

Einfluss von Software auf den Ressourcenverbrauch

Ergebnisse des Forschungsprojektes „Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software“



Jens Gröger, Öko-Institut e.V.
Bits & Bäume 2018
Berlin 17. November 2018

Forschungsfragen

- Gibt es einen **Zusammenhang** zwischen Software und Ressourcenverbrauch?
- Können **Kriterien** zur Beurteilung der Ressourcenbeanspruchung durch Software definiert werden?
- Eignen sich solche Kriterien zum **Vergleich** verschiedener Softwareprodukte?
- Können **Mindestanforderungen** definiert werden, die eine Ressourcen schonende Software beschreiben?



Forschungsprojekt

„Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik“

- Auftraggeber: Umweltbundesamt
- Abschlussdatum: Juli 2018

Bearbeitung durch Forschungsverbund

- Öko-Institut e.V., Bereich Produkte und Stoffströme
 - Expertise in der Nachhaltigkeitsanalyse von IKT-Geräten, Entwicklung von Vergabekriterien für Umweltzeichen
- Universität Zürich, FG Informatik und Nachhaltigkeit
 - Expertise in Ökobilanzierung von IKT-Geräten und Infrastruktur
- Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme in Wirtschaft, Umwelt und Verwaltung
 - Expertise in Bewertung und von Software, Green Software und Green Software Engineering



**Universität
Zürich** ^{UZH}



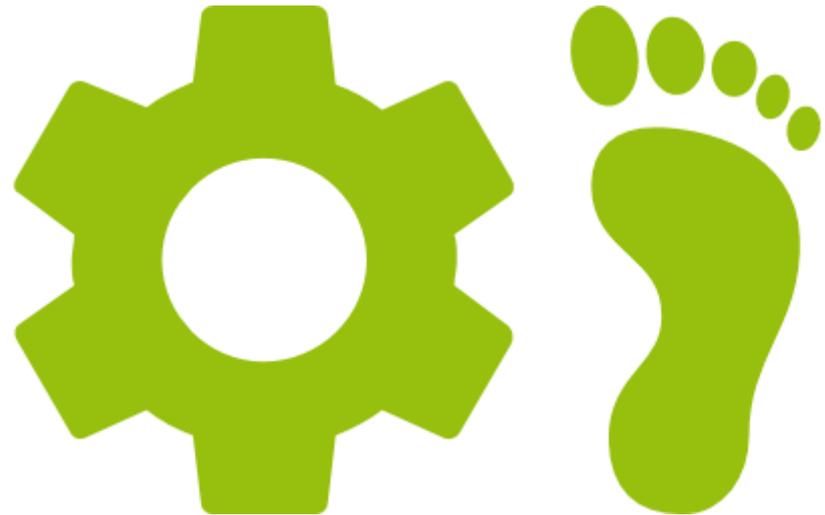
HOCHSCHULE TRIER
Umwelt-Campus Birkenfeld

Wichtigste Ergebnisse

- Software ist maßgeblich für **Energieverbrauch** und **Hardware-Inanspruchnahme** und kann zur Obsoleszenz der Hardware beitragen.
- Es wurde ein **Kriterienkatalog** für nachhaltige Software entwickelt, der sich zur Beurteilung und Optimierung von Software eignet.
- Die Anwendung des Kriterienkataloges zeigt **deutliche Unterschiede** zwischen verschiedenen Softwareprodukten.
- Es konnten Kriterien identifiziert werden, die sich für ein Umweltzeichen für Software eignen.

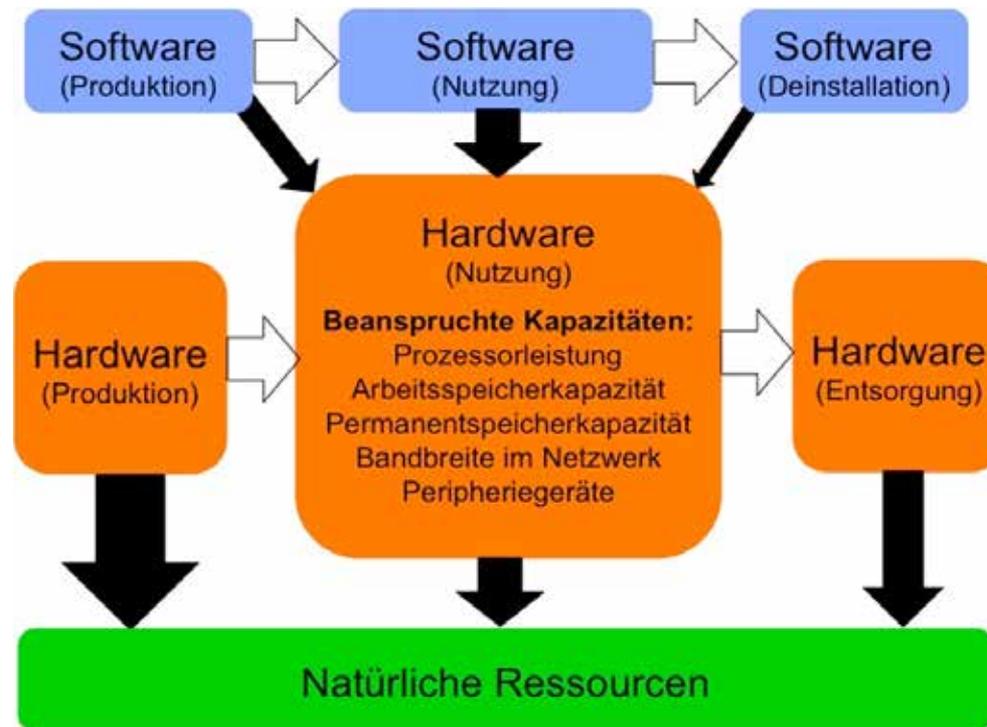


Zusammenhang zwischen Software und Ressourcenverbrauch



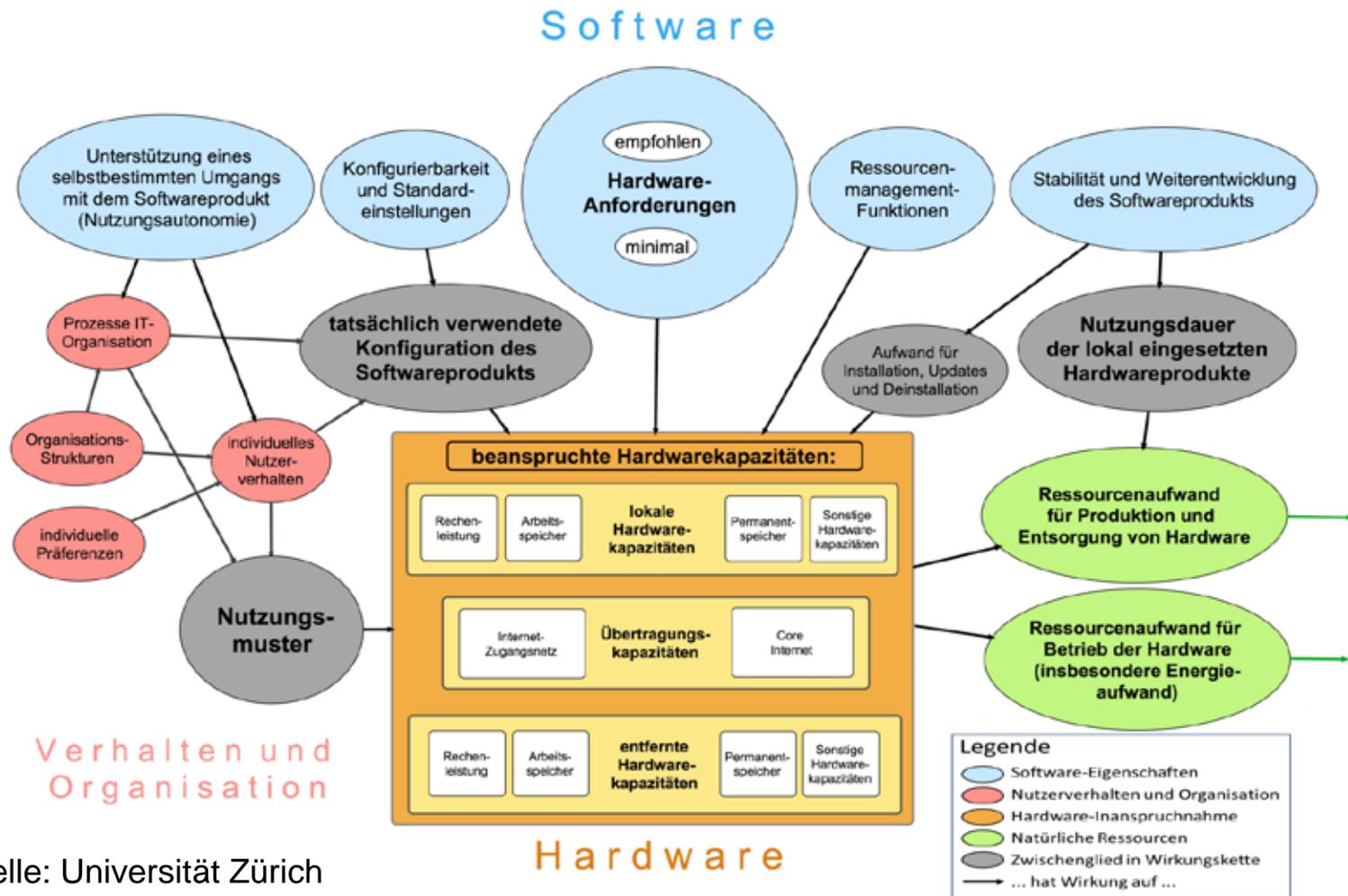
Wirkungsmodell von Software (einfach)

- Umweltwirkungen von Software entstehen entlang des gesamten Lebenszyklus – vermittelt über die Beanspruchung von Hardware



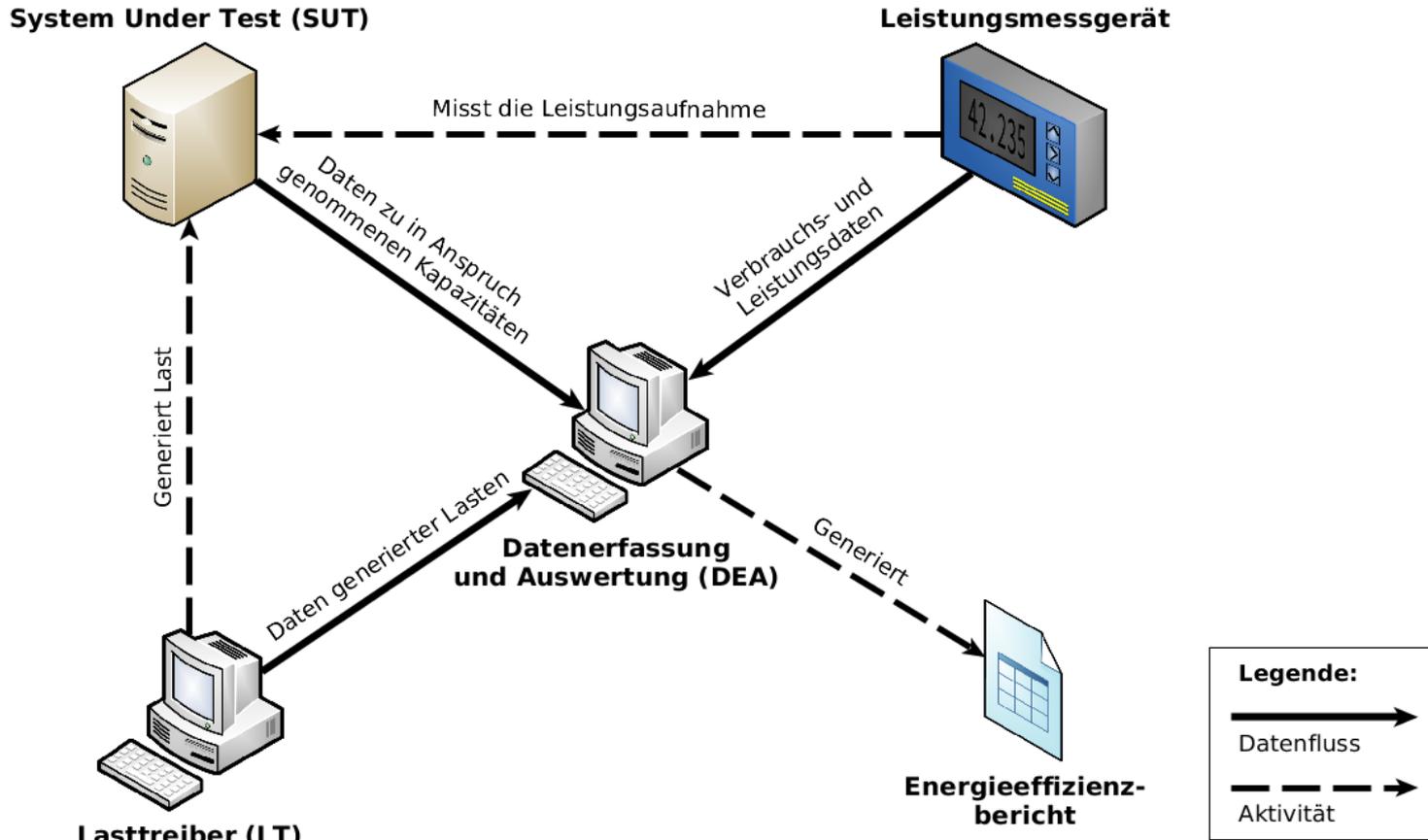
Quelle: Universität Zürich

Wirkungsmodell von Software (ausführlicher)



Quelle: Universität Zürich

Messaufbau zur Untersuchung von Software



Quelle: Hochschule Trier

Weitere Voraussetzungen

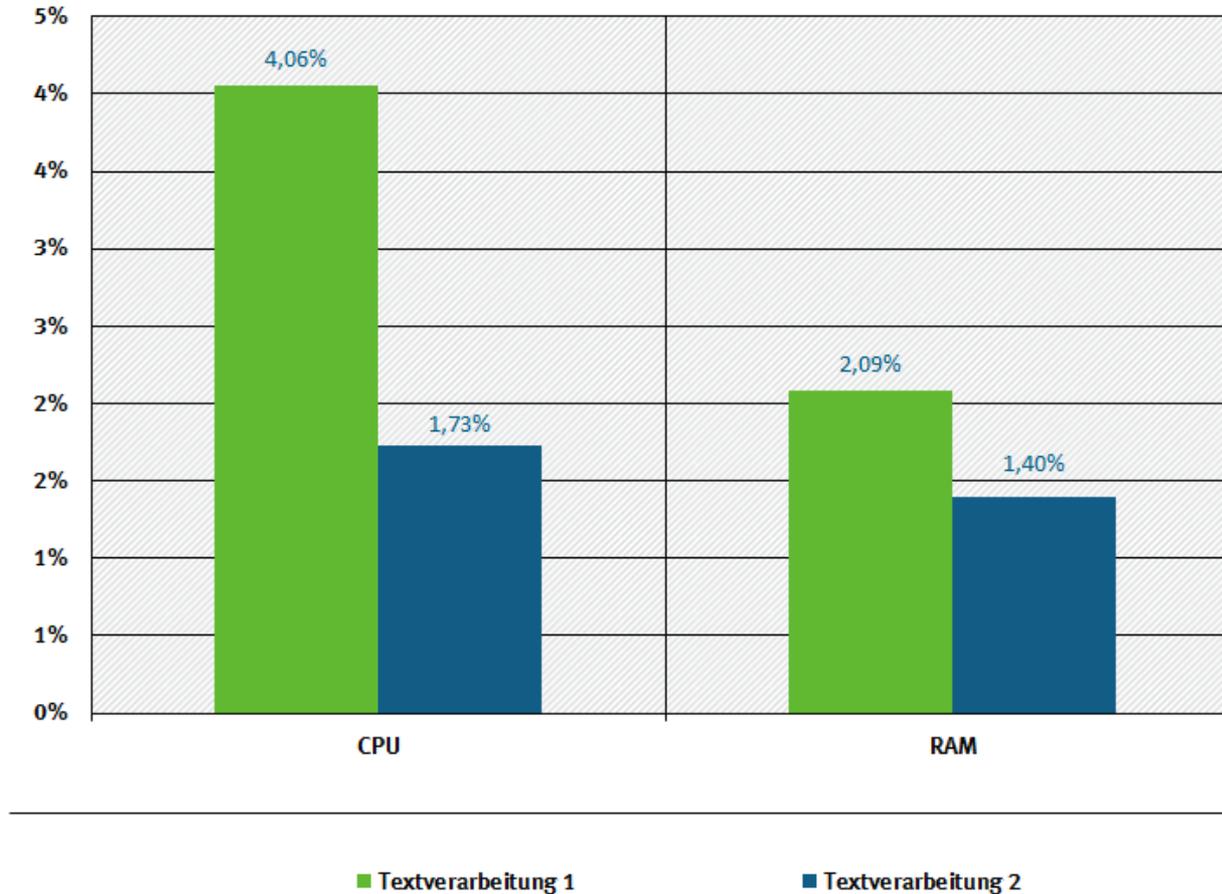
- Festlegung eines **Referenzsystems** zur Durchführung der Messungen (System Under Test)
 - Rücksetzbar auf einen „sauberen“ Zustand (Aufspielen eines Festplatten-Images vor jeder Messung)
- Festlegung eines **Standardnutzungsszenarios** für die Anwendung der Software
 - Simulation einer realitätsnahen Nutzung der Software
 - Interaktion mit dem Anwender
 - Abarbeitung von Aufgaben, für die die Software entwickelt wurde
- Bedienung der Software über einen **Lasttreiber**, ohne das System Under Test durch die Lasttreiber-Software selbst zu belasten
- Mehrfache Durchführung der Messungen, um Fehler zu vermeiden

Im Vorhaben untersuchte Softwareproduktgruppen

	Produktgruppe	Architektur	Produkte
1	Textverarbeitung	Lokale Anwendung	Es wurden zwei Textverarbeitungsprogramme gewählt (TVP1 und TVP2). Bei TVP1 handelt es sich um ein proprietäres Produkt, TVP2 ist quelloffen.
2	Browser	Anwendung mit entfernter Verarbeitung	Es wurden drei Internetbrowser (B1, B2 und B3) gewählt. B1 und B2 sind quelloffen, B3 ist proprietär.
3	Content Management System	Anwendung mit entfernter Verarbeitung	Es wurden drei CMS (CMS1, CMS2 und CMS3) gewählt. Alle CMS sind quelloffen.
4	Datenbank	Serverdienst	Es wurden drei Datenbanksysteme (DB1, DB2 und DB3) gewählt. DB1 und DB2 sind quelloffen, DB3 ist proprietär.

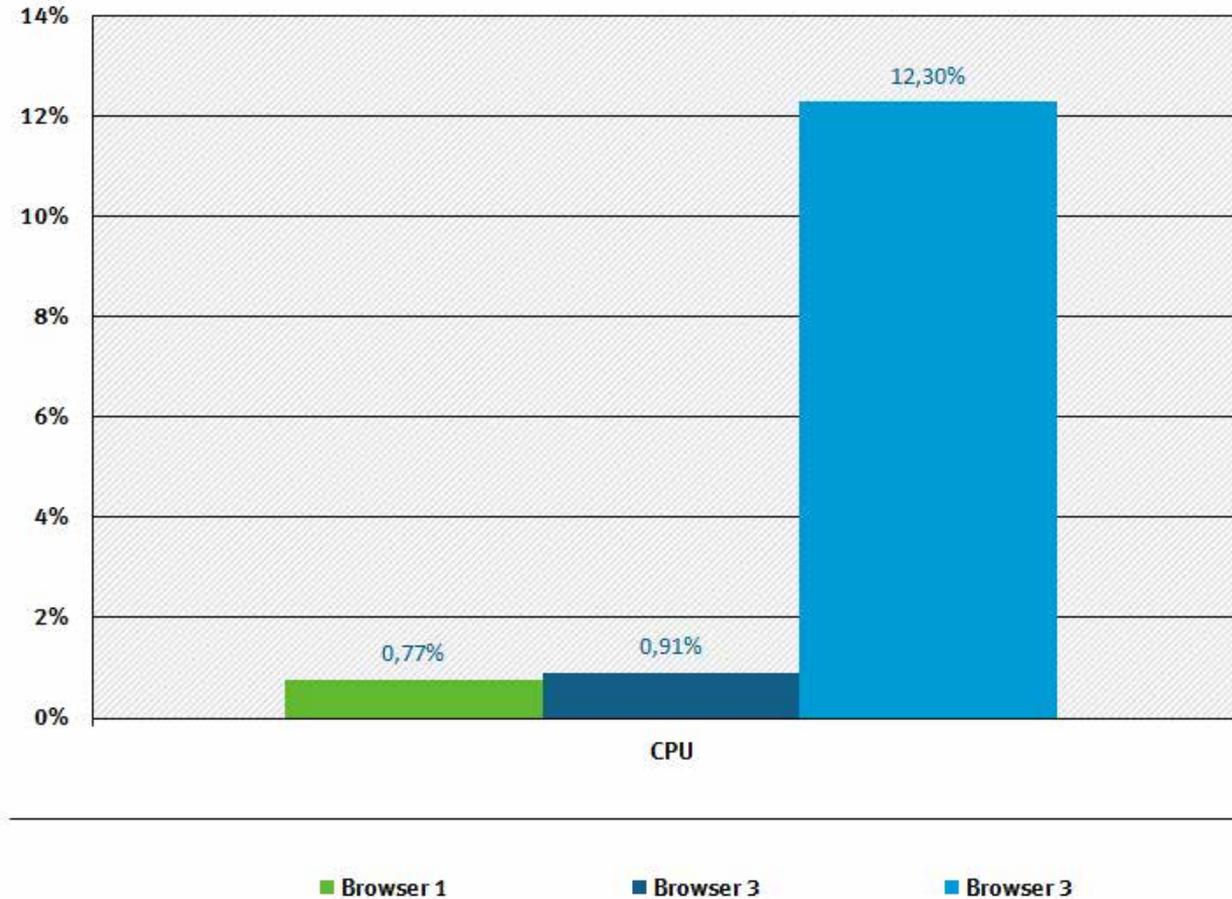
Exemplarische Messergebnisse Leerlauf (1)

- Hardware-Auslastung im Leerlauf zweier Textverarbeitungsprogramme



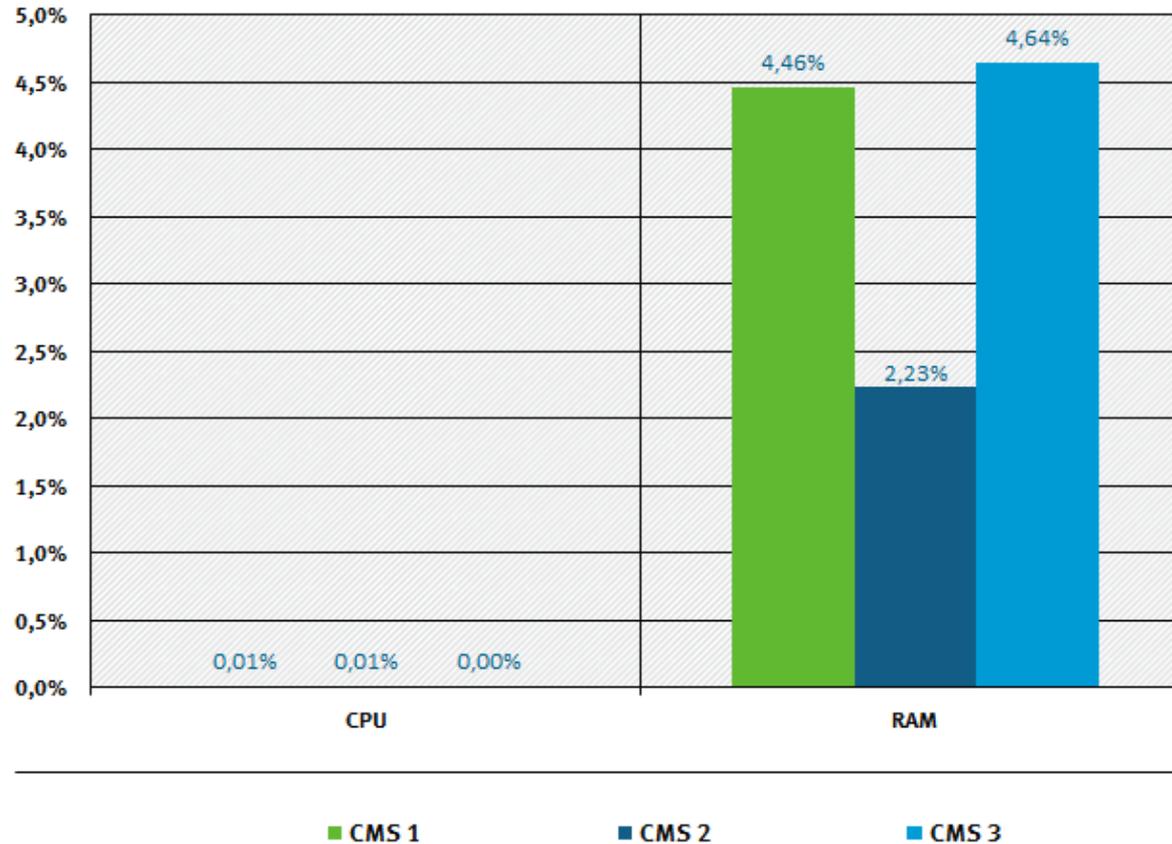
Exemplarische Messergebnisse Leerlauf (2)

- Hardware-Auslastung (CPU) im Leerlauf dreier Internetbrowser



Exemplarische Messergebnisse Leerlauf (3)

- Hardware-Auslastung im Leerlauf dreier Content Management Systeme (CMS) auf einem Server



Standardnutzungsszenarien

Textverarbeitungsprogramm

1. Gesamten Text bearbeiten
2. Inhaltsverzeichnis einfügen & aktualisieren
3. Ansicht anpassen
4. Inhalte hinzufügen & bearbeiten
5. PDF erzeugen
6. Speichern

Browser

1. E-Mail lesen/schreiben
2. Youtube-Video anschauen
3. Online-Shop besuchen
4. Lesezeichen setzen
5. Add-on installieren
6. Datei downloaden

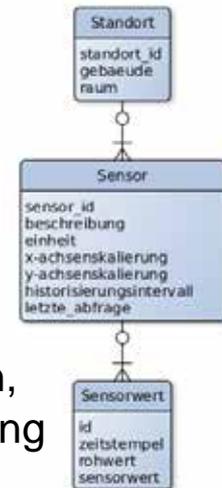
CMS

1. Kommentare beantworten
 2. Neue Seite erstellen
 3. Alle Seiten veröffentlichen
 4. PDF-Dateien hochladen
 5. PDF-Dateien verlinken
 6. Seite betrachten
- Zusätzlich: Lastgenerierung zur Simulation von Besuchern

Datenbank

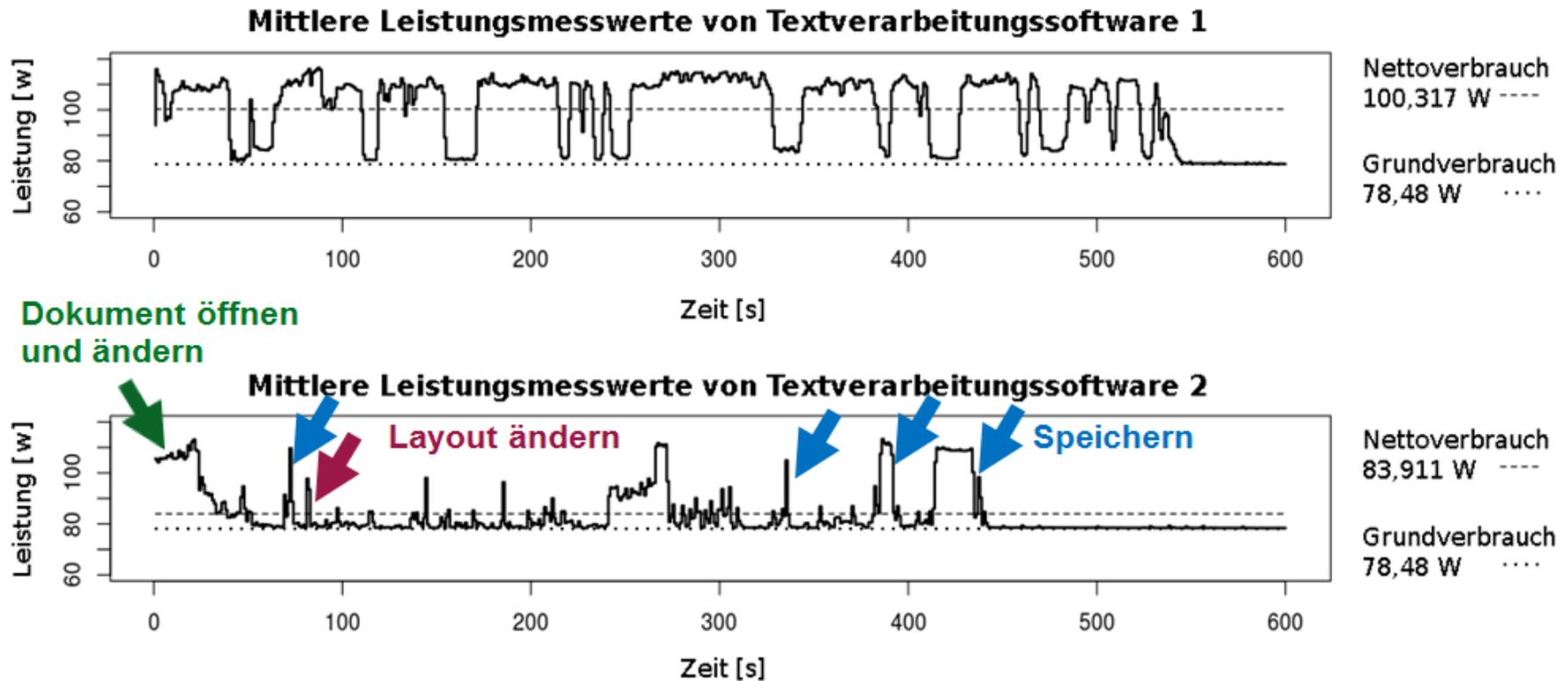
Schema bereits vorhanden

1. Daten eintragen
 2. Daten lesen
 3. Daten verändern
 4. Daten löschen
- 230 Durchgänge je Funktion,
120.000 Zugriffe je Durchgang



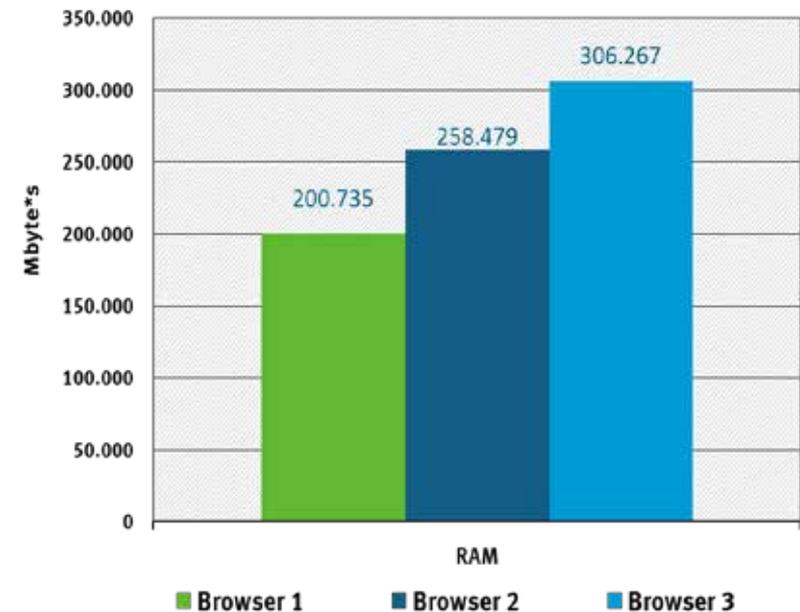
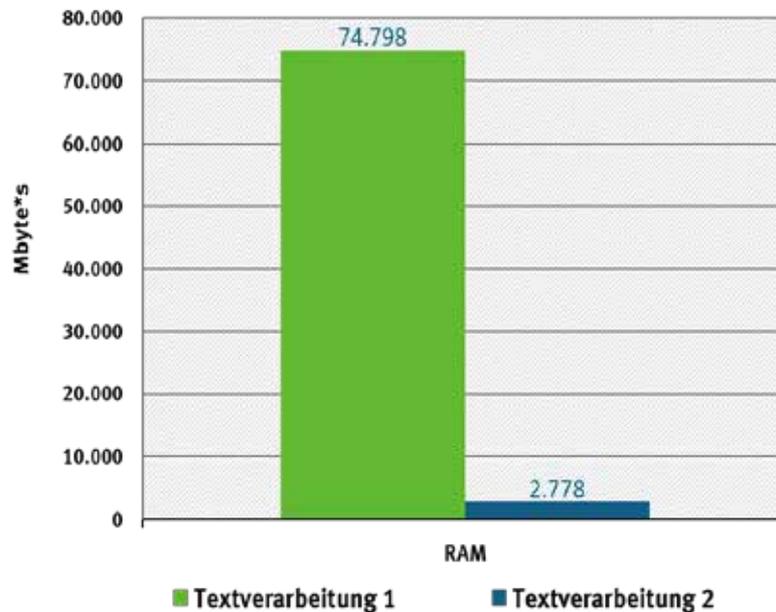
Beispiel eines Messzyklus Standardnutzungsszenario

- Messung der Nutzung der Software über 10 Minuten



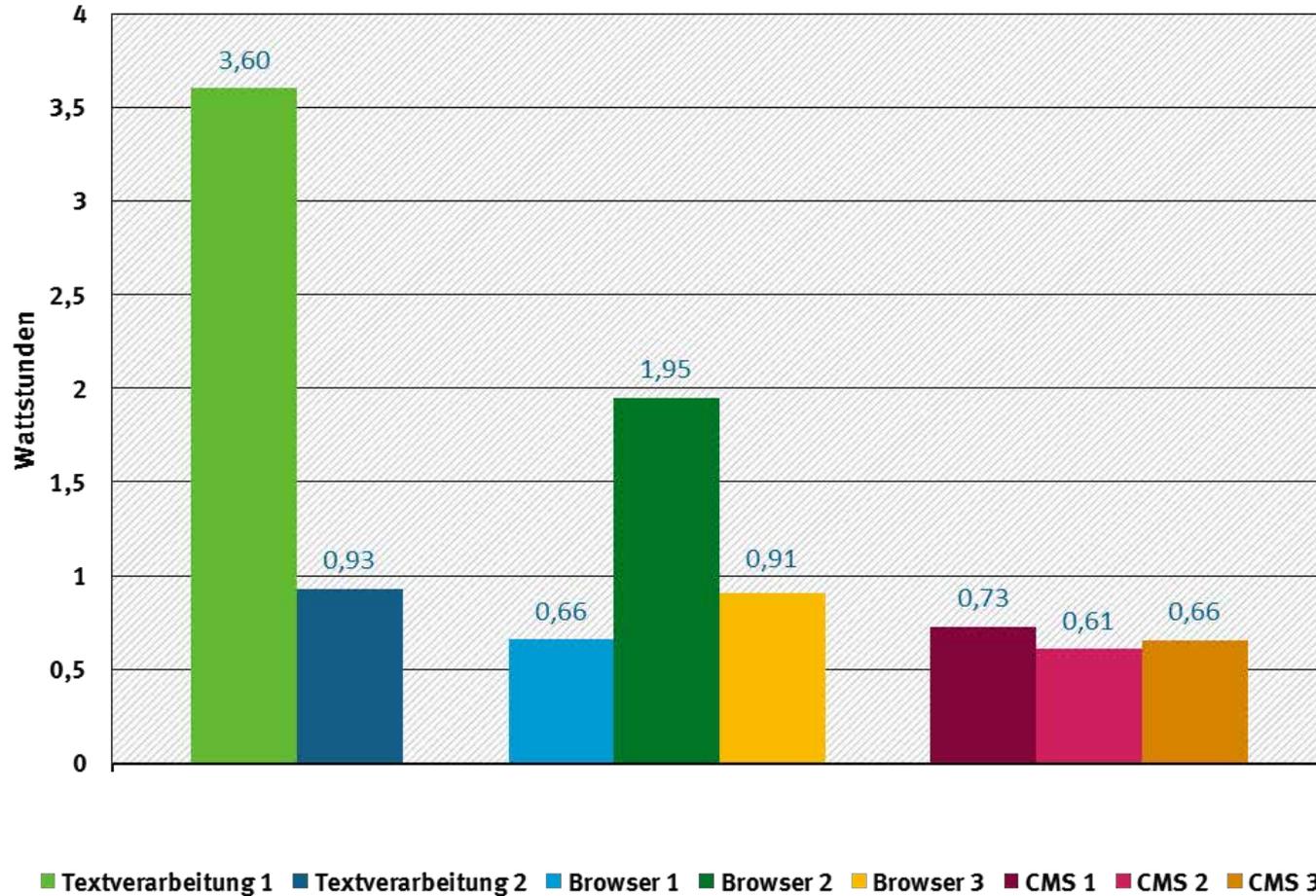
Exemplarische Messergebnisse Nutzungsszenario (1)

- Inanspruchnahme von Arbeitsspeicher bei Ausführung eines Standardnutzungsszenarios



Exemplarische Messergebnisse Nutzungsszenario (2)

- Energieverbrauch bei Ausführung eines Standardnutzungsszenarios



Kriterienkatalog Sustainable Software Design



Vorstellung des Kriterienkatalogs (1)

1 Ressourceneffizienz

- In welchem Ausmaß werden bei gegebener Funktionserfüllung Hardwarekapazitäten und damit indirekt natürliche Ressourcen beansprucht?

2 Potenzielle Hardware-Nutzungsdauer

- Zu welchem Grad sind Hardware-Erneuerungszyklen von Software-Erneuerungszyklen entkoppelt?

3 Nutzungsautonomie

- Respektiert der Hersteller des Softwareprodukts die Autonomie des Nutzenden im Umgang mit dem erworbenen Produkt?

Vorstellung des Kriterienkatalogs (2)

1 Ressourceneffizienz

1.1 Hardwareeffizienz

- Welche Hardwarekapazitäten müssen für den Betrieb des Softwareprodukts bereitgehalten werden und wie werden sie während des Betriebs ausgelastet?

1.2 Energieeffizienz

- Wie viel elektrische Energie verbraucht die Hardware bei Nutzung des Softwareprodukts zur Ausführung eines Standardnutzungsszenarios?

1.3 Ressourcenmanagement

- In welchem Ausmaß trägt das Softwareprodukt während seines Betriebs zu einem effizienten Management der von ihm beanspruchten Ressourcen bei?

Vorstellung des Kriterienkatalogs (3)

2 Potenzielle Hardware-Nutzungsdauer

2.1 Abwärtskompatibilität

- Garantiert der Hersteller des Softwareprodukts, dass das aktuelle Release auf einem Referenzsystem von vor n Jahren betrieben werden kann?

2.2 Plattformunabhängigkeit und Portabilität

- Kann das Softwareprodukt auf verschiedenen aktuell verbreiteten produktiven Systemumgebungen (Hardware und Software) betrieben werden und können die Nutzenden zwischen diesen ohne Nachteil wechseln?

2.3 Hardwaresuffizienz

- Bleibt die Menge an beanspruchter Hardwarekapazität bei Weiterentwicklung des Softwareprodukts auch bei Funktionserweiterungen über die Zeit konstant?

Vorstellung des Kriterienkatalogs (4)

3 Nutzungsautonomie

3.1 Transparenz

- Sind ressourcenrelevante Aspekte des Softwareprodukts für Nutzende mit vernünftigen Aufwand nachvollziehbar? Können Nutzende die Daten, die sie mit dem Softwareprodukt erzeugt haben, mit anderen Softwareprodukten weiterverwenden?

3.2 Deinstallierbarkeit

- Lässt sich das Softwareprodukt einfach, rückstandsfrei und ohne vermeidbare Nachteile deinstallieren?

3.3 Wartungsfunktionen

- Bietet das Softwareprodukt einfach zu benutzende Funktionen, die es erlauben, eingetretene Schäden an Daten und Programmen zu beheben?

3.4 Unabhängigkeit von Fremdressourcen

- Lässt sich das Softwareprodukt möglichst unabhängig von Ressourcen betreiben, die nicht der Kontrolle der Nutzenden unterliegen?

3.5 Qualität der Produktinformation

- Unterstützt die angebotene Information über das Softwareprodukt seine ressourcenschonende Nutzung?

Vorstellung des Kriterienkatalogs (5)

- Kriterienkatalog online verfügbar unter: <http://green-software-engineering.de/kriterienkatalog>
- Ergänzend verfügbar
 - Auswertungssoftware zu Hardwareauslastung und Energieverbrauch
 - Erfassungstool zur Erhebung der Bewertungskriterien und zum Export als XML-File
- Forschungsbericht soll in Kürze beim Umweltbundesamt veröffentlicht werden: <https://www.umweltbundesamt.de>



Ausblick



Ausblick

- Anwendung der Kriterien für nachhaltige Software beim **öffentlichen Einkauf** von fertiger Software und bei der Beauftragung von Software-Programmierleistungen
- **Freiwillige Anwendung** des Kriterienkataloges durch Softwareentwickler
- Entwicklung eines Umweltzeichens **Blauer Engel** für nachhaltige Softwareprodukte (läuft bereits)
- Auszeichnung von nachhaltiger Software und Kennzeichnung mit einem „**Nachhaltigkeits-Fußabdruck**“ (XML-Datei zur Software mit Software-Eigenschaften)



Vielen Dank für Ihre Aufmerksamkeit!

Jens Gröger

Öko-Institut e.V. - Büro Berlin

Bereich Produkte & Stoffströme

Tel.: 030 – 40 50 85 – 378

j.groeger@oeko.de



Weiterführende Informationen zum Projekt:

<http://green-software-engineering.de/kriterienkatalog>