ÖFIT-Trendschau

Bibliographische Angaben

Öffentliche Informationstechnologie in der digitalisierten Gesellschaft

Trendthema 55

Grüne Software

Herausgeber:

Kompetenzzentrum Öffentliche IT
Kompetenzzentrum Öffentliche IT
Fraunhofer-Institut FOKUS
Kaiserin-Augusta-Allee 31, D-10589 Berlin
Telefon: +49 30 3463 - 7173
Telefax: + 49 30 3463 - 99 - 7173
info@oeffentliche-it.de
www.oeffentliche-it.de
www.fokus.fraunhofer.de

Autorinnen und Autoren der Gesamtausgabe: Mike Weber, Stephan Gauch, Faruch Amini, Tristan Kaiser, Jens Tiemann, Carsten Schmoll, Lutz Henckel, Gabriele Goldacker, Petra Hoepner, Nadja Menz, Maximilian Schmidt, Michael Stemmer, Florian Weigand, Christian Welzel, Jonas Pattberg, Nicole Opiela, Florian Friederici, Jan Gottschick, Jan Dennis Gumz, Fabian Manzke, Rudolf Roth, Dorian Grosch, Maximilian Gahntz, Jens Fromm

Autorinnen und Autoren einzelner Trendthemen: Michael Rothe, Oliver Schmidt

ISBN: 978-3-9816025-2-4

Autorinnen/Autoren: Dorian Grosch

Bibliographische Angabe: Dorian Grosch, Grüne Software, In: Jens Fromm und Mike Weber, Hg., 2016: ÖFIT-Trendschau: Öffentliche Informationstechnologie in der digitalisierten Gesellschaft. Berlin: Kompetenzzentrum Öffentliche IT,
https://www.oeffentliche-it.de/-/grune-software

Lizenz: Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung 3.0 Deutschland Lizenz (CC BY 3.0 DE) http://creativecommons.org/licenses/by/3.0 de/legalcode. Bedingung für die Nutzung des Werkes ist die Angabe der Namen der Autoren und Herausgeber.

Grüne Software



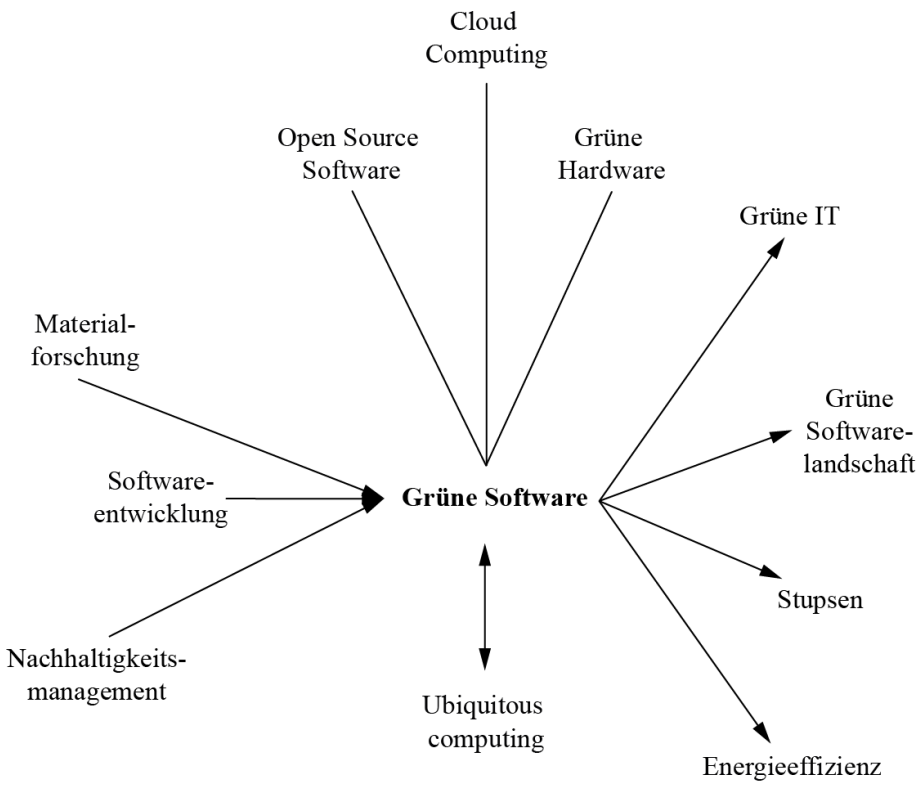
Software ist die Seele von Computern. Sie steuert Prozesse, dient zur Kommunikation und ermöglicht die Schaffung der vielseitigsten Werkzeuge unserer Zeit. Auch wenn Software immateriell ist, hat ihr Betrieb reale Kosten über die Lizenzgebühren hinaus: den Stromverbrauch, der je nach Programmierung unterschiedlich ausfallen kann, und die Rohstoffe für die Hardware, auf der die Software ausgeführt wird. Digitalisierung bietet so nicht nur einen wichtigen Lösungsbaustein für eine nachhaltige Lebensweise, mit dem rasant ansteigenden Einsatz digitaler Anwendungen explodiert auch deren Ressourcenverbrauch. Software ist Lösungsbaustein und Teil des Problems zugleich. Kann Software so gestaltet werden, dass sie ressourceneffizient und nachhaltig ist?

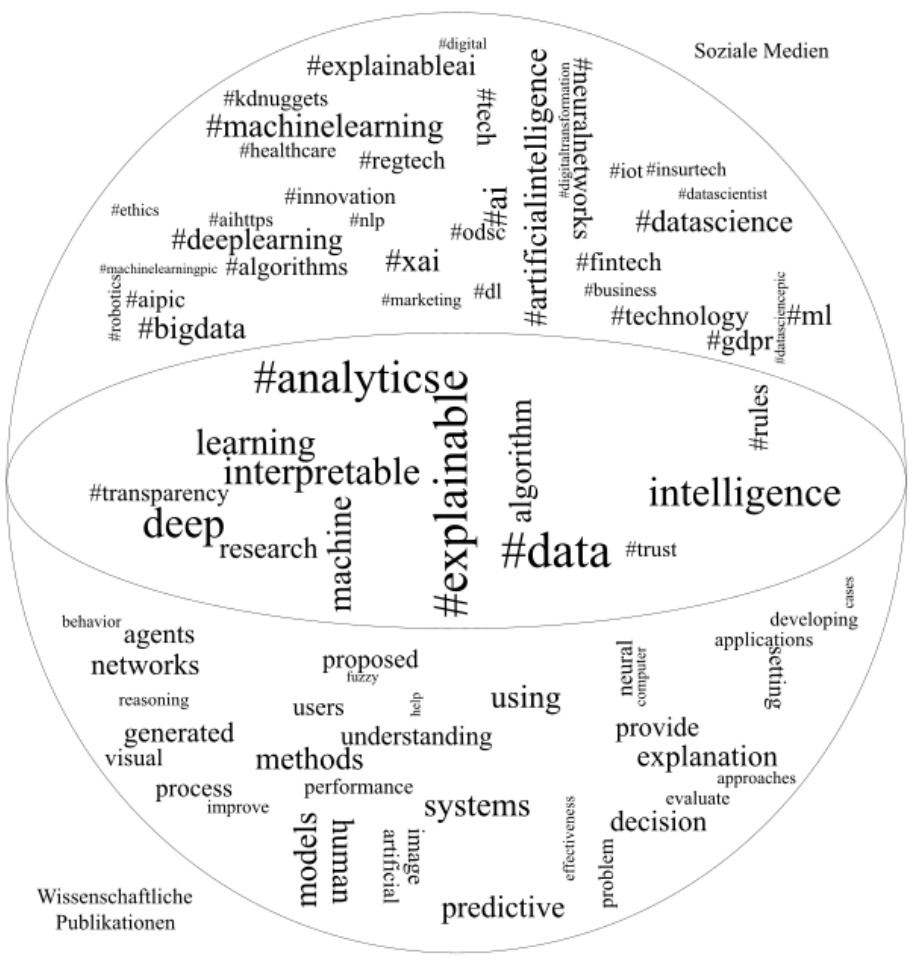
Unsichtbare Stromfresser

Wenn Software auf einem Computer ausgeführt wird, verbraucht ihr Betrieb Strom. Dieser ist zwar schwer von dem Stromverbrauch der Hardware zu trennen und erscheint deshalb leicht als unbedeutend, doch bestimmt die ausgeführte Software letztlich die Höhe des Energieverbrauchs, wie anhand der **Blockchain** bereits kontrovers diskutiert wird. Insbesondere wenn Software ineffizient programmiert oder nicht für die eingesetzte Hardware optimiert wurde, verursacht sie erhöhten Stromverbrauch und damit vermeidbare Kosten. Dabei muss der Stromverbrauch nicht vor Ort anfallen, wenn ressourcenintensive Rechenoperationen auf entfernte Computersysteme (siehe **Cloud Computing**) ausgelagert werden.

Die benötigte Hardware schlägt sich durch ihren aufwändigen Herstellungsprozess ebenfalls signifikant auf die Ressourcenbilanz nieder. Zudem agiert Software nicht in einem Vakuum, sondern ist eingebettet in und interagiert mit technologischen und sozialen Prozessen. Deswegen müssen indirekte Effekte zu den Auswirkungen von Software hinzugezählt werden, beispielweise wenn Software menschliches Verhalten und technologische Systeme beeinflusst.

Begriffliche Verortung





Was ist grüne Software?

Die Nachhaltigkeit von Software lässt sich ganz allgemein in drei Kategorien betrachten. Die erste Kategorie bilden die direkten Energiekosten und die Umweltkosten der benötigten Hardware und Infrastruktur. Sie lässt sich als Effizienz der IT-Systeme fassen. Es gibt eine Vielfalt von Ansätzen, um Software nachhaltiger zu machen. Die Effizienz von IT-Systemen lässt sich durch qualitativ hochwertige Programmierung, effizientere Hardware und Innovationen in der Hardwareproduktion verbessern. Dabei ist auch die Softwarearchitektur ausschlaggebend: Besonders bei Cloud-Desktop-Lösungen lässt sich durch sinnvolle Arbeitsverteilung

Selbstverstärkende und Rebound-Effekte definieren die dritte und letzte Kategorie. Dazu gehört das Phänomen, dass effizientere Software zu einer steigenden Nutzung digitaler Systeme beiträgt, weil beispielsweise im Rahmen der Effizienzsteigerung flüssigere und ansprechendere Bedienkonzepte realisiert werden konnten oder die Effizienzsteigerungen als Rechtfertigung für eine erheblich stärkere Nutzung herangezogen werden. Der Ressourcenverbrauch steigt also insgesamt an, weil der Ressourcenverbrauch der einzelnen Systeme pro Nutzung sinkt. Solche selbstverstärkenden Effekte sind oft das Resultat

Energie sparen. Außerdem liegt viel Potenzial in der effizienten Abstimmung von Hard- und Software. Durch die Berücksichtigung längerer Lebenszyklen für Softwareprodukte und eine bewusste Gestaltung der Entwicklungsprozesse lassen sich auch im Bereich der Softwareentwicklung Nachhaltigkeitsziele erreichen.

Die zweite Kategorie betrifft die indirekten Effekte, die Software etwa auf das Verbrauchsverhalten von Nutzer:innen oder auf die Energienutzung anderer technologischer Systeme hat. Diese Kategorie umfasst also die Wirkungen der Software. So lassen sich Ressourcen etwa durch intelligente und adaptive Steuerung von Smart Homes oder Industrieanlagen einsparen. Dabei können die Nachhaltigkeitsbewertungen für die einzelnen Komponenten und für das Gesamtsystem deutlich voneinander abweichen, wie sich an der effizienten Steuerung von Braunkohlekraftwerken und ihrer Bedeutung für die Nachhaltigkeit der Elektrizitätsversorgung leicht nachvollziehen lässt. Softwaresysteme können auch eine Rolle in der Energieeinsparung einnehmen, indem sie Nutzer:innen über ihren Energieverbrauch informieren und so zu einem nachhaltigen Verhalten anregen (siehe **Stupsen**).

komplexer Interaktionen wirtschaftlicher und gesellschaftlicher Phänomene. Dabei bedürfen Rebound-Effekte in der Regel einer umfassenden politischen Gestaltung, um die unerwünschten Wirkungen eingrenzen zu können.

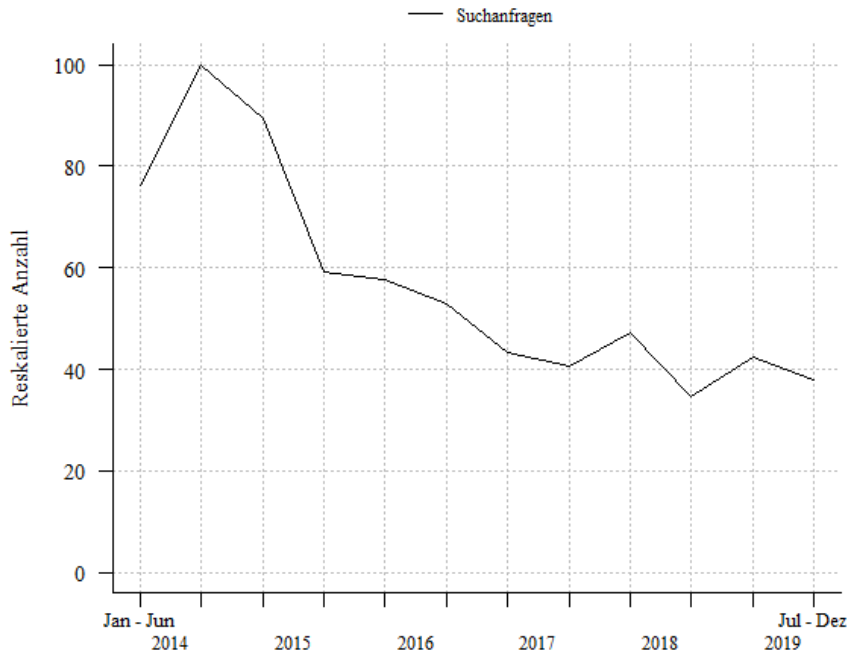
Wie grün ist Software?

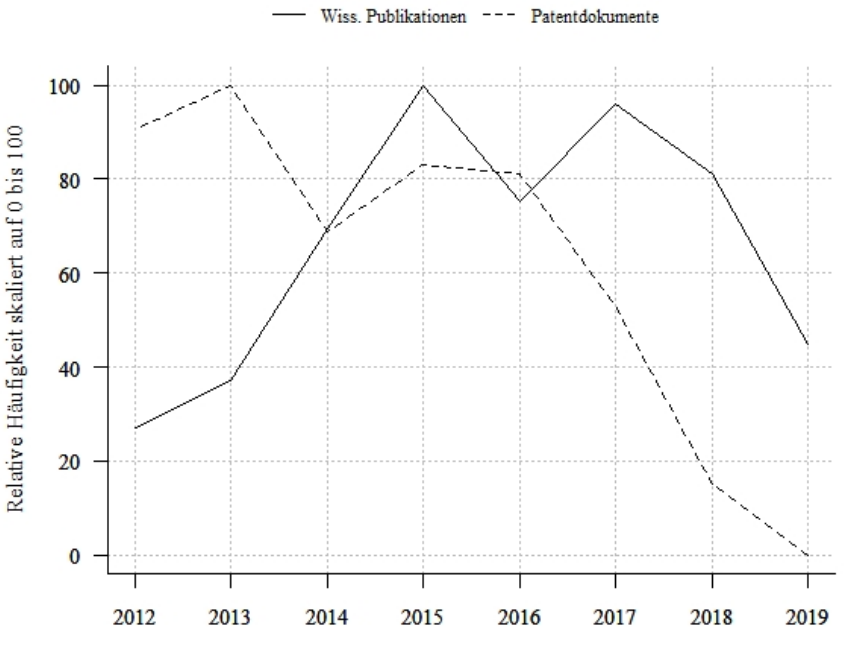
Das Ziel von grüner Software ist es, eine möglichst hohe Ressourceneffizienz zu realisieren und dabei die negativen Auswirkungen digitaler Technologien auf die Umwelt zu minimieren. Dabei sind Softwareentwickler:innen vor die Herausforderung gestellt, Funktionalität, IT-Sicherheit und Nutzererfahrung mit Nachhaltigkeitsaspekten zu verbinden. Neben Energieeffizienz sind dafür weitere Faktoren ausschlaggebend: Die Abwärtskompatibilität zu älteren Computern und Betriebssystemen und die Ungebundenheit der Nutzer:innen z. B. in Sachen Interoperabilität und Datenformate spielen eine signifikante Rolle, um die Lebensspanne von Hardware zu maximieren.

Software ist immer nur grün im Vergleich zu anderer Software für ähnliche Anwendungen. In bestimmten Anwendungsfällen wird erwartet, dass die Software kontinuierlich läuft. In anderen Fällen entscheiden die Benutzer:innen selbst, wann die Software gestartet und gestoppt wird. Außerdem spielt die Hardware eine große Rolle: ist die Software für einen Desktop-PC, ein Smartphone, ein IoT-Gerät oder einen Hochleistungsserver gedacht?

Aus diesen Gründen ist der Vergleich von Softwareprodukten nur innerhalb festgelegter Einsatzszenarien sinnvoll. Beispielsweise ließe sich mit einem standardisierten Messvorgang feststellen, welcher Desktop-Internetbrowser im Betrieb energiesparender ist. Um weitere Faktoren für die Nachhaltigkeit eines Softwareprodukts zu bestimmen, müssen zusätzlich funktionale Eigenschaften, der Entwicklungsprozess und die sozialen und technologischen Nebeneffekte der Software analysiert werden. Solche mehrstufigen und mehrdimensionalen Nachhaltigkeitsbewertungen von Softwareprodukten können sowohl für die Regulierung als auch beim Einkauf berücksichtigt werden.

Themenkonjunkturen





Der Weg zur grünen Softwarelandschaft

Viele IT-Unternehmen beanspruchen inzwischen für sich, grüne und nachhaltige Strategien zur Senkung der Energiekosten zu berücksichtigen und so zur ökologischen Nachhaltigkeit beizutragen. Solche Strategien adressieren gleichermaßen finanzielle Einsparpotenziale wie auch umweltpolitische Zielstellungen. Damit nachhaltige Softwareentwicklung im Mainstream ankommt, sind umfangreicher Wissenstransfer und politische Anreize notwendig. Das Konzept der Open-Source-Software bietet hierfür eine Möglichkeit, energieeffiziente Softwarebausteine kollektiv (weiter) zu entwickeln und gleichzeitig Methoden zum grünen Softwaredesign öffentlich zu vermitteln.

Trotz finanzieller Vorteile bei der effizienten Gestaltung soziotechnischer Systeme ist die Entwicklung hin zu grüner Software kein Selbstläufer. Die direkten und indirekten Wirkungen solcher Systeme können durchaus unter Nachhaltigkeitsaspekten problematisch sein. Deswegen ist es erforderlich, Software nicht isoliert zu betrachten, sondern Nachhaltigkeitsziele übergreifend umzusetzen. Dies setzt ein tiefes Verständnis von komplexen soziotechnologischen Interaktionen und ihrer vielfältigen Wirkungen voraus. Ein erster Schritt zur politischen Einflussnahme wäre die Berücksichtigung von Nachhaltigkeitsaspekten bei der Beschaffung von IT-Systemen. In Zukunft sollten also selbst IT-Infrastrukturanforderungen neu gedacht werden, um den Wandel zu grüner Software einzuleiten.

Folgenabschätzung

Möglichkeiten

- Energie- und Kosteneinsparung durch effizientere Software
- Umfassende Bewertung von Software nach Nachhaltigkeitskriterien
- Kollektives Engagement bei der Weiterentwicklung von grüner Open-Source-Software
- Erschließung neuer Marktnischen für nachhaltige Software
- Entwicklung eines umfassenderen, übergreifenden Nachhaltigkeitsbegriffs

Wagnisse

- Erhöhter Entwicklungsaufwand
- Vollständige Nachhaltigkeitsbewertung angesichts hoher Komplexität und zahlreicher Wechselwirkungen
- Erhöhte Nutzung effizienter Software (Rebound-Effekt)
- Spannungsverhältnis zwischen Benutzungskomfort, IT-Sicherheit und Ressourcenschonung

Handlungsräume

Nachhaltigkeit standardisieren

Methoden zur Bewertung von Software nach Nachhaltigkeitsaspekten sind noch im Entwicklungsstadium. Eine Standardisierung der relevanten Aspekte könnte bspw. die Entstehung und Etablierung eines Gütesiegels begünstigen.

Grüne Softwarebausteine fördern

Die Förderung von Softwareprojekten, die energieeffiziente Softwarebausteine entwickeln, ist eine gute Investition in die grüne Softwarelandschaft. Die Verfügbarkeit solcher Softwarebausteine trägt zur Verbreitung von grüner Software bei.

Nachhaltigkeit als Vergabekriterium berücksichtigen

Die Einführung von Nachhaltigkeit als Vergabekriterium bei Anschaffungen von IT-Softwarelösungen für die öffentliche Hand schafft Anreize für ressourcenschonende Softwareentwicklung und kann die Entstehung einer grünen Softwarelandschaft unterstützen.

Wissenstransfer unterstützen

Der Entwurf und die Vermittlung von Methoden, Leitlinien und Checklisten zu Softwareoptimierung sowie das Schaffen von Fortbildungsangeboten sind wichtige Elemente eines erfolgreichen Wissenstransfers.