

## Auswirkungen von Software-Systemen auf Nachhaltigkeit und die Mitverantwortung von Software-Entwicklern

Nachhaltigkeit ist ein wichtiges Thema. Software-Entwickler sind sich jedoch der Auswirkungen von Software-Systemen auf die Nachhaltigkeit nur vage bewusst. Darüber hinaus fehlen Methoden und Werkzeuge, die sie bei der Erstellung nachhaltiger Software-Systeme unterstützen. In diesem Artikel beschreiben wir ein Analyse-Framework, das Software-Entwickler dabei unterstützt, potenzielle Auswirkungen ihrer Software auf die Nachhaltigkeit schon während der System-Entwicklung zu berücksichtigen.



Konzept	Beschreibung
<b>Sustainability Debt</b>	„der versteckte Effekt früherer Entscheidungen über Software-intensive Systeme, die sich negativ auf die wirtschaftliche, technische, ökologische, soziale und individuelle Nachhaltigkeit des zu entwerfenden Systems auswirken. Effekte in diesen Dimensionen können sich auf drei verschiedenen Ebenen manifestieren: (1) die direkten Auswirkungen [...] (2) aktivierende Auswirkungen [...] und (3) systemische Veränderungen [...] [6]
<b>Nachhaltigkeitsdimensionen</b>	Man kann fünf Dimensionen der Nachhaltigkeit betrachten [6]:  Die <b>individuelle</b> Dimension bezieht sich auf die Aufrechterhaltung des Humankapitals (wie Gesundheit, Bildung, Fähigkeiten, Wissen, Führung und Zugang zu Dienstleistungen).  Die <b>soziale</b> Dimension zielt darauf ab, die gesellschaftlichen Gemeinschaften in ihrer Solidarität und ihren Diensten zu erhalten.  Die <b>wirtschaftliche</b> Dimension zielt darauf ab, Kapital und Wertschöpfung zu erhalten.  Die <b>Umweltdimension</b> bezieht sich auf die Verbesserung des menschlichen Wohlergehens durch den Schutz der natürlichen Ressourcen: Wasser, Land, Luft, Mineralien und Ökosystemleistungen.  Die <b>technische</b> Dimension bezieht sich auf die Langlebigkeit von Informationen, Systemen und Infrastrukturen und deren angemessene Entwicklung bei sich verändernden Umgebungsbedingungen.
<b>Wirkungsreihenfolgen</b>	Die Auswirkungen beeinflussen kurz-, mittel- und langfristig alle Dimensionen der Nachhaltigkeit.  <b>Direkte</b> Auswirkungen betreffen die unmittelbaren Einflüsse, die sich aus der Herstellung, Verwendung und Entsorgung von Software-Systemen ergeben, wie z. B. der Energieverbrauch.  <b>Aktivierende</b> Effekte befassen sich mit den Vorteilen und Auswirkungen der fortlaufenden Nutzung des Software-Systems, wie beispielsweise eine Websuchmaschine, die Kosten für den Zugriff auf Informationen senkt.  <b>Strukturelle</b> Effekte betreffen Veränderungen, die sich aus der mittel- bis langfristigen Nutzung von Software-Systemen durch eine sehr große Anzahl von Menschen ergeben, wie neue Gesetze, Politik oder soziale Normen.

Tabelle 1: Wichtigste Nachhaltigkeitskonzepte im Zusammenhang mit dem SusAF

Nachhaltigkeit bezieht sich auf die Fähigkeit, in ökologischen, wirtschaftlichen, sozialen, individuellen und technischen Kontexten Bestand zu haben [1] (siehe Tabelle 1). Das klingt nicht nur komplex – tatsächlich ist es das auch! Schauen wir uns an, inwieweit exemplarische Software-Systeme unsere Gesellschaft beeinflussen:

1. *Airbnb* bietet *Peer-to-Peer*-Kurzzeitmieten an. Studien zu den Auswirkungen von Airbnb in New York City zeigen, dass die Aussichten, mit der Vermietung über die Plattform 55 % mehr Geld zu verdienen, Tausende von Hauseigentümern dazu veranlasste, ihre Häuser vom Langzeitmietmarkt zu nehmen, was zu einem Anstieg der Mietpreise um 1,4 %

führte. Es wurde auch festgestellt, dass der Dienst eine Gentrifizierung hervorruft und die Segregation in der Stadt erhöht [2].

2. *Uber* und *Lyft* bieten einen Service für Online-Fahrdienst-Vermittler (Taxis) an. Eine kürzlich durchgeführte Studie [3] berechnete ihre Reiseeffizienz in Bezug auf Zeit und Entfernung, Auswirkungen auf das Parken, Verhaltensänderungen und Auswirkungen auf zukünftige Arten von Reisen. Positiv zu vermerken ist, dass der Bedarf an einem eigenen Auto geringer ist (wodurch auch die damit verbundenen Parkanforderungen gelöst werden) und folglich mehr Flexibilität bei der Kombination mehrerer Transportmittel für eine Reise

besteht. Auf der anderen Seite verlängert die Mitfahrgelegenheit die zurückgelegten Strecken im Vergleich zur Nutzung eines öffentlichen Verkehrssystems (im Durchschnitt um 9 Meilen) und die Fahrer haben unsichere Arbeitsbedingungen.

3. Entscheidungsfindung durch künstliche Intelligenz kann zu Diskriminierung mit weitreichenden Auswirkungen auf Menschen, Gesellschaften und Organisationen führen. Nehmen Sie zum Beispiel die Software COMPAS. Sie wird (in Teilen der USA) zur Vorhersage von Straftaten angewendet, wo sie Richter bei der Entscheidung unterstützt, ob ein Verurteilter Bewährung bekommen soll oder nicht. Obwohl das System keine Hautfarbe oder Rasse als Eingabe verwendet, stellt sich heraus, dass das System gegen dunkelhäutige Menschen voreingenommen ist [4] und ungenaue Vorhersagen macht [5]. Dies ist unter anderem auf historisch voreingenommene Trainingsdaten zurückzuführen und die Voreingenommenheit wird nun zementiert.

Dimension	Themen
Gesellschaftliche	Gemeinschaftssinn; Vertrauen; Inklusivität und Vielfalt; Gleichberechtigung; Teilnahme & Kommunikation
Individuelle	Gesundheit; Lebenslanges Lernen; Privatsphäre; Sicherheit; Behörden
Ökologische	Bedarfsstoffe und Ressourcen; Boden-, Luft- und Wasserverschmutzung; Energie; Biodiversität und Landnutzung; Logistik und Transport
Wirtschaftliche	Wert; Kundenpflege; Lieferkette Unternehmensführung und Prozesse; Innovation und F&E
Technische	Wartbarkeit; Benutzerfreundlichkeit; Erweiterbarkeit und Anpassungsfähigkeit; Sicherheit; Skalierbarkeit

Tabelle 2: Kernpunkte pro Dimension

Wie in diesen Beispielen gezeigt, kann die Entwicklung, Verwendung und Weiterentwicklung Software-intensiver Systeme zu neuen Auswirkungen auf die sie umgebenden sozialen und ökologischen Systemen führen. Solche Effekte können als *Sustainability Debt* [6] ausgedrückt werden (Definition siehe Tabelle 1). Um diese Schulden zu reduzieren, müssen Systeme entworfen werden, die die Nachhaltigkeit des sozio-technischen Systems, in das sie integriert sind, erhalten oder dazu beitragen.

Wir haben jedoch festgestellt, dass Software-IngenieurInnen, sobald sie beginnen, die Komplexität von Nachhaltigkeit zu verstehen, das Gefühl haben, nicht über das Wissen, die Erfahrung oder die Autorität zu verfügen, um die Software nachhaltiger zu machen, und sich eine angemessene Unterstützung bei solchen Bemühungen wünschen [7, 8].

Das hier vorgestellte *Sustainability Awareness Framework (SusAF)* (<https://zenodo.org/record/3676514>) bietet die diskursiven Strukturen und die Unterstützung, die erforderlich sind, um potenzielle Auswirkungen aufzudecken, die entstehen, wenn Software-Systeme entworfen und in die Gesellschaft eingebettet werden. Weitere Details zum Framework sind in [9] zu finden.

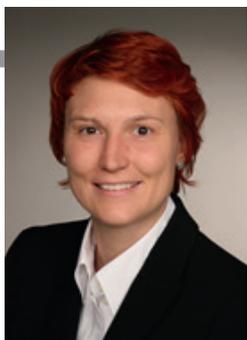
## Das Sustainability Awareness Framework

Das Sustainability Awareness Framework (SusAF) besteht aus einer Reihe von Fragen, Richtlinien, Beispielen und einem Visualisierungsdiagramm, die verwendet werden, um die Identifizierung und Diskussion der möglichen Auswirkungen eines Systems auf die Nachhaltigkeit zu fördern und zu steuern. Es wurde von der *Karlskrona-Allianz für nachhaltiges Softwaredesign* (siehe [www.sustainabilitydesign.org](http://www.sustainabilitydesign.org)) entwickelt und evaluiert. Das SusAF und dessen Elemente werden im Folgenden kurz diskutiert.

Die Fragen sind in fünf Gruppen unterteilt – eine pro Nachhaltigkeitsdimension (siehe Tabelle 1). Innerhalb jeder Dimension werden fünf Schlüsselthemen betrachtet, wie in Tabelle 2 dargestellt.

Die Themen in den Fragensätzen wurden bewusst auf fünf pro Dimension begrenzt, um eine erste einfache Übung zu ermöglichen, in der verschiedene Standpunkte zusammenge-

### Stefanie Betz



**Stefanie Betz** ist Professorin für Sozioinformatik an der Hochschule Furtwangen (HFU). Zuvor war sie Junior-Forschungsgruppenleiterin am Karlsruher Institut für Technologie und Senior Researcherin am Blekinge Institute of Technology in Schweden. Ihre Forschung konzentriert sich auf die Wechselwirkungen zwischen Gesellschaften und Software-Systemen. Sie untersucht, wie Software-Praktiker eine *Wirkungsperspektive* anstelle einer *Orientierungsperspektive* entwickeln können. Insgesamt argumentiert Dr. Betz, dass die entscheidende Rolle, die Software in der Gesellschaft spielt, einen Paradigmenwechsel in der Denkweise des Software Engineering erfordert. Sie ist Autorin mehrerer Publikationen zu Nachhaltigkeit, Gesellschaft und Software-Engineering und eine der Mitautoren des *Karlskrona-Manifests für Nachhaltigkeitsdesign* ([www.sustainabilitydesign.org](http://www.sustainabilitydesign.org)). Ihre Forschungsergebnisse wurden in Konferenzen und Fachzeitschriften veröffentlicht, darunter Communications of the ACM, das Journal of Software and Systems, IEEE Software, die Internationale Konferenz für Requirements Engineering und die Internationale Konferenz für Software Engineering.



bracht und das Bewusstsein aller Beteiligten für die Beziehung der Software zur Nachhaltigkeit geschärft werden sollen. Das Framework kann Themen hervorheben, die genauer untersucht werden sollten, oder auf übersehene Akteure verweisen. Zum Beispiel fragen wir zur Gleichberechtigung (unter sozialer Dimension): *Kann das System dazu führen, dass Menschen unterschiedlich behandelt werden?* Zum Beispiel, weil das System Datenanalysen durchführt oder menschliche Entscheidungen beeinflusst.

Zudem fördert das Framework, sobald die Fragen beantwortet wurden, das langfristige Denken, indem es die Teilnehmenden auffordert, Wirkungsketten für die diskutierten Punkte zu berücksichtigen und zeitliche Auswirkungen bei hoher Nutzerzahl zu berücksichtigen: *Was passiert, wenn viele Menschen weltweit dieses System jahrzehntelang verwenden?* Zum Beispiel, wenn Millionen von Menschen weltweit mehrere Jahre ein System nutzen.

Schließlich werden die Ergebnisse der Diskussion im in Abbildung 1 dargestellten Sustainability Awareness Diagram (SusAD) visualisiert. Das SusAD ist ein angepasstes Netzdiagramm, das in fünf Schichten unterteilt ist – eine für jede Nachhaltigkeitsdimension – und drei konzentrische Bereiche, die die direkten, aktivierenden und systemischen Auswirkungen eines Software-Systems darstellen (Definitionen siehe Tabelle 1). Ziel ist es, ein Skizzierwerkzeug bereitzustellen, um die Visualisierung und Diskussion der potenziellen Wirkungsketten zu vereinfachen; über Wirkungsordnungen und Nachhaltigkeitsdimensionen hinweg.

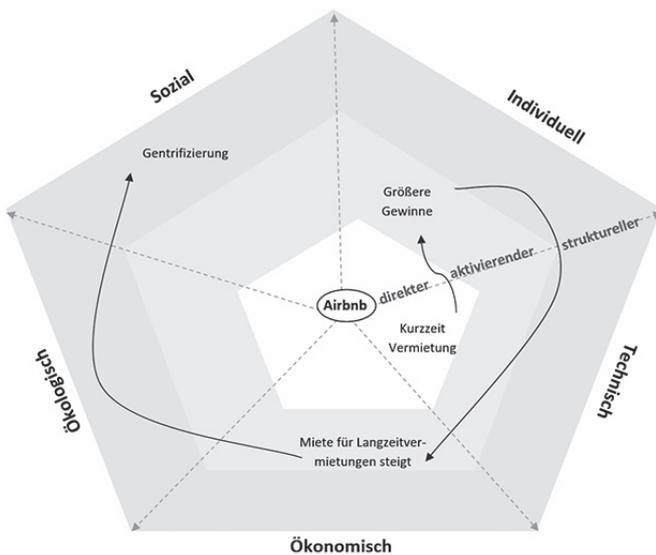


Abbildung 1: SusAD – Vereinfachtes Wirkungskettendiagramm für AirBnB. Das Beispiel zeigt, wie durch das Vorgehen der Eigentümer, ihre Immobilien an Kurzzeitbesucher zu vermieten, die Angebote auf dem Mietmarkt abnehmen. Dies führt wiederum zu einer Erhöhung der Durchschnittsmiete, was wiederum das Risiko einer Gentrifizierung erhöht.

Das Anwendungsformat für das Framework ist nicht vorgeschrieben. Es wurde sowohl im Rahmen der Anforderungserhebung mit Einzelinterviews [8] als auch in Workshops mit Stakeholdern angewandt. Der entscheidende Punkt ist, eine Diskussion über die genannten Fragen in dem für den jeweiligen Fall am besten geeigneten Format einzuleiten

## Fazit

Obwohl Software-Entwickler in der Regel nicht in der Lage sind, den Zweck eines zu entwickelnden Software-Systems zu definieren, entbindet sie dies nicht von der Verantwortung für die Auswirkungen, die dieses System in seiner letztendlichen sozio-technischen gelegenen Umgebung haben würde.

Software-Entwickler sollten *ausdrücklich Mitverantwortung* für die technologischen Lösungen übernehmen, die sie in die Gesellschaft einführen. So wären sie in einer starken Position, um die Diskussion unter den *Stakeholdern* über die möglichen Auswirkungen dieser Systeme zu moderieren. Das erfordert einen Paradigmenwechsel sowohl in der Praxis als auch in der Ausbildung der Software-Entwicklung [1, 10]. Dieser kann damit beginnen, dass Auftraggebern und Entwicklern aufgezeigt wird, dass Software-Systeme langfristige Auswirkungen auf die Gesellschaft haben werden und solche Auswirkungen im Software-Entwurfsprozess berücksichtigt werden müssen.

Zu diesem Zweck haben wir das SusAF entwickelt. Das Framework soll keine umfassende Analyse der Auswirkungen auf die Nachhaltigkeit ermöglichen. Stattdessen hilft es, über mögliche Nachhaltigkeitseffekte des Systems auf vereinfachte Weise nachzudenken, die dann wahrscheinlich in die Diskussion Fragen einbringen, die weitere Analysen und schließlich Änderungen an der Gestaltung des Systems fördern können.

Wir weisen darauf hin, dass das SuSAF-Framework als erster Schritt in Richtung eines solchen Paradigmenwechsels dienen und dazu beitragen kann, Software-Entwickler und -Entwicklerinnen für ihre Verantwortung zur Entwicklung nachhaltiger Software-Systeme zu sensibilisieren. Wollen Sie das SuSAF-Framework einmal ausprobieren, würden wir uns freuen, wenn Sie sich bei uns, den Autoren des Frameworks melden.

## Referenzen

- [1] C. Becker, S. Betz, C. Ruzanna, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, C. Venters, "Requirements: The Key to Sustainability," IEEE Software, vol. 33, no. 1, pp. 56–65, 2016, doi: 10.1109/MS.2015.158.
- [2] D. Wachsmuth and A. Weisler, "Airbnb and the rent gap: Gentrification through the sharing economy," Environment and Planning A: Economy and Space, vol. 50, no. 6, pp. 1147–1170, 2018, doi: 10.1177/0308518X18778038.
- [3] Impacts of Ridesourcing-Lyft and Uber-on Transportation Including VMT, Mode Replacement, Parking, and Travel Behavior, 2017. [Online]. Available: <https://pdfs.semanticscholar.org/e2cf/15b3a462917337062834c69213bf8ed41144.pdf>
- [4] J. Angwin, J. Larson, S. Mattu, L. Kirchner, "Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks," ProPublica, 23 May 2016; [www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing](http://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing).
- [5] J. Dressel and H. Farid, "The accuracy, fairness, and limits of predicting recidivism," Science Advances, vol. 4, no. 1, eaao5580, 2018, doi: 10.1126/sciadv.aao5580.
- [6] S. Betz, C. Becker, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, C. Venters, "Sustainability Debt: A Metaphor to Support Sustainability Design Decisions," in Proc. 4th Int'l Workshop



Requirements Eng. for Sustainable Systems (RE4SuSy 15), 24th August 2015, Ottawa, Canada.

[7] R. Chitchyan, S. Betz, L. Duboc, B. Penzenstadler, S. Easterbrook, C. Ponsard, C. Colin, "Evidencing sustainability design through examples", in Proc. 4th Int'l Workshop Requirements Eng. for Sustainable Systems (RE4SuSy 15), 24th August 2015, Ottawa, Canada.

[8] I. Groher and R. Weinreich, "An Interview Study on Sustainability Concerns in Software Development Projects," in 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Vienna, Austria, Aug. 2017 – Sep. 2017, pp. 350–358.

[9] L. Duboc, S. Betz, B. Penzenstadler, S.A. Kocak, R Chitchyan, O. Leifler, J. Porras, N. Seyff, C. Venters, "Do we Really Know What we are Building? Raising Awareness of Potential Sustainability Effects of Software Systems in Requirements Engineering," in 2019 IEEE 27th International Requirements Engineering Conference (RE), 2019, pp. 6–16.

[10] B. Penzenstadler, L. Duboc, C. Venters, S. Betz, N. Seyff, K. Wnuk, R. Chitchyan, S. Easterbrook, C. Becker, "Software Engineering for Sustainability: Find the Leverage Points!," IEEE Softw., vol. 35, no. 4, pp. 22–33, 2018, doi: 10.1109/MS.2018.110154908.

Lorenz M. Hilty

## Software und Nachhaltigkeit

### Wie Fremdbestimmung durch Software materielle Ressourcen entwertet

*Die digitale Transformation könnte eine Wirtschaftsweise ermöglichen, die natürliche Ressourcen schont. Software als immaterielles Produkt kennt keinen Verschleiß. Die universelle Hardware, die zur Ausführung von Software benötigt wird, kommt mit immer weniger Material und Energie pro Leistungseinheit aus. Dennoch wächst der ökologische Fußabdruck der digitalen Transformation. Warum gelingt es uns bisher nicht, Digitalisierung und Nachhaltigkeit in Einklang zu bringen?*

#### Software als immaterielles und nachhaltiges Produkt

Aus wirtschaftswissenschaftlicher Sicht ist Software ein immaterielles Produktionsgut. Als solches ist Software immateriell wie ein Roman oder ein Musikstück. Software benötigt zwar, wie alle Informationsgüter, ein materielles Trägermedium, doch dieses ist praktisch vernachlässigbar (Linde 2008).

erschienen in der FfF-Kommunikation, herausgegeben von FfF e.V. - ISSN 0938-3476 [www.fff.de](http://www.fff.de)

... dieses Modell ist es gleich schnell zu rechnen wie auf dem Papier. Im gleichen Zeitraum verdoppelt sich die Fläche des Rechenplatzes des Hamburger Rechenzentrums. Der Energiefaktor von rund einer Million gebildeten Transistoren pro Quadratmeter hat sich die Energieeffizienz der digitalen Hardware alle 1,57 Jahre verdoppelt (Aebischer & Hilty 2015).

Im Hinblick auf das Ziel nachhaltiger Produktions- und Konsummuster sind Software-Hardware-Kombinationen auf den ersten Blick ideale Produkte. Veränderte Bedürfnisse lassen sich ohne die Herstellung neuer Hardware erfüllen, denn es muss lediglich die immaterielle Software geändert werden. Software-Produktion ist Wertschöpfung, die nicht auf dem Umsatz von Tonnen von Material, sondern auf der präzisen Formulierung von Gedanken beruht (Hilty 2017). Man könnte also über längere Zeiträume mit der gleichen universellen Hardware leben und dennoch vom Fortschritt in der Software profitieren. Intelligente Steuerung und Regelung durch Software könnte außerdem viele andere material- und energieintensive Prozesse so optimieren, dass sie mit den Prozessen des globalen Ökosystems konsistent werden. Wenn diese Vision *nicht* Ihrer täglichen Erfahrung im Umgang mit Software-Produkten entspricht, so betrachten Sie sie zunächst einmal als ein optimistisches Szenario.

Allerdings hat diese erfreuliche Entwicklung auch ihre Schattenseite: In der heutigen digitalen Hardware ist mehr als das halbe Periodensystem enthalten, darunter auch sehr seltene und schwer zu gewinnende Metalle. Wir haben die relative Dematerialisierung dieser Technik also mit einer höheren Materialkomplexität bezahlt. Selbst unter optimalen industriellen Bedingungen werden im Elektronik-Recycling von 50-60 Metallen maximal 17 zurückgewonnen, alle anderen verteilen sich so fein, dass es praktisch nicht mehr möglich ist, sie jemals zurückzuholen. Wir entziehen diese – im Einzelfall winzigen, in der Summe aber relevanten – Metallmengen damit der Nutzung durch zukünftige Generationen (Hilty 2018). Hinzu kommt, dass wir weit davon entfernt sind, optimale industrielle Bedingungen für Recycling überhaupt zu verwirklichen. Selbst in den meisten EU-Ländern klafft zwischen entstehenden und rezyklierten Mengen von Elektronikschrott eine Lücke von über 50 % (Huisman et al. 2017).

#### Hardware als energie- und materialeffizientes Produkt

Im Jahr 1996 hat das deutsche Klimarechenzentrum in Hamburg das weltweit erste Klimamodell gerechnet, das gezeigt hat, dass der beobachtete Klimawandel vom Menschen verursacht ist. Wie viele Smartphones hätte man 25 Jahre später gebraucht,

Aus Sicht der Nachhaltigkeit wäre es also vernünftig, die digitale Hardware so sparsam und so lange wie möglich zu nutzen und das ursprüngliche Prinzip digitaler Technologie – die Software kommt und geht, die Hardware bleibt – auf breiter Basis zu rehabilitieren. Leider ist der gegenläufige Trend zu beobachten: Software wird dafür eingesetzt, den Konsum von Hardware zu beschleunigen. Materiell hochkomplexe technische Güter werden immer schneller zu Abfall.