



Themenkurzprofil Nr. 80
Juni 2025

Green Coding: Beitrag zu einer nachhaltigen digitalen Zukunft?

Mona Hille

In Kürze

Die zunehmende Digitalisierung und der Ausbau der Informations- und Kommunikationstechnik (IKT) führen zu einem stark wachsenden Energieverbrauch. Der IKT-Sektor ist bereits heute für einen erheblichen Anteil der globalen Emissionen verantwortlich. Insbesondere Rechenzentren verzeichnen einen kontinuierlichen Anstieg des Energiebedarfs. 2020 lag dieser bei rund 16 Mrd. Kilowattstunden. Software durchdringt alle Lebens- und Wirtschaftsbereiche und spielt eine Schlüsselrolle in der digitalen Transformation. Mit der verstärkten Nutzung von Clouddiensten rückt auch der Energieverbrauch von Software in den Fokus. Neben der Hardware beeinflusst auch die Effizienz des Programmcodes den Energieverbrauch maßgeblich: Schlechter Code erhöht Emissionen.

Ziel von Green Coding ist es, Software deutlich energieeffizienter zu gestalten, um dem steigenden Ressourcenbedarf der Digitalisierung ökologisch zu begegnen. Ein Ansatzpunkt ist etwa die Optimie-

rung von Code und Algorithmen. Auch die Wahl der Programmiersprache hat Einfluss, ist aber aufgrund bestehender Strukturen schwer zu ändern. Künftig könnte generative künstliche Intelligenz (KI) zur Effizienzsteigerung beitragen, etwa bei der automatisierten Codeoptimierung.

Trotz des hohen Energieverbrauchs fehlt es bislang an gesellschaftlichem und politischem Bewusstsein für das Thema. Selbst das Umweltzeichen Blauer Engel für Software ist seit seiner Einführung 2020 noch wenig bekannt. Verbindliche Standards und Messmetriken könnten dazu beitragen, das Einsparpotenzial besser auszuschöpfen. Gesellschaftlich wie politisch sind zentrale Fragen zu klären: Wie lassen sich bestehende Pfadabhängigkeiten überwinden? Welche Anreize, Regulierungen und Technologien sind nötig, damit Green Coding breit umgesetzt wird?

Hintergrund und Entwicklungsstand

Mit Green Coding werden Quellcode, Softwarearchitektur, Datenübertragung und Ausführungsverhalten (z. B. CPU- und Speicherzugriffe, Ausführungsgeschwindigkeit, Reaktion auf Benutzereingaben) gezielt so angepasst, dass der Energieverbrauch und die Nutzung von Hardwareressourcen durch Software während ihrer Entwicklung, Ausführung und Bereitstellung minimiert werden. Programmcode in Form von Software ist heute ein allgegenwärtiger Bestandteil nahezu aller Lebens- und Wirtschaftsbereiche. Ob in Smartphones, Fahrzeugen, Haushaltsgeräten, industriellen Produktionsanlagen oder medizinischen Geräten, digitale Steuerungs- und Informationssysteme sind in fast jedem technischen Produkt integriert. Selbst traditionelle Branchen wie die Landwirtschaft, das Bauwesen oder das Handwerk erfahren durch softwaregestützte Systeme, etwa durch Sensorik, Automatisierung oder datenbasierte Analyse, tiefgreifende Transformationen.

Angesichts der zunehmenden Verbreitung softwarebasierter Systeme und ihres wachsenden Ressourcenbedarfs (Energie, Rohstoffe etc.) gewinnt die Frage nach der ökologischen Verantwortung digitaler Technologien an Bedeutung. Green Coding setzt genau hier an und beschreibt die nachhaltige Softwareentwicklung. Bevor sich das Green Coding etablierte, lag der Nachhaltigkeitsfokus von Informationstechnologie (IT) vor allem auf der Optimierung von Hardware. Die Wurzeln reichen bis in die 1980er Jahre zurück, als Fujitsu mit seinem Green-IT-Programm umweltgerechte Produkte mit dem Fokus auf Recycling entwickelte und bereits 1993 für die Hardware des Green PC mit dem klassischen Blauen Engel zertifiziert wurde (Mauerer 2015). Eine Verfestigung des Nachhaltigkeitsansatzes in der IKT erfolgte in den frühen 2000er Jahren, als die Reduktion des Energie- und Ressourcenverbrauchs in der Hardwareproduktion, im Betrieb und bei der Entsorgung von IT-Geräten in den Mittelpunkt rückten (Dick/Naumann 2010; Naumann et al. 2011; Stobbe et al. 2009).

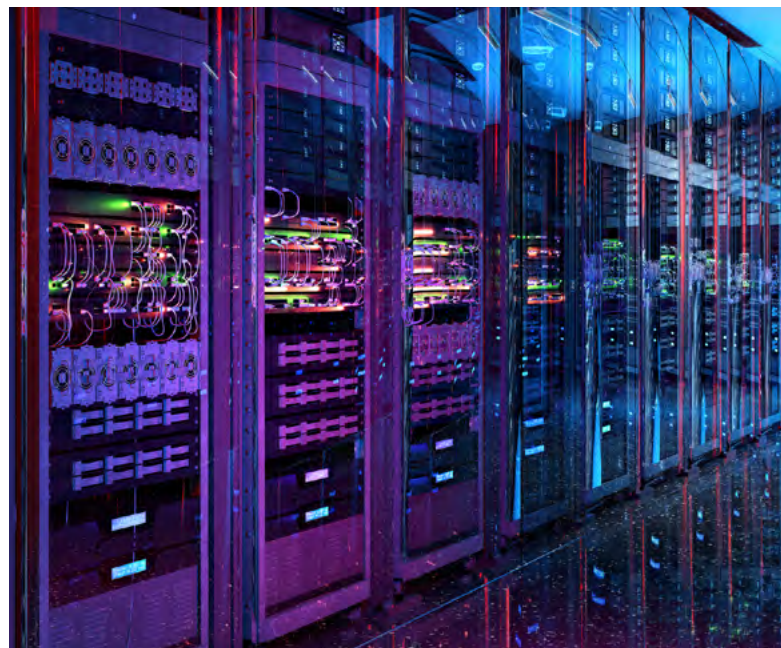
Zwischen 2010 und 2018 hatte sich die Auslastung von Rechenzentren, also die Menge an Rechenaufgaben, die dort verarbeitet werden, weltweit mehr als versechsfacht. Dies zeigt, wie stark digitale Dienste gewachsen sind (Radovanovic et al. 2023). Die Digitalisierung und der damit verbundene Ausbau von IKT-Infrastrukturen macht die IT zu einem bedeutenden Energieverbraucher (Bunse et al. 2013). Mit dem enormen Anstieg der Datenverarbeitung, unter anderem durch die zunehmende Verlagerung von Geschäftsprozessen in die Cloud (Gröger et al. 2018), rückt in Fachkreisen mittlerweile der Energieverbrauch von Software stärker in den Fokus (Gröger et al. 2018; ITU 2024; Junger et al. 2023; Kern et al. 2018; Rohde et al. 2021). 2024 belief sich der Stromverbrauch sämtlicher Rechenzentren auf rund 415 Terawattstunden (TWh), was etwa 1,5 % des weltweiten Stromverbrauchs entspricht, und wird sich bis 2030 voraussichtlich mehr als verdoppeln (IEA 2025, S. 13 f.). Auch die CO₂-Emissionen steigen mit der digitalen Transformation fortwährend an (Granskog et al. 2024; Janson 2022; Junger et al. 2023; The Shift Project 2019, S. 22 f.). Der IKT-Sektor verursacht Schätzungen zufolge zwischen 1,8 und 3,9 % der globalen CO₂-Emissionen und wächst nicht zuletzt durch die große Verbreitung von KI weiter (dena 2024). Besonders das Training und die Ausführung großer KI-Modelle erfordern erhebliche Rechenkapazitäten.

Ab 2020 gründeten sich Organisationen, wie beispielsweise die Green Software Foundation und das Marmelab, die daran arbeiten, wissenschaftliche Erkenntnisse und praktische Codingansätze der energieeffizienten Softwareentwicklung zusammenzuführen, um sie einer breiteren Zielgruppe, z. B. als frei verfügbare Tools und Codes zur Messung oder Reduktion von CO₂-Emissionen von Softwareprogrammen, als Open Source¹ zur Verfügung zu stellen. Ergänzend zu diesen Organisationen fördern Projekte, wie etwa „ecoCode“, das in Zusammenarbeit mit der Industrie durchgeführt wird, die Weiterentwicklung und Verbreitung von Standards in der Softwareprogrammierung mit dem Ziel, Energieeffizienz zu erhöhen und CO₂-Emissionen zu reduzieren (Le Goaër 2023).

Reduktion des Energieverbrauchs durch Green Coding

Software verbraucht Strom, indem sie Rechenprozesse auf der zugrunde liegenden Hardware auslöst. Beim Ausführen von Programmbefehlen werden elektrische Signale durch Transistoren in Prozessoren und Speicherbausteine geleitet, was physikalische Schaltvorgänge erfordert und entsprechend Strom verbraucht. Dieser Verbrauch hängt direkt mit der Struktur und Effizienz des Quellcodes, der Art und dem Umfang der auszuführenden Berechnungen sowie dem Volumen der verarbeiteten Daten zusammen.

1 Der Open-Source-Ansatz ermöglicht es Entwickler/innen, nachhaltige Codes frei zu nutzen und individuell anzupassen.



teten Daten zusammen. Auch Speichern und Übertragen von Daten sind softwaregesteuerte Prozesse, die Energie benötigen. Faktoren wie die Wahl von Algorithmen, die Struktur von Schleifen (Wiederholung von Anweisungen) und Datenobjekten oder die Nutzung von Parallelisierung beeinflussen maßgeblich den Energiebedarf. Dabei kann ineffizient programmierter Code zu unnötigen Rechenzyklen, hohem Speicherzugriff oder vermehrter Kommunikation mit Hardware führen und dadurch den Stromverbrauch erheblich erhöhen. Green Coding zielt auf die Reduktion dieses Energieverbrauchs ab. Im Schnitt lassen sich durch Green-Coding-Lösungen 15 bis 20 % des Energieverbrauchs von Software einsparen (Bergt 2025).

Eine effektive Maßnahme zur Senkung des Energieverbrauchs von Software bietet die Anwendung validierter Green-Software-Patterns (Entwurfsmuster) beim Design von Software. Diese werden über Plattformen wie GitHub² bereitgestellt und unterstützen Entwickler/innen dabei, energieeffiziente Softwarearchitekturen umzusetzen (dena 2024; Warncke/Mathew 2022). Beispiele dafür sind Cachingmechanismen, also Zwischenspeichermuster, die den Rechenaufwand und den Energieverbrauch reduzieren, indem redundante Datenbank- und Netzwerkanfragen vermieden werden (dena 2024), oder verzögertes Laden (Lazy Loading), bei dem Daten nur bei Bedarf geladen werden, wodurch unnötige Datenverarbeitung und Speicherzugriffe vermieden werden (Al-Hawari 2022).

Auch die Wiederverwendung energieeffizienter Codebausteine bei der Entwicklung von Software aus offenen Repositories³ trägt zur Reduktion des Ressourcenverbrauchs bei. Durch die optimierte Auslastung der Systeme wird so auch die Lebensdauer der eingesetzten Hardware verlängert (Pandey 2024; Zinoviev 2021). Ergänzend kommen zunehmend KI-gestützte Werkzeuge wie GitHub Copilot⁴, Code Optimizer⁵ von ChatGPT oder Amazon Q-Developer⁶ zum Einsatz, die Code analysieren und verbessern (Buchh 2023). Diese Auto-Coder-Tools können nicht nur die Energieeffizienz des Codes steigern, sondern auch den Aufwand in der Entwicklung senken. Gleichzeitig verursachen sie selbst einen beträchtlichen Strombedarf, was ihren Einsatz ambivalent macht (Vartziotis et al. 2024).

Ein weiterer Einflussfaktor ist die gewählte Programmiersprache. Hardwarenahe, kompilierte⁷ Sprachen wie C, C++⁸, Fortran

oder Rust benötigen durch ihre direkte Übersetzung in Maschinencode und effiziente Speicherverwaltung⁹ deutlich weniger Rechenressourcen und Energie als interpretierte Sprachen wie Python, Ruby oder Perl. C gilt als besonders energiesparend, während Python durch dynamische Typisierung und Laufzeitinterpretation¹⁰ bis zu 75-mal mehr Energie als C verbrauchen kann (González 2023; Hahn 2022; Pereira et al. 2017 u. 2021; Yagmur 2022). Neuere Programmiersprachen wie Julia oder Zig verbinden Energieeffizienz mit moderner Syntax, stoßen jedoch bei der Migration bestehender Systeme auf hohe Hürden. Denn der Umbau von Software und veraltetem Code (Legacy Code) auf eine neue Programmiersprache ist mit beträchtlichen Migrationskosten verbunden und auf die Akzeptanz in der Community angewiesen. Damit sich eine neue Programmiersprache durchsetzen kann, müssen bestehende Systeme aufwendig angepasst werden. Selbst wenn also die Vorteile auf der Hand liegen, ist es kein Automatismus, dass effiziente oder neue Programmiersprachen die populärste Programmiersprache Python in der breiten Anwendung ersetzen (Stack overflow 2022).

Spezialisierte Verfahren zur Messung des Energieverbrauchs von Software

Zur Identifikation der Einsparpotenziale werden spezialisierte Tools und Messverfahren eingesetzt, die den Energieverbrauch einzelner Softwarekomponenten präzise erfassen (Abdurachmanov et al. 2014; Cabot et al. 2019; Green Software Foundation 2025; Guldner 2024; Kern et al. 2013). Tools wie jPowerMonitor¹¹ messen etwa den Energieverbrauch von Java-Anwendungen¹² (Keilhofer 2022). Eco-CI¹³ analysiert den Energieeinsatz in CI/CD-Pipelines¹⁴ in Echtzeit. Das Green Metrics Tool¹⁵ identifiziert mittels reproduzierbarer Messungen und Visualisierungen Energiehotspots im Code und bietet datenbasierte Optimierungen für den gesamten Softwarelebenszyklus an. Um den Energieverbrauch von Webanwendungen sichtbar zu machen, lassen sich mit GreenFrame¹⁶ die CO₂-Emissionen quantifizieren (Zaninotto 2022). Auf Basis der Ergebnisse dieser Messungen lassen sich gezielte Maßnahmen zur Optimierung der Softwarekomponenten anstoßen, wie etwa die Reduktion redundanter Prozesse oder die Verbesserung von Speicherzugriffen (Abdurachmanov et al. 2014; Cabot et al. 2019; Green Software Foundation 2025; Guldner 2024; Kern et al. 2013).

2 <https://github.com/Green-Software-Foundation/patterns>

3 <https://github.com/Green-Software-Foundation/real-time-cloud>

4 <https://github.com/features/copilot>

5 <https://chatgpt.com/g/g-V4lByQfrs-code-optimizer>

6 <https://aws.amazon.com/de/q/developer/>

7 Kompilierte Sprachen wie C oder Rust generieren direkt Maschinencode, der ohne Laufzeitumgebung ausgeführt wird. Im Gegensatz dazu erfordern interpretierte Sprachen wie Python zusätzliche Schicht des Interpreters, die kontinuierlich Energie verbraucht.

8 Durch Compileroptimierungen sowie Templatemetaprogrammierung ist C++ äußerst effizient. Diese Programmiersprache eignet sich besonders für komplexe Systeme, die hohe Performance und nachhaltige Ressourcennutzung erfordern.

9 Effiziente Parallelisierung durch Threads (Rust) oder Low-Level-APIs (C) reduziert die Gesamtlaufzeit von CPU-intensiven Tasks. Python Global Interpreter Lock (GIL) limitiert dagegen echte Parallelverarbeitung, was Energie verschwendet.

10 Weil jede Variable während der Ausführung Typprüfungen braucht, werden zusätzliche CPU-Zyklen beansprucht.

11 <https://github.com/msg-systems/jpowermonitor>

12 Dabei werden die CPU-Nutzung sowie der kumulierte Energieverbrauch erfasst und detaillierte Einblicke in die Leistung der Software während der Laufzeit gegeben (Keilhofer 2022).

13 <https://github.com/green-coding-solutions/eco-ci-energy-estimation>

14 Continuous Integration und Continuous Deployment Environment: Bezeichnung für eine moderne Entwicklungsinfrastruktur, in der Software automatisch gebaut, getestet und bereitgestellt wird

15 <https://github.com/green-coding-solutions/green-metrics-tool>

16 <https://github.com/marmelab/greenframe-cli>



Grüner Strom allein reicht nicht aus

Dem steigenden Energieverbrauch in der IKT-Branche wird vermehrt dadurch begegnet, dass Rechenzentren mit Strom aus erneuerbaren Energien betrieben werden bzw. Rechenprozesse dann ausgeführt werden, wenn niedrige CO₂-Intensitäten erreichbar sind. Der Carbon-Intensity-Ansatz richtet den Betrieb von Software dynamisch an den aktuellen CO₂-Intensitätswerten des Stromnetzes aus (Green Software Foundation 2023; IONOS Redaktion 2023; Ruiz De Villa Suárez et al. 2021; Wirtschaft Digital BW 2024). So zeigte etwa IBM (o. J.), wie durch eine dynamische automatisierte Zuweisung und Steuerung von Aufgaben auf vorhandene Kapazitäten (Workload Scheduling) und optimierte Softwarearchitekturen der Energieverbrauch in Rechenzentren gesenkt werden kann. Der Carbon-Aware-Ansatz umfasst die Auslagerung von Rechenprozessen an Rechenzentren, die mit erneuerbarer Energie betrieben werden, bzw. den Neubau von Rechenzentren an Orten, an denen direkter Zugang zu erneuerbaren Energiequellen besteht. Google beispielsweise hat mit ihrem „Carbon-Intelligent Compute Management“ ein globales System aufgebaut, mit dem Rechenprozesse dort ausgeführt werden, wo die CO₂-Intensitätswerte am geringsten sind (Breukelman et al. 2024; Buchanan et al. 2023). Diese Ansätze sind ein verbreitetes Mittel, um digitale Prozesse mit einem hohen Anteil an erneuerbaren Energien auszuführen. Es ist allerdings ein Missverständnis, dass für einen nachhaltigeren Umgang mit Energie in der IKT ausreicht, die Umstellung auf erneuerbare Energiequellen zu forcieren. Die Vorstellung, dass erneuerbarer Strom unbegrenzt verfügbar ist und klimaneutral genutzt werden kann, bedarf einer differenzierten Betrachtung. Auch der Verbrauch erneuerbarer Energien ist nicht ohne ökologische

und systemische Auswirkungen. Ein wachsender Strombedarf belastet die Stabilität der Energieversorgung. Der Ausbau regenerativer Energiequellen ist mit dem Verbrauch begrenzter Rohstoffe, Flächen und Infrastruktur verbunden, der seinerseits ökologische und soziale Folgelasten mit sich bringt. Um diese zu minimieren, muss ein Wechsel zu regenerativen Energiequellen mit einer konsequenten Reduktion des Energieverbrauchs durch effiziente Softwaregestaltung Hand in Hand gehen.

Softwarezertifizierung durch den Blauen Engel

Ein Anreiz für Verbraucher/innen, nachhaltige Softwareprodukte zu erwerben, ist der Blaue Engel für ressourcen- und energieeffiziente Softwareprodukte, der seit 2020 vergeben wird. Er kennzeichnet Softwareprodukte, die in besonderem Maße sparsam mit Hardwareressourcen umgehen und auch einen niedrigen Energieverbrauch im Betrieb aufweisen. Zentrale Bewertungsparameter sind die Reduzierung des Energieverbrauchs, beispielsweise durch optimierte Softwarearchitekturen und effiziente Programmierung, Transparenz der Informationen zu ihren Systemvoraussetzungen, Messmethoden und Energieverbrauchskennzahlen sowie die Nutzungsautonomie. Letztere bedeutet einerseits, dass Software auch auf älteren Geräten lauffähig bleibt, und andererseits, dass Nutzer/innen die Möglichkeit haben, ihre Anwendungen flexibel zu steuern (Gröger et al. 2021; 2023). Bislang wurden lediglich drei Softwareanbieter ausgezeichnet: das Green Metrics Tool, der KDE Okular PDF-Viewer und der Nextcloud-Server (Blauer Engel o. J.). Um die Zertifizierung zu erhalten, ist es erforderlich, den Energieverbrauch der Software zu messen. Selbst wenn eine Software also die Kriterien erfüllt, die der Blaue Engel an ressourcen- und

energieeffiziente Software legt, muss der Softwareanbieter dies durch Messergebnisse belegen. Weil die Verbraucher/innen der Energieeffizienz von Software bislang keinen besonderen Wert beimessen – die Zertifizierung sich also wirtschaftlich nicht rechnet –, haben bislang lediglich die drei genannten Anbieter den Extraaufwand für die Zertifizierung auf sich genommen (Bergt 2025). Und das, obwohl Initiativen wie beispielsweise die Green Software Foundation und die Gesellschaft für Informatik die Weiterentwicklung nachhaltiger Codegenerierung erleichtern, indem sie technische Lösungsansätze zur Messung und Reduktion des Energieverbrauchs entwickeln und diese als Open-Source-Codepakete zur Verfügung stellen (Bühne et al. 2025; Hilty et al. 2009; Junger et al. 2023; Kolbe/Zarnekow 2013; Vartziotis et al. 2024). Entwickler/innen hätten insofern die Möglichkeit, durch den Einsatz von Green Coding bereits in den frühen Phasen der Modellentwicklung Softwareelemente zur Messung und Optimierung ihrer Programme zu integrieren (Verdecchia et al. 2023).

Anzeichen für Veränderungen in der Softwareentwicklung

Green Coding gewinnt seit 2020 deutlich an Bedeutung. Erste Nachhaltigkeitsinitiativen in der IT lassen sich bis in die 1980er Jahre zurückverfolgen. In den 2000er Jahren verlagerte sich der Fokus stärker auf Energie- und Ressourceneffizienz in der Hardwareproduktion, im Betrieb und in der Entsorgung von IT-Systemen. Seit 2020 rückt zunehmend die Softwareentwicklung in den Mittelpunkt. Auch große Technologieunternehmen zeigen seit einigen Jahren ein wachsendes Interesse an Green Coding hinsichtlich der Produkte und Dienstleistungen, die sie ihrer Kundschaft anbieten. Microsoft (2024) misst und reduziert den Stromverbrauch seiner Softwareangebote in Verbindung mit Clouddiensten. AWS bietet Kund/innen ein Tool zur Erfassung der durch ihre Cloudnutzung verursachten CO₂-Emissionen an¹⁷ (Amazon 2024). Gleichzeitig bleibt die konkrete Umsetzung dieser Ansätze innerhalb der Unternehmen selbst unklar. Wenngleich die großen Digitaldienstleister ihren Kund/innen Softwareoptimierungstools anbieten, finden sich in den Nachhaltigkeitsberichten dieser Unternehmen keine Ausführungen dazu, dass energieeffiziente Programmiermethoden systematisch in den eigenen IT-Infrastrukturen eingesetzt werden.¹⁸ Eine Ausnahme bildet der Audiostreaminganbieter Spotify. In dessen Nachhaltigkeitsberichten kann nachvollzogen werden, dass, obwohl die Zahl der monatlich aktiven Nutzer/innen um über 20 % gestiegen ist, das Wachstum der CO₂-Emissionen der Cloudprozesse auf einen Anstieg der Emissionen um 1,8 % im Vergleich zum Vorjahr begrenzt werden konnte (Grythberg 2024; Spotify 2024). Es ist anzunehmen, dass dies durch den

Einsatz spezifischer Tools zur Softwareoptimierung erreicht werden konnte (Spotify 2025, S. 12).

Gesellschaftliche und politische Relevanz

Insbesondere das steigende Angebot von Tools zur Messung und Reduktion von Energieverbräuchen zeigt, dass Green Coding als Handlungsfeld zunehmend an gesellschaftlicher und politischer Relevanz gewinnt. Die technische Basis ist vorhanden, erste Ansätze der Integration finden statt. Die großen Digitalunternehmen aber setzen die Tools, die sie ihren Kund/innen verkaufen, offenbar selbst nicht ein.

Transformationseffekte und Gestaltungsräume

Das Forschungsprojekt „Potentials of Green Coding“,¹⁹ das von der Hochschule Trier und der HTW Berlin durchgeführt wird, zielt darauf ab, das Bewusstsein für das Potenzial von Green Coding zu stärken und Wissen zur technologischen Umsetzung zu teilen (Wohlgemuth o. J.). Zentraler Ansatz ist dabei die Verankerung von nachhaltiger Softwareentwicklung in Lehrplänen, um zukünftige Generationen in Green Coding zu schulen. Angesichts des großen Potenzials der Senkung des Energieverbrauchs durch Anpassungen in der Software, ist der Bewusstseinswandel bei Unternehmen und Verbraucher/innen über die Chancen nachhaltiger Software ein wichtiger Faktor. Unternehmen, die Green-Coding-Prinzipien in ihre Entwicklungsprozesse integrieren, übernehmen Verantwortung und tragen aktiv zur nachhaltigen Entwicklung der Digitalisierung bei.

Der digitale Wandel ermöglicht es, traditionelle Abläufe zu überdenken und durch nachhaltige Lösungen zu ersetzen. Unternehmen, die sich auf die Entwicklung energieeffizienter digitaler Anwendungen konzentrieren, können Systeme nach ihren spezifischen Bedürfnissen gestalten und sich als Vorreiter positionieren. Solche Innovationsimpulse können dazu führen, dass Unternehmen schneller auf Marktveränderungen reagieren und ihre Strategien zukunftsorientiert gestalten (Hoffmann 2024; Lorenz o. J.). Über den technischen Bereich hinaus vermag es die digitale nachhaltige Transformation auch, einen kulturellen Wandel innerhalb von Organisationen zu fördern. Das kann nicht nur im Allgemeinen das Umweltbewusstsein und das Vertrauen in moderne Technologien stärken, sondern für die Unternehmen auch ganz konkret zu einem wirtschaftlichen Wettbewerbsvorteil führen (dena 2024; Junger et al. 2023).

Unbenommen der Chancen und Gestaltungsräume, die das Green Coding bietet, stellt die Umstrukturierung bestehender und über Jahre gewachsener Softwaresysteme Unternehmen vor erhebliche technologische und kulturelle Herausforderungen.

¹⁷ <https://aws.amazon.com/de/aws-cost-management/aws-customer-carbon-footprint-tool/>

¹⁸ <https://sustainability.aboutamazon.com/products-services/aws-cloud#increasing-efficiency>; dass AWS selbst energieeffiziente Software zur Emissionsreduktion einsetzt, kann nicht nachvollzogen werden.

¹⁹ <https://wohlgemuth.f2.htw-berlin.de/forschung/potentials-of-green-coding/>

Ein zentrales Hindernis sind die Migrationskosten. Der Wechsel der Programmiersprache ist mit grundlegenden unternehmerischen und betrieblichen Entscheidungen verbunden. Historisch gewachsener Code erfordert umfassende Überarbeitungen, um veraltete und ineffiziente Bestandteile zu verbessern. Dies ist mit erheblichem Aufwand sowie initialen Investitionen in Tools, Schulungen und Prozessumstellungen verbunden. Zwar ist zu erwarten, dass sich die Transformationsausgaben langfristig durch geringere Kosten und eine verbesserte Hardwarelaufzeit amortisieren, jedoch stellen sie kurzfristig eine erhebliche finanzielle Belastung dar (Blue Floyd 2023; dena 2024).

Förderung energieeffizienter KI-Software als zentraler Teil nachhaltiger Digitalpolitik

Künstliche Intelligenz (KI) existiert nicht ohne Code: Jede KI-Anwendung basiert auf einer programmierten Modellarchitektur, die festlegt, wie Daten verarbeitet werden. Erst auf dieser technischen Grundlage kann durch datenbasiertes Training ein lernfähiges Modell entstehen. Green Coding bietet hier ein zentrales Optimierungspotenzial, um die Energieverbräuche, die durch KI entstehen, zu verringern, da bereits durch gezielte Codegestaltung und algorithmische Effizienz deutliche Einsparungen erzielt werden können. Angesichts der rasanten Ausbreitung von KI-Systemen gewinnt dieses Feld strategisch an Bedeutung. Die hohe Zugänglichkeit und die große Verbreitung von KI-Tools zeigen sich deutlich in ihrer Nutzungsrate. Zwischen April 2023 und Oktober 2024 ist die Nutzung generativer KI in Deutschland deutlich angestiegen. Während im Frühjahr 2023 lediglich 23 % der Befragten entsprechende Systeme ausprobierten, lag dieser Anteil im Herbst 2024 bereits bei 53 %. Insgesamt nutzt rund ein Viertel der Befragten generative KI wie ChatGPT mehrmals pro Woche. Die Nutzungsrate von 78% bei der jüngeren Altersgruppe deutet darauf hin, dass diese Technologien immer stärker in den Alltag und das Berufsleben eindringen (Bühler 2024; Wintergerst

2024). Ob damit einhergehend auch die Aufmerksamkeit auf die von der Technologie ausgehenden Auswirkungen, wie beispielsweise dem Strom- und Wasserverbrauch, zunimmt, ist gegenwärtig noch unklar. Dennoch kann die Förderung energieeffizienter KI-Software ein wichtiges Element der Gestaltung einer nachhaltigen Digitalpolitik sein.

Gesellschaftlicher Bewusstseinswandel

Vielen Nutzer/innen ist nicht bewusst, wie viel Energie digitale Anwendungen verbrauchen. Gleichzeitig erwarten sie jederzeit schnellen Zugriff auf Informationen und Dienste. Genau hier entsteht ein Spannungsfeld zwischen Komfortansprüchen und dem Ziel, digitale Angebote umweltverträglicher zu gestalten. Nutzer/innen sind an sofortige Antwortzeiten und maximale Funktionalität, wie automatische Backups oder Echtzeitanalyse, gewöhnt. Das Ziel, den Erwartungen der Kund/innen gerecht zu werden, um auf dem Markt konkurrenzfähig zu bleiben, steht der Entscheidung für nachhaltige Software entgegen, die Rechenprozesse unter anderem dadurch energieeffizienter macht, dass weniger Prozesse redundant und damit etwas langsamer ausgeführt werden (Blue Floyd 2023; Rüdiger 2023). Es bedarf eines gesamtgesellschaftlichen Bewusstseinswandels beim Management, den Entwickler/innen bis hin zu den Endnutzer/innen, damit nachhaltige Lösungen nicht als Verzicht wahrgenommen werden und Nachhaltigkeit zum festen Bestandteil der (Unternehmens-)Kultur wird (dena 2024).

Die Gründung der Green Software Foundation, die Einführung des Blauen Engels für Software, der wachsende wissenschaftliche Diskurs sowie die Entwicklung vielfältiger Open-Source-Codes zur Integration energieeffizienter Programmeinheiten sind Hinweise auf ein stärkeres Bewusstsein für die Bedeutung von Green Coding. Auch die Tools zur Messung des CO₂-Fußabdrucks, die große Clouddienstleister ihren Kund/innen zur Verfügung stellen, zeigen, dass das Potenzial energieeffizienter Software als wichtiger Hebel für mehr Nachhaltigkeit zunehmend erkannt wird. Bis jedoch signifikante Reduktionspotenziale gehoben werden können, ist es noch ein weiter Weg.

Erfordernis politischer Leitlinien und verbindlicher Standards

Technologische Lösungen allein reichen nicht aus, um die ökologischen und ökonomischen Effekte von Green Coding tatsächlich zu realisieren. Es braucht zusätzlich politische Leitlinien, gesellschaftliche Sensibilisierung und wirtschaftliche Anreize. Ein wesentliches Hemmnis in der praktischen Umsetzung ist die Herausforderung, die tatsächlichen CO₂-Emissionen von Software valide abzuschätzen. Weil es bislang keine standardisierten Messverfahren zur Quantifizierung oder verbindliche Kriterien gibt, um den Energieverbrauch von Software präzise zu erfassen und zu bewerten, dominieren inkonsistente Bewertungsansätze. Einige Unternehmen messen den Stromverbrauch pro Transaktion, andere pro Serverknoten – Vergleichbarkeit ist damit kaum



gegeben. Die Gesellschaft für Informatik arbeitet an einheitlichen Standards. Fortschritte in diesem Bereich sind essenziell, um objektive und vergleichbare Daten zur Nachhaltigkeit von Software zu erhalten und kontinuierliche Verbesserungen anzustoßen (Pereira et al. 2017). Damit Unternehmen ihre Software ressourcenschonend umbauen, bedarf es einheitlicher Standards und verbindlicher Messmetriken. Sie sind der Grundstein für nachhaltige Veränderungen. Die Förderung der Entwicklung solcher Metriken und Standards sollte als zentrales Element der Digitalisierungsbestrebungen in Deutschland verankert werden (dena 2024). Es sollte allerdings mitgedacht werden, dass die Einführung neuer Standards und Zertifizierungen für nachhaltige Softwaresysteme mit möglichst wenig bürokratischem Mehraufwand einhergeht. Denn dies könnte als weiteres Hemmnis für den Fortschritt der digitalen Transformation wirken. Deswegen sollten vereinfachte und automatisierte Dokumentationsanforderungen und Zertifizierungsprozesse sowie standardisierte Messverfahren von vornherein mitgedacht werden, um die Bereitschaft zur Implementation von Green Coding zu erhöhen.

Damit Green Coding als Querschnittsthema wahrgenommen wird, könnten Förderrichtlinien mit entsprechenden Aufforderungen ergänzt werden, aus denen hervorgeht, dass bei Erforschung und Entwicklung digitaler Technologien auch energieeffiziente Software entwickelt werden soll. Darüber hinaus könnten beispielsweise gezielte Green-Coding-Förderprogramme Unternehmen unmittelbar dabei unterstützen, die nachhaltige Softwareentwicklung voranzutreiben, oder steuerliche Vorteile bzw. veränderte öffentliche Vergabekriterien die Integration energieeffizienter Software beschleunigen.

Schließlich erscheint es sinnvoll, dass beispielsweise das Bundesumweltministerium kommunikative Impulse in Richtung der Gesellschaft und spezifisch der Nutzer/innen durch gezielte Informationskampagnen setzt. Die Potenziale von energieeffizienter Software und realisierte Einsparpotenziale könnten öffentlich sichtbar gemacht werden, um digitale Nachhaltigkeit als Qualitätskriterium gesellschaftlich zu verankern (dena 2024).

Mögliche vertiefte Bearbeitung des Themas

Das Kurzprofil bietet einen Überblick über den Entwicklungsstand und die Herausforderungen von energieeffizienter Softwareentwicklung. Green Coding ist ein vergleichsweise neuer Handlungsansatz, bei dem es bereits Best Practices und frei zugängliche Codepakete gibt, bislang aber keine allgemeinverbindlichen Standards etabliert wurden. Diese jedoch sind erforderlich, um eine systematische, skalierbare und energieeffiziente Entwicklung von Softwareanwendungen und deren Integration in bestehende und neue Systeme voranzutreiben.

Insbesondere in großen IKT-Strukturen, wie z. B. in Rechenzentren der öffentlichen Hand, bei cloudbasierten Verwaltungsprozessen oder im Rahmen föderaler IKT-Kooperationen, ist eine durchgängig nachhaltige Softwareentwicklung noch nicht auf der Tagesordnung. Dabei lassen sich bereits heute konkrete Handlungsoptionen identifizieren:

- systematische Analyse von Umsetzungstreibern und -barrieren bei der Einführung energieeffizienter Programmierstandards insbesondere in komplexen Softwareprojekten der öffentlichen Verwaltung;
- Integration von Green-Coding-Aspekten in bestehende Rahmenwerke, etwa im Kontext von Sicherheitsrichtlinien, Ausschreibungsbedingungen oder Digitalisierungsstrategien;
- Verankerung von Green Coding in Förderprogrammen, z. B. durch Berücksichtigung in der Bundesförderung für effiziente digitale Infrastrukturen oder im Rahmen (klimaorientierter) Innovationsprogramme.

Eine vertiefte Auseinandersetzung mit dem Thema durch das TAB erscheint derzeit nicht erforderlich. Zur Entwicklung technischer Standards, zur Sammlung praktischer Erfahrungen bei großen Softwareprojekten und zur Weiterentwicklung des regulatorischen Rahmens kann das TAB aktuell nur begrenzt beitragen.

Literatur

- Abdurachmanov, D. et al. (2014): Techniques and tools for measuring energy efficiency of scientific software applications. In: Journal of Physics: Conference Series(608), Art. 012032, <https://doi.org/10.1088/1742-6596/608/1/012032##>
- Al-Hawari, F. (2022): Software design patterns for data management features in web-based information systems. In: Journal of King Saud University – Computer and Information Sciences 34(10, Part B), S. 10028–10043, <https://doi.org/10.1016/j.jksuci.2022.10.003##>
- Amazon (2024): Amazon Sustainability Report 2023. <https://sustainability.aboutamazon.com/2023-amazon-sustainability-report.pdf>
- Bergt, S. (2025): Den Recyclingansatz brauchen wir auch für Software. Blauer Engel für Software. TAZ, <https://taz.de/Blauer-Engel-fuer-Software/!6082820/> (17.6.2025)

- Blauer Engel (o. J.): Ressourcen- und energieeffiziente Softwareprodukte (DE-UZ 215). Zertifizierte Produkte. <https://www.blauer-engel.de/de/produktwelt/software> (17.6.2025)
- Blue Floyd (2023): Wie lässt sich Green Coding umsetzen? Definition „Nachhaltige Programmierung“. <https://www.datacenter-insider.de/green-coding-energieeffizienz-software-entwicklung-a-07a202532b5b0bad6504558b138b8856/> (17.3.2025)
- Breukelman, E. et al. (2024): Carbon-Aware Computing in a Network of Data Centers: A Hierarchical Game-Theoretic Approach. In: European Control Conference, S. 798–803, <https://doi.org/10.23919/ECC64448.2024.10591261##>
- Buchanan et al. (2023): Carbon-aware computing (Whitepaper). Measuring and reducing the carbon intensity associated with software in execution. Microsoft, o. O.
- Buchh, I. (2023): Amazon CodeWhisperer führt neue KI-gestützte Code-Korrekturen, Unterstützung für IaC und Integration mit Visual Studio ein. Amazon, <https://aws.amazon.com/de/blogs/germany/amazon-codewhisperer-fuehrt-neue-ki-gestuetzte-code-korrekturen-unterstuetzung-fuer-iac-und-integration-mit-visual-studio-ein/> (17.6.2025)
- Bühler, J. (2024): Zwei Jahre ChatGPT: KI-Nutzung, -Qualität und Risiken für die Demokratie. Pressekonferenz. TÜV-Verband, Berlin
- ühne, S. et al. (2025): Ressourceneffizienz im Software Lifecycle. Wie Ressourcenschonung, Langlebigkeit und Nachhaltigkeit in der Softwareentwicklung berücksichtigt werden können. Band 3: Leitfaden für Entwicklung & Betrieb. Bitkom e. V., Berlin
- Bunse, C. et al. (2013): Entwicklung und Klassifikation energiebewusster und energieeffizienter Software. In: Marx Gómez, J. et al. (Hg.): IT-gestütztes Ressourcen- und Energiemanagement. Berlin, S. 557–566
- Cabot, J. et al. (2019): Measuring Systems and Architectures: A Sustainability Perspective. In: IEEE Software 36(3), S. 98–100, <https://doi.org/10.1109/MS.2019.2897833##>
- dena (2024): Green Coding. Mit stromsparender Software zu einer nachhaltigeren Digitalisierung (Autor/innen: Hoffmann, G.-D. et al.). Deutsche Energie-Agentur GmbH, o. O.
- Dick, M.; Naumann, S. (2010): Enhancing Software Engineering Processes towards Sustainable Software Product Design. In: Greve, K.; Cremers, A. (Hg.): EnviroInfo 2010. Integration of Environmental Information in Europe. Proceedings of the 24th International Conference Proceedings of the 24th International Conference on Informatics for Environmental Protection. Aachen, S. 706–715
- González, M. (2023): Green Programming: Reducing your carbon emissions when coding. DataScience, <https://data-science.aero/green-programming-reducing-your-carbon-emissions-when-coding/> (17.6.2025)
- Granskog, A. et al. (2024): The role of power in unlocking the European AI revolution. McKinsey & Company, <https://www.mckinsey.com/industries/electric-power-and-natural-gas/our-insights/the-role-of-power-in-unlocking-the-european-ai-revolution> (17.6.2025)
- Green Software Foundation (2023): How Intesa and NTT Data Measure Software Carbon Footprint and Carbon Intensity. <https://greensoftware.foundation/articles/how-intesa-and-ntt-data-measure-energy-consumption-of-software> (17.3.2025)
- Green Software Foundation (2025): Calculating Your Carbon Footprint: A Guide to Measuring Serverless App Emissions on AWS. <https://greensoftware.foundation/articles/calculating-your-carbon-footprint-a-guide-to-measuring-serverless-app-emissions-o> (17.3.2025)
- Gröger, J. et al. (2018): Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik. Abschlussbericht. Umweltbundesamt, Dessau-Roßlau
- Gröger, J. et al. (2021): Weiterentwicklung des Umweltzeichens Blauer Engel, Rahmenvorhaben 2015–2019. Abschlussbericht. Umweltbundesamt, Dessau-Roßlau
- Gröger, J. et al. (2023): Weiterentwicklung des Umweltzeichens Blauer Engel, Rahmenvorhaben 2018–2021. Umweltbundesamt, Dessau-Roßlau
- Grythberg, E. (2024): Sustainability Outlook 2024. Spotify, <https://hrblog.spotify.com/2024/02/27/sustainability-outlook-2024> (27.5.2025)
- Guldner, A. (2024): Assessing the Resource and Energy Efficiency of Software and Artificial Intelligence Systems. Dissertation, Trier
- Hahn, S. (2022): Grünes Programmieren: C und Rust energieeffizient, Python und Perl Schlusslicht. heise online, <https://www.heise.de/news/Gruenes-Programmieren->

- C-und-Rust-energieeffizient-Python-und-Perl-Schlusslicht-7259319.html (17.6.2025)
- Hilty, L. et al. (2009): The Role of ICT in Energy Consumption and Energy Efficiency. ICT-ENSURE: European ICT Environmental Sustainability Research, St. Gallen
 - Hoffmann, D. (2024): Green Coding bedeutet Transformation zur Nachhaltigkeit. Ein strategischer Imperativ für die Zukunftsfähigkeit von Unternehmen. Datacenter Insider, <https://www.datacenter-insider.de/green-coding-bedeutet-transformation-zur-nachhaltigkeit-a-4138db6c9b131a69b-8c951082f76efa7/> (27.5.2025)
 - IBM (o. J.): The State of Sustainability Readiness Report 2024. <https://www.ibm.com/think/reports/sustainability-readiness> (27.5.2025)
 - IEA (2025): Energy and AI. World Energy Outlook Special Report. International Energy Agency, o. O.
 - IONOS Redaktion (2023): Green Coding im Überblick. <https://www.ionos.de/digitalguide/websites/web-entwicklung/green-coding/> (17.3.2025)
 - ITU (2024): Schätzung zur Anzahl der Internetnutzer weltweit für die Jahre 2005 bis 2024 (in Millionen). Statista GmbH, <https://de.statista.com/statistik/daten/studie/805920/umfrage/anzahl-der-internetnutzer-weltweit/> (27.5.2025)
 - Janson, M. (2022): Der CO₂-Fußabdruck unseres digitalen Lebens. Statista, <https://de.statista.com/infografik/27216/co2-emissionen-durch-informationstechnik-in-deutschland-pro-kopf/> (17.3.2025)
 - Junger, D. et al. (2023): Potentials of Green Coding – Findings and Recommendations for Industry, Education and Science. 11. Workshop Umweltinformatik zwischen Nachhaltigkeit und Wandel (UINW), Lecture Notes in Informatics (LNI). Gesellschaft für Informatik, Bonn, https://doi.org/10.18420/inf2023_137##
 - Keilhofer, H.-P. (2022): Energieverbrauchsmessung mittels jPowerMonitor Java Agent. Banking Vision, <https://banking.vision/energieverbrauchsmessung-jpowermonitor-java-agent/> (27.5.2025)
 - Kern, E. et al. (2013): Integrating Aspects of Carbon Footprints and Continuous Energy Efficiency Measurements into Green and Sustainable Software Engineering. In: Environmental Informatics and Renewable Energies, S. 300–308
 - Kern, E. et al. (2018): Sustainable software products—Towards assessment criteria for resource and energy efficiency. In: Future Generation Computer Systems 86, S. 199–210, <https://doi.org/10.1016/j.future.2018.02.044##>
 - Kolbe, L. M.; Zarnekow, R. (2013): Green IT. Erkenntnisse und Best Practices aus Fallstudien. Berlin
 - Le Goaër, O. (2023): Decarbonizing Software with Free and Open Source Software: The ecoCode Project. Research Projects Exhibition Papers Presented at the 35th International Conference on Advanced Information Systems Engineering (CAiSE 2023), Juni 2023. Saragossa
 - Lorenz, T. (o. J.): Die Bedeutung von Green Coding in der Softwareentwicklung. ComputerWeekly, <https://www.computerweekly.com/de/meinung/Die-Bedeutung-von-Green-Coding-in-der-Softwareentwicklung> (27.5.2025)
 - Maurer, J. (2015): Siemens, Pleiten und Green IT: Die Geschichte von Fujitsu. Computerwoche, <https://www.computerwoche.de/article/2623781/die-geschichte-von-fujitsu.html> (17.6.2025)
 - Microsoft (2024): How can we advance sustainability? 2024 Environmental Sustainability Report. <https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/msc/documents/presentations/CSR/Microsoft-2024-Environmental-Sustainability-Report.pdf>
 - Naumann, S. et al. (2011): The GREENSOFT Model: A reference model for green and sustainable software and its engineering. In: Sustainable Computing: Informatics and Systems 1(4), S. 294–304, <https://doi.org/10.1016/j.suscom.2011.06.004##>
 - Pandey, S. (2024): Code Reusability In Software Development. Browserstack, <https://www.browserstack.com/guide/importance-of-code-reusability> (27.5.2025)
 - Pereira, R. et al. (2017): Energy efficiency across programming languages: how do energy, time, and memory relate? In: Association for Computing Machinery (Hg.): Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering. SPLASH ,17: Conference on Systems, Programming, Languages, and Applications: Software for Humanity. <https://greenlab.di.uminho.pt/wp-content/uploads/2017/09/paperSLE.pdf> (17.3.2025)
 - Pereira, R. et al. (2021): Ranking programming languages by energy efficiency. In: Science of Computer Programm-

- ing 205, Art.10 102609, <https://doi.org/10.1016/j.sci-co.2021.102609##>
- Radovanovic, A. et al. (2023): Carbon-Aware Computing for Datacenters. In: IEEE Transactions on Power Systems 38(2), S. 1270–1280, <https://doi.org/10.1109/TPWRS.2022.3173250##>
 - Rohde, F. et al. (2021): Nachhaltigkeitskriterien für künstliche Intelligenz. Entwicklung eines Kriterien- und Indikatorensets für die Nachhaltigkeitsbewertung von KI-Systemen entlang des Lebenszyklus. Berlin
 - Rüdiger, A. (2023): Der Status Quo bei und Technologien für Green Coding. Grüne Software ist noch Mangelware. Datacenter Insider, <https://www.datacenter-insider.de/der-status-quo-bei-und-technologien-fuer-green-coding-a-37a47e852db0884255841ab88be7407e/?cft=rdt> (17.6.2025)
 - Ruiz De Villa Suárez, G. et al. (2021): GreenCoding. GFT Technologies SE, Stuttgart
 - Spotify (2024): Equity & Impact Report 2023. Stockholm
 - Spotify (2025): Equity and Impact Report 2024. Stockholm
 - Stack overflow (2022): Developer Survey 2022. <https://survey.stackoverflow.co/2022/#most-popular-technologies-language-prof> (27.5.2025)
 - Stobbe, L. et al. (2009): Abschätzung des Energiebedarfs der weiteren Entwicklung der Informationsgesellschaft. Fraunhofer-Institut für Zuverlässigkeit und Mikrointegration; Fraunhofer-Institut für System- und Innovationsforschung, Berlin
 - The Shift Project (2019): Lean ICT. Towards digital sobriety (Autor/innen: Ferreboeuf, H. et al.). O. O.
 - Vartziotis, T. et al. (2024): Learn to Code Sustainably: An Empirical Study on LLM-based Green Code Generation. In: 2024 IEEE/ACM International Workshop on Large Language Models for Code (LLM4Code), S. 30–37
 - Verdecchia, R. et al. (2023): A systematic review of Green AI. In: WIREs Data Mining and Knowledge Discovery 13(4), Art. e1507, <https://doi.org/10.1002/widm.1507##>
 - Warncke, F.; Mathew, L. (2022): Green Software Patterns. Green Software Foundation, <https://patterns.greensoftware.foundation/> (17.6.2025)
 - Wintergerst, R. (2024): Künstliche Intelligenz in Deutschland. Perspektiven aus Bevölkerung & Unternehmen. Bitkom e. V., o. O.
 - Wirtschaft Digital BW (2024): Green Coding. Klimaschonende Software. <https://www.wirtschaft-digital-bw.de/aktuelles/thema-des-monats/green-coding> (17.3.2025)
 - Wohlgemuth, V. (o. J.): Potentials of Green Coding. Hochschule für Technik und Wirtschaft Berlin, <https://wohlgemuth.f2.htw-berlin.de/forschung/potentials-of-green-coding/> (29.4.2025)
 - Yagmur, U. (2022): C, C++, Rust, Python, and Carbon (When to use Which?). Medium, <https://medium.com/codex/c-c-rust-python-and-carbon-when-to-use-which-2912a88f205b> (17.6.2025)
 - Zaninotto, F. (2022): GreenFrame Is Open-Source. Marmelab, <https://marmelab.com/blog/2022/11/09/greenframe-open-source.html> (17.6.2025)
 - Zinoviev, D. (2021): Resourceful Code Reuse. Raleigh

Herausgeber

Büro für Technikfolgen-Abschätzung
beim Deutschen Bundestag (TAB)

Bildnachweise

AndreyPopov/iStock (S. 1); quantic69/iStock (S. 2);
Weedesign/iStock (S. 4); NicoElNino/iStock (S. 6)

ISSN: 2629-2874

DOI: 10.5445/IR/1000183038

Horizon SCANNING

Das Horizon-Scanning ist Teil der Foresight-Aktivitäten des
TAB und wird vom Institut für Innovation und Technik (iit) in
der VDI/VDE Innovation + Technik GmbH durchgeführt.
www.tab-beim-bundestag.de/horizon-scanning