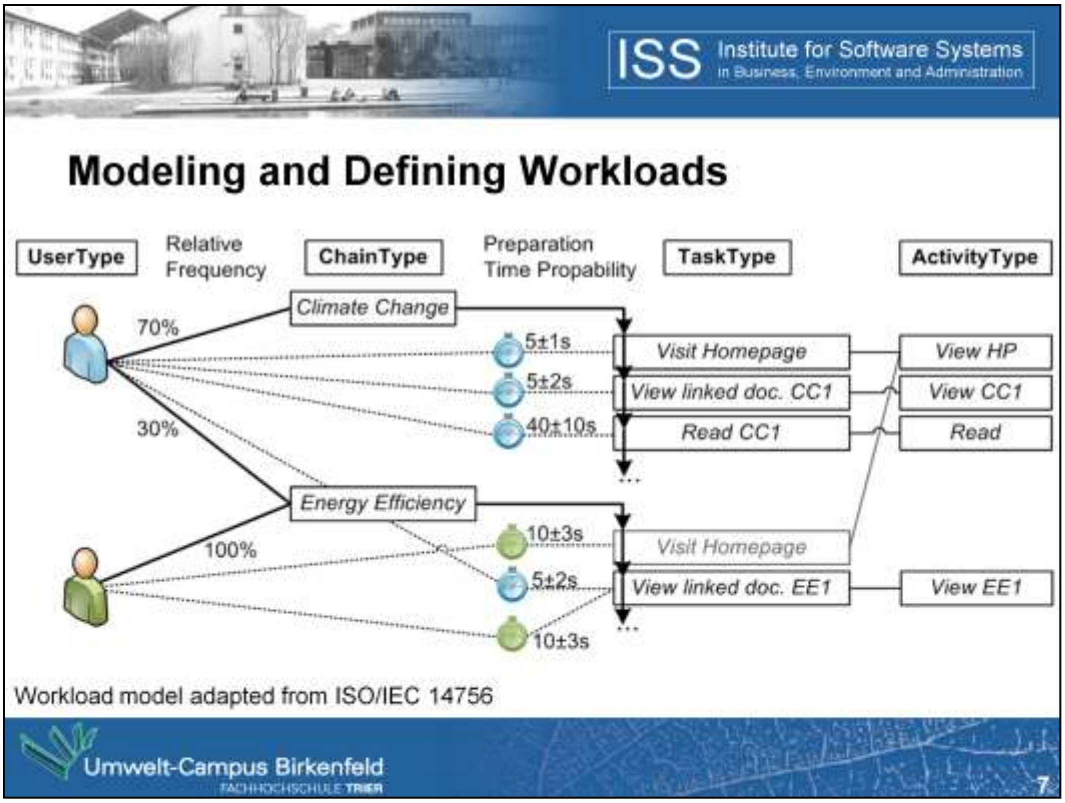
Basically, software has no energy consumption. Instead, we are measuring the energy consumption of a specific combination of hardware components that execute software components (e.g. operating system, runtime environment, application program). This is the so called “System Under Test” (abbr. SUT).

This SUT is connected to a power or energy meter (abbr. PM), which measures the consumed energy.

The Workload Generator (abbr. WG) applies the statistically reproducible workload to the SUT. It can be either directly executed on the SUT (e.g. in the case of measuring desktop software), but it can be also executed on a separate computer (e.g. in the case of measuring server software).

The so called “Data Aggregator and Evaluator” (abbr. DAE) collects the different readings from the SUT (CPU performance data), PM (power/energy readings), and the WG (workload statistics).

After aggregating the data, it generates the so called “Significant Report”. This report states, which of two compared systems consumes less energy and is therefore for more energy efficient.

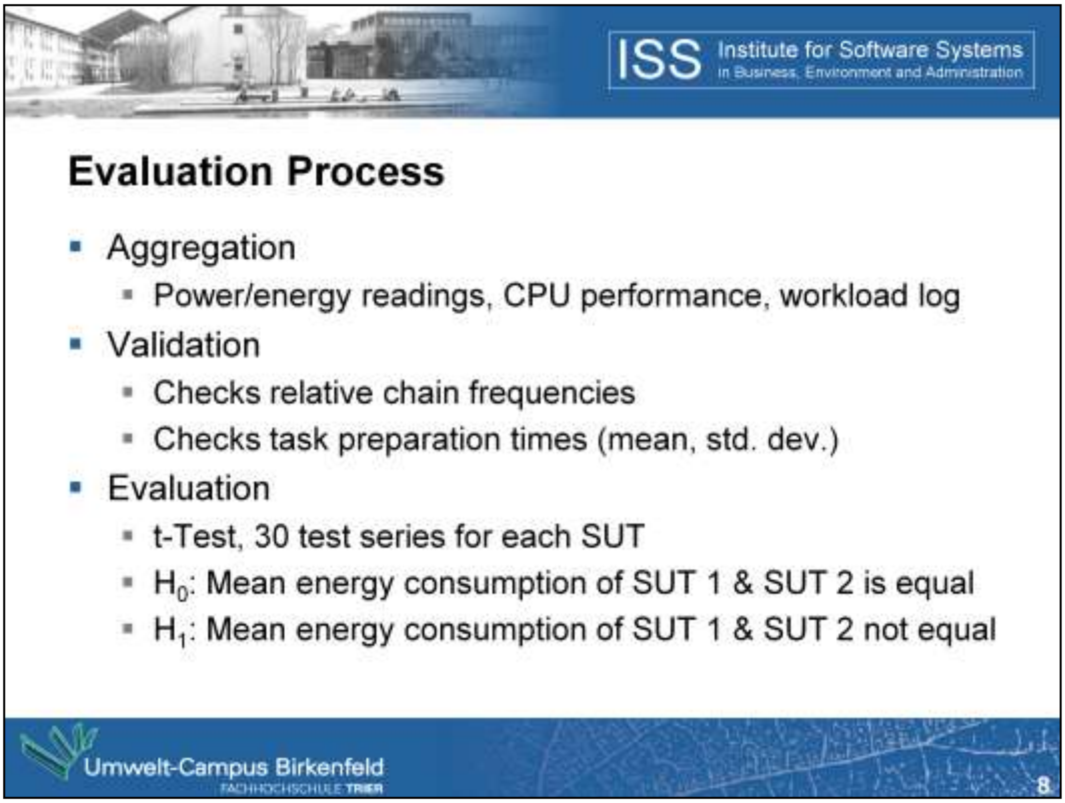


We did not invent the workload model by our own. Instead, we adapted the workload model from ISO 14756, which describes a measurement and rating method for computer systems performance.

The basic idea of the model is that users execute several task chains (one could also call them workflows), which consist of several tasks, which themselves are defined by a specific activity performed by the user and the preparation time (one could also call it “think-time”).

Due to the fact that we need to emulate users of different kinds, the workload model defines user types. For each user type, one can define different task preparation time propabilities. These preparation times are defined by mean and standard deviation. Each user type can also execute several task chains. For each user type, the relative frequency of task chain types is defined.

A complete workload definition also includes the number of users and their type, which should be emulated by the WG.



The evaluation process is performed in three steps:
1. Aggregation: DAE collects necessary readings from SUT, PM, WG
2. Validation: Answers the question if generated workloads comply with parameters predefined in the workload definition

This means: Checking that the relative chain frequencies are for each user type within acceptable tolerance

Checking that the task preparation times (mean, standard deviation) are for each user type within acceptable tolerance

The acceptable tolerance values need to be defined for each workload set.

3. Evaluation: If the validation has not failed, the mean energy consumption of two SUTs is evaluated with a statistical significance test.

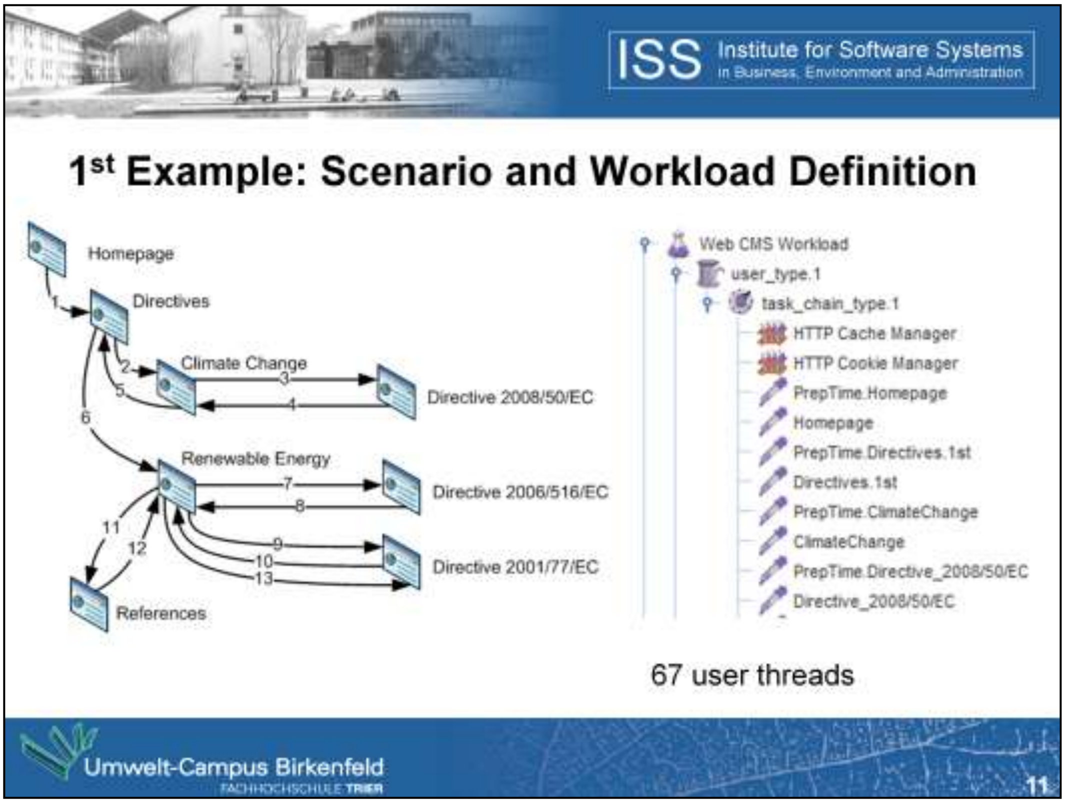
For this purpose we apply a standard t-Test for unpaired samples. Due to the fact, that we did not know in the beginning whether or not the samples will be normal, we applied 30 measurement experiments to get 30 samples of mean energy consumption for each SUT. According to the central limit theorem, we can assume that the samples are approximately normal distributed.

Of course, conducting 30 measurement experiments is not practical for daily use, e.g. in continuous integration scenarios of agile software development projects, because this takes a long time. Hence, for daily use, one may use less measurements.

III. Example Measurements

1st Example: Software on Servers

- Web CMS Joomla! (v.1.5)
  - Installation with vs. without disk cache for HTML fragments
  - No web browser cache support for generated web pages
  - Web browser cache for static content (via HTTPd)
- Workload Generator
  - Apache JMeter load and performance test tool
- Environment
  - 2 x Intel Xeon dual core CPU @ 2.4 GHz, 2 GB RAM
  - Ubuntu GNU/Linux 10.04 SMP (Kernel 2.6.32)
  - Apache Web Server 2.2, PHP 5.3, MySQL 5.1



The picture on the left hand side shows the structure of our example website. The numbers denote the order in which the user visits the different web pages.
The workload has only one user type and only one task chain.
In the task chain, four web pages are accessed several times: the Directives page 2 times, the Climate Change page 2 times, the Renewable Energy page 4 times, and one of the legal documents two times.

The picture on the right hand side shows corresponding Apache JMeter test plan.
The workload starts 67 threads, which represent a user single user.
This number was determined by experiment: with more threads the validation failed due to loss of accuracy in preparation times.

ISS Institute for Software Systems in Business, Environment and Administration

1st Example: Results

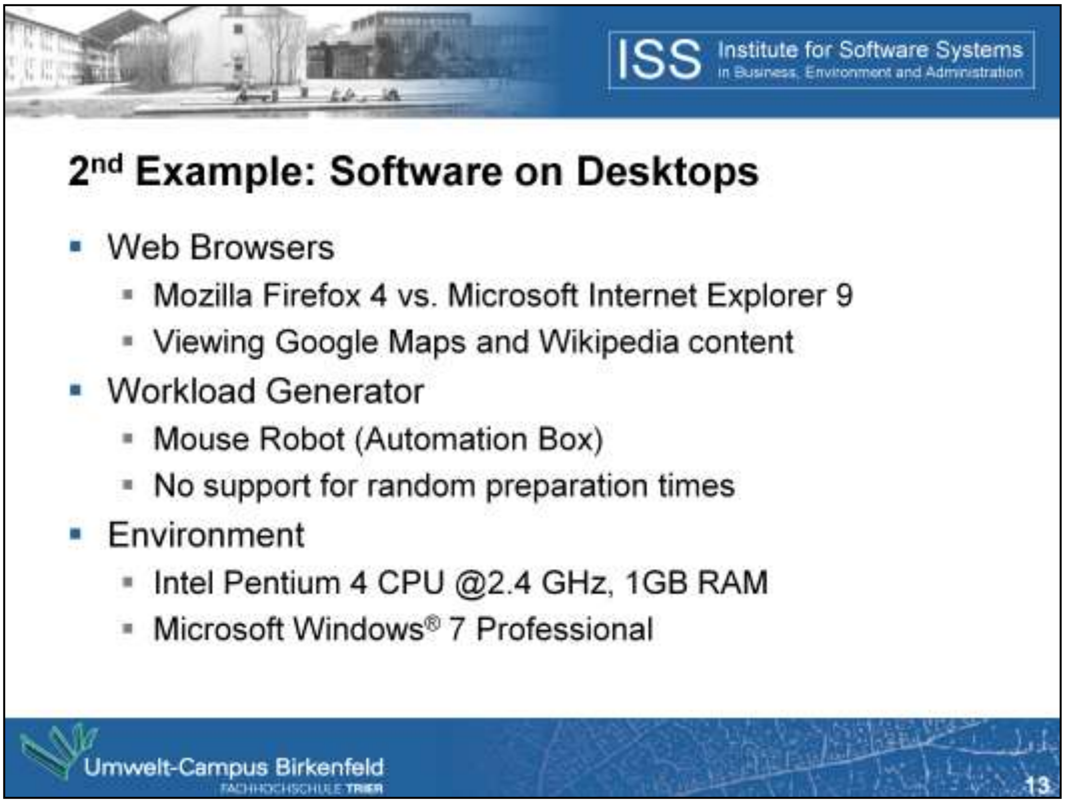
N = 30	Energy		Performance	
	Mean	*Std. Dev.*	*Mean*	*Std. Dev.*
Without Cache	33.94 Wh	0.163 Wh	50.7%	25.5%
With Cache	31.01 Wh	0.096 Wh	31.8%	16.8%

- ✓ Difference in means statistically significant ($p<0.01$)
- Approx. 4,000 page requests in 10 min. (576,000/day)
- ➢ Disk cache results in power savings of approx. 8.6%

Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

12

Projection to one year of 24/7 operation:
savings 153,9 kWh/a = 30,78€/a (0.20€/kWh)



Mozilla Firefox 4.0.1
Microsoft Internet Explorer 9.0.8112.16421IC

MouseRobot is a desktop automation tool. Unfortunately, it has no support for random preparation times, so we decided to use constant preparation times.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
This is an independent publication)and is not affiliated with, nor has it been authorized, sponsored, or otherwise approved by Microsoft Corporation.

2nd Example: Results

- Wikipedia

N = 30	Energy Mean	Energy Std. Dev.
Firefox 4.0.1	9.38 Wh	0.111 Wh
Internet Explorer 9.0.8	10.77 Wh	0.145 Wh

✓ Difference statistically significant ($p<0.01$)

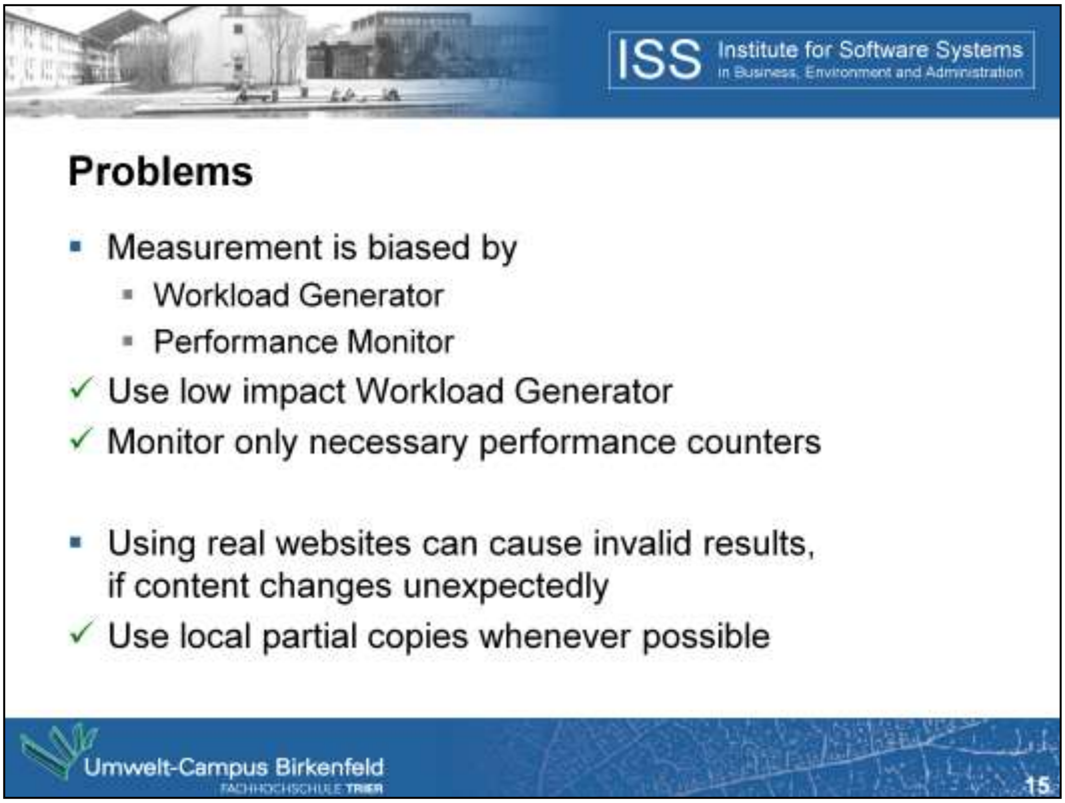
➢ Savings FF compared to IE approx. 12%

- Google Maps

N = 30	Energy Mean	Energy Std. Dev.
Firefox 4.0.1	11.87 Wh	0.244 Wh
Internet Explorer 9.0.8	14,79 Wh	1.434 Wh

✓ Difference statistically significant ($p<0.01$)

➢ Savings FF compared to IE approx. 19%

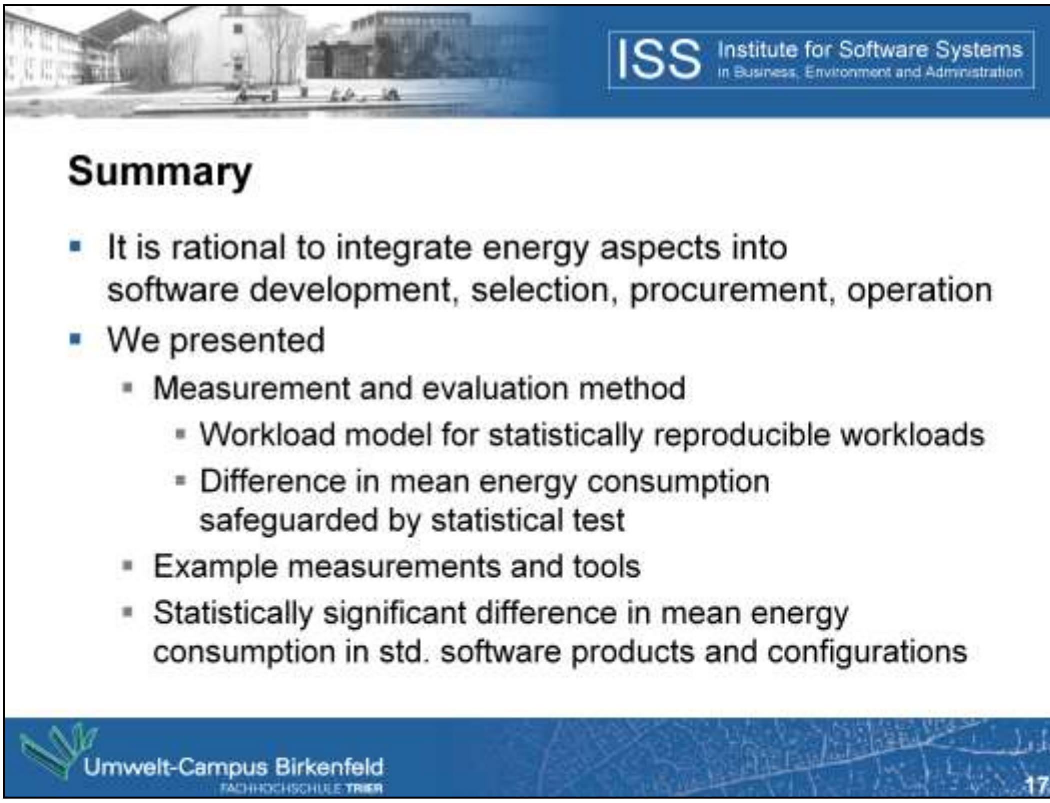


When conducting measurement experiments, there may occur several problems:

The measurement can be biased by
–the WG, if it is directly executed on the SUT, e.g. for desktop software
–the performance monitor that logs CPU performance readings
Hence, we propose to use a low impact WG and to monitor only performance counters that are necessary (e.g. CPU Total, WG, Application, Idle)

For the browser tests, we used real websites. This can lead to invalid measurement results, if the content on the websites changes unexpectedly (e.g. if advertising images are replaced by videos or new images)
Hence, we propose to use local partial copies or artificial websites whenever possible.

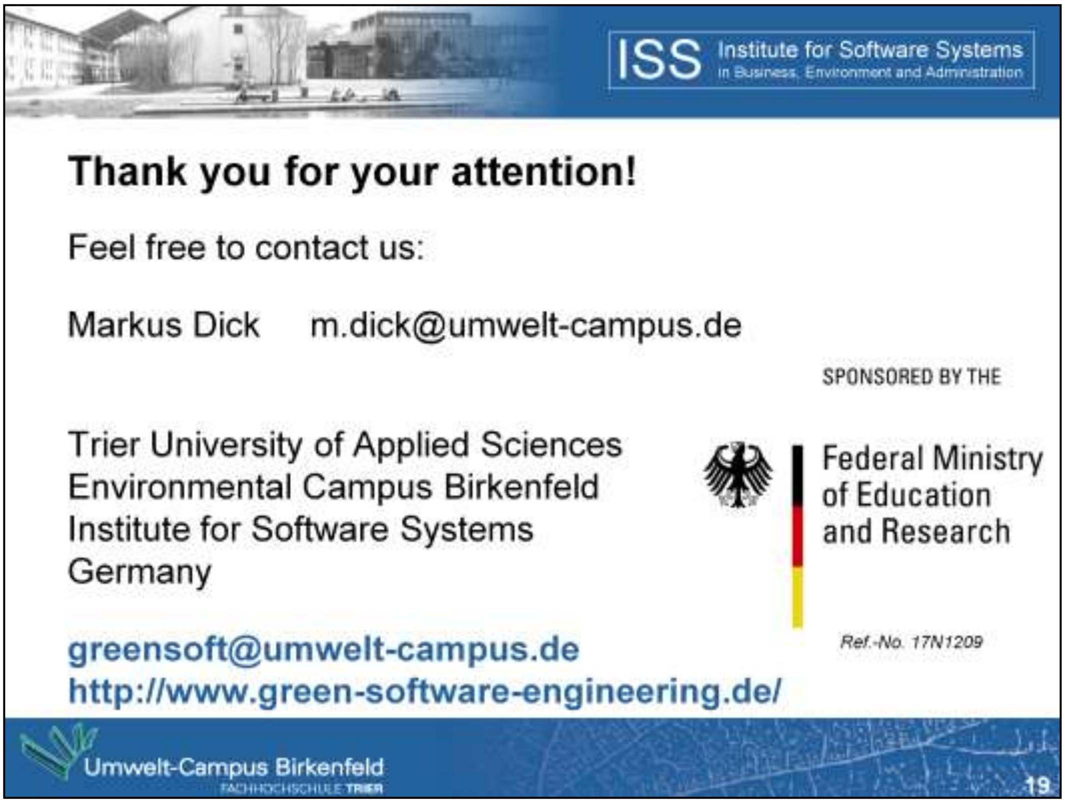
IV. Summary & Outlook



With our measurement method, we showed that there is a difference in mean energy consumption of different standard software products and even in slightly different configurations of software.

Outlook

- Measure different products, versions, configurations of standard software
- Compare software at different CPU performance levels
- Integrate white-box measurement to support developers
- Provide supporting software tools
- Enable measurement on virtualized servers
  by applying appropriate power models



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.

Statistics Web CMS “Joomla!”

	Gruppe	N	Mean	Std. Deviation	Std. Error Mean
Joomla10mins	cache	30	31,00867	,095835	,017497
	nocache	30	33,93680	,162914	,029744

		Levene's Test for Equality of Variances		t-test for Equality of Means				
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference
Joomla10mins	Equal variances assumed	9,771	,003	-84,852	58	1,570E-62	-2,928133	,034509
	Equal variances not assumed			-84,852	46,924	5,158E-53	-2,928133	,034509

Statistics output was generated with IBM SPSS Statistics 19

•Levene (< 0,01) → Equal variances not assumed
•t-Test (< 0,01) → H_0 rejected → Means are not equal

Statistics Web Browser

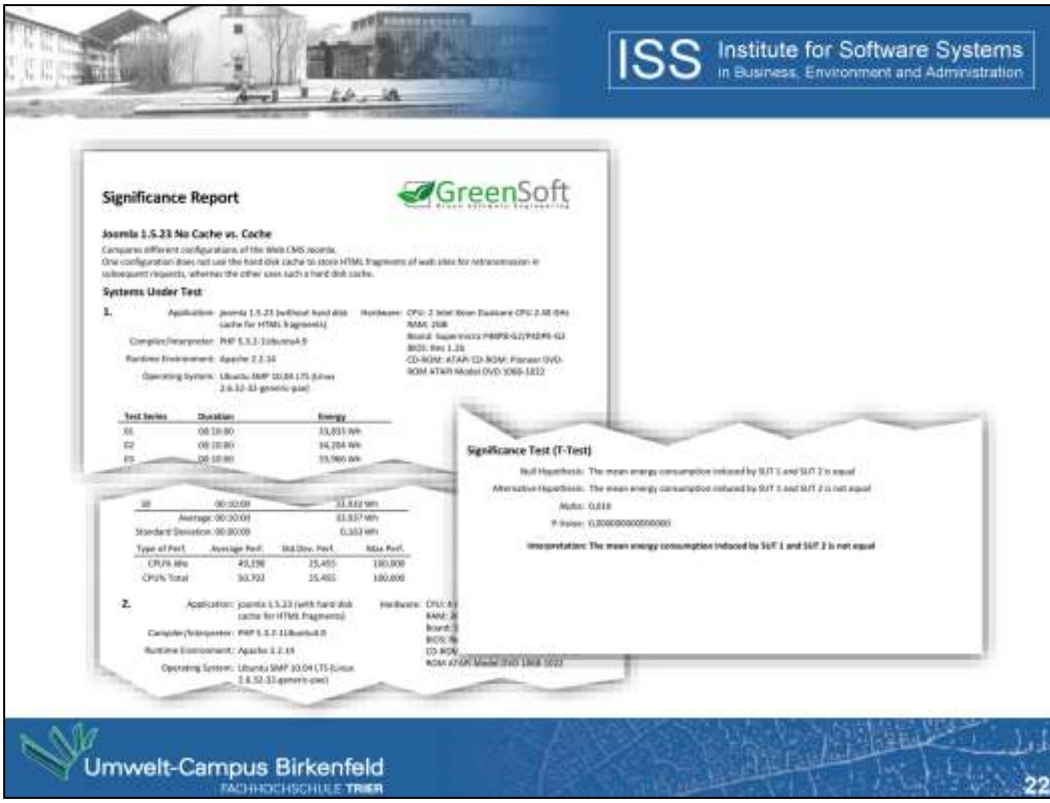
	Browser	N	Mean	Std. Deviation	Std. Error Mean
GoogleMaps	FF	30	11,8710	,24463	,04466
	IE	30	14,7880	1,43421	,26185
Wikipedia	FF	30	9,3820	,11124	,02031
	IE	30	10,7710	,14547	,02656

		Levene's Test for Equality of Variances		t-test for Equality of Means				
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference
GoogleMaps	Equal variances assumed	23,794	,000	-10,981	58	8,624E-16	-2,91700	,26563
	Equal variances not assumed			-10,981	30,686	3,721E-12	-2,91700	,26563
Wikipedia	Equal variances assumed	,735	,395	-41,544	58	7,500E-45	-1,38900	,03343
	Equal variances not assumed			-41,544	54,275	7,990E-43	-1,38900	,03343

Statistics output was generated with IBM SPSS Statistics 19

Web Browsers on Google Maps
•Levene (< 0,01) → Equal variances not assumed
•t-Test (< 0,01) → H_0 rejected → Means are not equal

Web Browsers on Wikipedia
•Levene (> 0,01) → Equal variances assumed
•t-Test (< 0,01) → H_0 rejected → Means are not equal



An example of a “Significance Report” generated with our prototypical DAE software “S3C Power Analyzer“