



Ausgabe H1/2025

Nachhaltigkeit in der IT

Brauchen wir wirklich Green IT?

Seite 4

Nachhaltigkeit vs. Fortschritt:
Ist KI ein Umweltproblem?

Seite 10

Sind IoT und Nachhaltigkeit im
Widerspruch oder eine Chance?

Seite 14

Bild: Lebenshilfe Niederösterreich

Analysis. Development. Test. Management. Better Results.

Lesen Sie in dieser Ausgabe:

Editorial.....3

Brauchen wir wirklich Green IT?.....4

Andreas Steiner

Ressourcenbewusster Umgang in der IT.....8

Markus Schwabeneder

Nachhaltigkeit vs. Fortschritt: Ist KI ein Umweltproblem?10

Martin Brandhuber

Sind IoT und Nachhaltigkeit im Widerspruch oder eine Chance?.....14

Martin Wildbacher

Nachhaltige Beziehungen bei SEQIS.....17

Susanne Greber

Kampfansage an technische Schulden – Ein Managementproblem mit technischen Folgen.....18

Alexander Weichselberger



Gemeinsam für mehr Nachhaltigkeit

Ihre Zufriedenheit ist uns wichtig, und wir schätzen Ihr Interesse an unseren Artikeln. Zugleich legen wir großen Wert auf den Schutz unserer Umwelt. Bitte teilen Sie uns mit, auf welche Weise Sie künftig von uns informiert werden möchten: postalisch und/oder elektronisch. Nutzen Sie dazu den QR-Code, um uns Ihre Präferenzen mitzuteilen. Sie können Ihre Wahl jederzeit ändern.

Über SEQIS QualityNews:

Dieses Magazin richtet sich an Gleichgesinnte aus den Bereichen IT Analyse, Development, Softwaretest und Projektmanagement im IT Umfeld. Die SEQIS Experten berichten über ihre Erfahrungen zu aktuellen Themen in der Branche. Die Leser des Magazins gestalten die Ausgaben mit: Schreiben Sie uns Ihre Meinung im SEQIS Blog (www.SEQIS.com/de/blog-index) oder als Leserbrief.

Wenn Sie dieses Magazin abbestellen möchten, senden Sie bitte ein Mail an marketing@SEQIS.com.

Impressum:

Information und Offenlegung gem. §5 E-Commerce-Gesetz und §25 Mediengesetz

Herausgeber: SEQIS GmbH,
Wienerbergstraße 3-5, A.1100 Wien
info@SEQIS.com, www.SEQIS.com

Gericht: Handelsgericht Wien
Firmenbuchnummer: 204918a
Umsatzsteuer-ID: ATU51140607
Geschäftsführung: Mag. (FH) Alexander Vukovic, Mag. (FH) Alexander Weichselberger, DI Reinhard Salomon

Druck: druck.at Druck- und Handelsgesellschaft mbH, 2544 Leobersdorf
Erscheinungsweise: 2x pro Jahr
Für die verwendeten Bilder und Grafiken liegen die Rechte für die Nutzung und Veröffentlichung in dieser Ausgabe vor.
Die veröffentlichten Beiträge, Bilder und Grafiken sind urheberrechtlich geschützt.
(Kunstwerke: Lebenshilfe Baden und Mödling, Fotos: Shutterstock, Pixabay, Pexels, Adobe Stock, unDraw).

Sämtliche in diesem Magazin zur Verfügung gestellten Informationen und Erklärungen geben die Meinung des jeweiligen Autors wieder und sind unverbindlich. Irrtümer oder Druckfehler sind vorbehalten. Hinweis im Sinne des Gleichbehandlungsgesetzes: Aus Gründen der leichteren Lesbarkeit wird die geschlechtsspezifische Differenzierung nicht durchgehend berücksichtigt. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung für beide Geschlechter.



DI Reinhard Salomon

Mag. (FH) Alexander Vukovic

Mag. (FH) Alexander Weichselberger

QualityNews

Sehr geehrte Leserin,
sehr geehrter Leser,

wir freuen uns, Ihnen die Ausgabe
für das erste Halbjahr 2025 zu
präsentieren.

Vielen Dank für das positive
Feedback zu unserer letzten
Ausgabe zu dem Themenbereich
„IT-Modernisierung“. Wir hoffen, wir
konnten Ihnen damit interessanten
Content zur Verfügung stellen und Sie
stellenweise auch gut unterhalten.
Über weitere Anregungen,
Themenwünsche und Feedback
Ihrerseits freuen wir uns.

Auch in dieser Ausgabe finden Sie
neben den branchenbezogenen
Artikeln auch nicht-technische
Bereiche:

Im Heft finden Sie einige Kunstwerke
der Lebenshilfe Niederösterreich der
Werkstätten Baden und Mödling.
Seit vielen Jahren zeigen die Klienten
aus der Werkstatt Baden und
Mödling, dass auch sie für ihre Sache
„brennen“. So wie unsere ExpertInnen
auch... .

In dieser Ausgabe dreht sich alles
um das Thema „Nachhaltigkeit
in der IT“, wobei wir aufgrund
der Omnipräsenz von Artificial
Intelligence einen entsprechenden
Schwerpunkt auf dieses Thema
gesetzt haben. Wie immer bieten
wir eine Auswahl von Fachartikeln
mit diesem Schwerpunkt, die wir
aus unserem blog.seqis.com für Sie
zusammengestellt haben.

Auf den folgenden Seiten geben Ihnen
unsere Experten einen Einblick in die
vielseitigen Aspekte zum Thema.

Wir wünschen Ihnen viel
Lesevergnügen mit der aktuellen
Ausgabe der SEQIS QualityNews!

Ihre SEQIS Geschäftsleitung

Brauchen wir wirklich Green IT?

von Andreas Steiner

Die aktuelle Lage der Klimakatastrophe und ihre Entwicklung

Um zu beurteilen, wie wichtig Nachhaltigkeit in der IT ist, zuerst zur aktuellen Lage: Fast jedes Jahr ein neues Jahrhunderthochwasser (August 2002 in Teilen Deutschlands, Tschechien und Österreich - Juni 2013 in Mitteleuropa, vor allem Bayern - Juli 2021 Ahrtal in Deutschland - Dezember 2023 Niedersachsen, Sachsen-Anhalt Thüringen und Nordrhein-Westfalen, Juni 2024 Teile Süddeutschlands)^[1], jedes Jahr ein neuer Hitzerekord (die letzten 10 Jahre sind die Top 10 der heißesten Jahre weltweit seit Beginn der Aufzeichnungen)^[2] und Dürreperioden, die Waldbrände begünstigen^[3] und sogar zu Trinkwasserrationierung führen. Der Amazonas Regenwald ist bereits ein Emissar von Treibhausgasen anstatt ein Speicher, wie Geo.de 2021 berichtet:

„Der brasilianische Teil des Amazonas-Gebiets hat zwischen 2010 und 2019 rund 18 Prozent mehr CO2 ausgestoßen als im selben Gebiet vom Regenwald gespeichert wurde.“^[4] (Thome, M. (2021, May 3).

Amazonas-Gebiet stößt mehr CO2 aus, als es bindet (GEO. Retrieved February 26, 2025).^[4]

Im Januar lag die globale Temperatur 1,75 Grad über dem vorindustriellen Niveau 5. Dies ist auch der Grafik zu entnehmen, in der man auch sieht, dass die Jahre 2023 und 2024 einen beachtlichen Sprung der globalen Temperatur aufweisen.

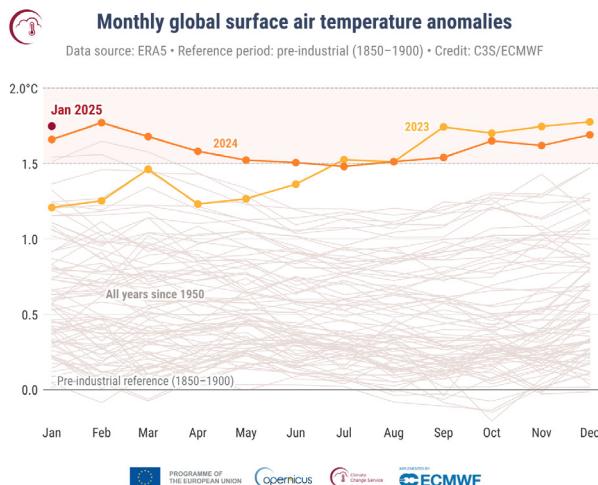


Abbildung 1: ECMWF Copernicus Climate Change Service (Quelle: <https://climate.copernicus.eu/copernicus-january-2025-was-warmest-record-globally-despite-emerging-la-nina>)

Das Pariser Klimaabkommen hat sich zum Ziel gesetzt, die globale Erderwärmung auf 1,5°C zu beschränken. Häufig ist zu lesen, dass sich diese Temperatur auf das Mittel von 1-2 Jahrzehnten bezieht. Das würde bedeuten, wir können erst im Nachhinein feststellen, ob wir das Ziel verfehlt haben. Unabhängig davon

sehen wir anhand der Grafik des 6. Weltklimaratsberichts und des aktuellen politischen Kurses, dass zu wenig getan wird und dieses Ziel jetzt schon als gescheitert angesehen werden kann. 2024 war sogar der Verbrauch von Öl, Gas und Kohle auf einem Allzeithoch.

Projected global GHG emissions from NDCs announced prior to COP26 would make it *likely* that warming will exceed 1.5°C and also make it harder after 2030 to limit warming to below 2°C.

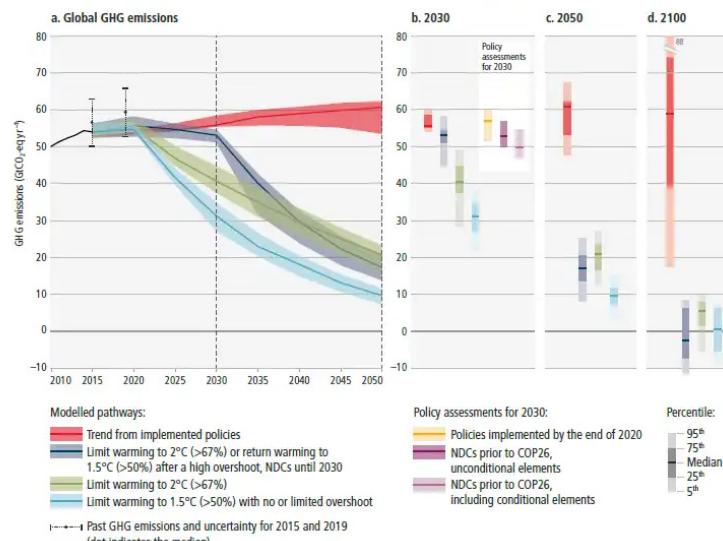


Abbildung 2: (Quelle: Kriegler, E., et al. "Data for Figure SPM.4 - Summary for Policymakers of the Working Group III Contribution to the IPCC Sixth Assessment Report." MetadataWorks, <https://doi.org/10.48490/ys3e-mq98>)



Bereits jetzt ist absehbar, dass kein aktuell lebender Mensch mehr einen Rückgang der Erwärmung erleben wird. Die Klimakatastrophe kann nur noch gebremst, aber zu unseren Lebzeiten nicht rückgängig gemacht werden. Also, es ist höchste Zeit zu handeln, denn das Fenster, etwas zu tun, schließt sich immer weiter.

Die Hauptverursacher, Potentiale und die Rolle der IT

Die größten Faktoren für das Steigen der Emissionen sind vor allem Energie-, Industrie- und Bausektor mit ~79%. Danach kommt die Land- und Forstwirtschaft sowie andere Landnutzung Sektoren mit insgesamt 22%.^[9]

Die EU Kommission^[10] nennt konkreter die folgenden Gründe für das weitere Anwachsen der Emissionen:

- Verbrennung von Kohle, Erdöl und Erdgas
- Abholzung von Wäldern (Entwaldung)
- Intensivierung der Viehzucht
- Stickstoffhaltige Dünger

Der Informations- und Kommunikationssektor (IKT-Sektor) ist dabei gerade mal für 2% der weltweiten CO2 Emissionen verantwortlich. Sieht man sich den Stromverbrauch an, sind es 5–9%.^[11] Das sind durchaus beachtliche Zahlen, doch im Vergleich zu den oben genannten Faktoren eher gering. Würde man die Energiegewinnung vollständig auf erneuerbare Energien umstellen, würde der Anteil an CO2 Emissionen des IKT-Sektors auch beachtlich schrumpfen. Jedoch ist die Belastung von Klima und Umwelt nicht ausschließlich auf CO2-Emissionen, beschränkt und die vollständige Umstellung auf erneuerbare Energien würde auch noch einige Zeit und mehr politischen Willen benötigen.

Im sechsten Sachstandbericht der Arbeitsgruppe 3 zur Minderung des Klimawandels habe ich folgende Aussage gefunden:

„In Entwicklungsländern wurde Innovation aufgrund von weniger förderlichen Rahmenbedingungen verzögert. Digitalisierung kann Emissionssenkungen ermöglichen, kann aber auch negative Nebeneffekte haben, wenn sie nicht angemessen gesteuert wird.“^[12]

Als Beispiel: Der Einsatz von KI und Robotern kann Produktionsprozesse weiter optimieren, indem der Materialverbrauch reduziert und die Energieeffizienz gesteigert wird. Gleichzeitig könnte dadurch die Produktion gesteigert und so die Auswirkungen einer geringeren Belastung für die Umwelt verloren gehen. (Sollten Sie sich für KI und Nachhaltigkeit interessieren, finden Sie auch einen eigenen Artikel dazu in unserer Quality News: „Nachhaltigkeit vs. Fortschritt: Ist KI ein Umweltproblem?“).

Das bedeutet, dass sich eine Effizienzsteigerung nicht auf eine erhöhte Produktion, sondern nur auf reduzierte Kosten auswirken sollte. Außerdem ist es wichtig, den Energieverbrauch so weit wie möglich zu reduzieren. Selbst bei ausschließlicher Nutzung erneuerbarer Energien bleibt ein verantwortungsvoller Umgang mit Rohstoffen und unserer Umwelt unerlässlich – denn die Naturgesetze sind unveränderlich.

Wie können wir Green IT und Nachhaltigkeit im Unternehmen umsetzen und wie kann davon profitiert werden?

Natürlich ist individuelles Verantwortungsbewusstsein zu befürworten. Zum Beispiel ist es sinnvoll, nicht benötigte Arbeitsgeräte (z.B. über das Wochenende) abzuschalten, anstatt nur in den Standby-Modus zu versetzen. Jedoch würde ich an dieser Stelle noch nicht von Green IT sprechen, abgesehen davon, dass es nur sehr marginale Verbesserung bringt, vermutlich nicht von allen umgesetzt wird. Hier meine Vorschläge für konkrete Handlungsempfehlungen:

- Home Office: Durch Home Office lassen sich Arbeitswege vermei-

den und so den CO2 Ausstoß für Arbeitswege stark reduzieren. Dadurch, dass das Büro nicht mehr voll besetzt ist, können geteilte Arbeitsplätze eingerichtet und somit Kosten gespart werden. Richtig umgesetzt, können sie für Arbeitnehmer ein attraktiveres Unternehmen werden und bieten einen Vorteil gegenüber der Konkurrenz.

- Videokonferenzen: Manchmal lässt es sich nicht vermeiden, zu einem Kunden zu fahren. Videokonferenzen können Reisekosten sparen, sie sind einfacher zu planen und kurzfristiger umzusetzen und das weltweit.
- Auf erneuerbare Energien setzen: Erneuerbare Energie ist die am nachhaltigsten produzierte Energie. Darum ist es vorteilhaft, Strom von Anbietern zu beziehen, die ausschließlich erneuerbare Energien beziehen. So kann auf Dauer Geld gespart werden und unterstützt den Ausbau dieses Sektors.
- Energieeffiziente Hardware verwenden: Sei es bei Servern oder anderer IT-Ausrüstung. Es ist sinnvoll, auf energieeffiziente Hardware zu achten. Diese verbrauchen weniger Strom, was zu einer reduzierten Wärmeentwicklung und damit zu einer geringeren Kühlungsanforderung führt.
- Cloud Computing und Virtualisierung: Vor der Beschaffung von eigenen Servern kann es hilfreich sein, zu beurteilen, ob eine Cloud Computing Lösung vorteilhafter ist. Diese verringern den Bedarf physischer Hardware und auch den anfallenden Elektroschrott. Außerdem können Cloud-Anbieter energieeffizientere Lösungen anbieten und erleichtern die Skalierbarkeit, wenn Ihr Unternehmen wächst. Durch

Virtualisierung von Workspaces, vielleicht auch durch Auslagerung an Cloud-Providern, lässt sich zusätzliche Hardware einsparen, da mehrere Virtual Machines auf einem Server betrieben werden können. Durch die Auslagerung der Virtualisierung können auch andere diese Systeme nutzen, wenn sie nicht benötigt werden.

Es gibt noch zahlreiche andere Möglichkeiten für mehr Nachhaltigkeit im Unternehmen, wie nachhaltige Beschaffung, Optimierung des Produktlebenszyklus auf Langlebigkeit und Recycling. Sogar nachhaltige Softwareentwicklung wird immer wichtiger.

Für die Themen ressourceneffizientes Coding, nachhaltiges Testdatenmanagement und vieles mehr lesen Sie gerne die weiteren interessanten Artikel in unserer aktuellen Ausgabe der Quality News: Nachhaltigkeit in der IT!

Quellen und weiterführende Informationen:

[¹¹] Wangerin, Claudia. „Flutschäden: Das nächste Jahrhunderthochwasser kommt bestimmt – was tun?“ Telepolis, 3 June 2024, <https://www.telepolis.de/features/Flutschaden-Das-naechste-Jahrhunderthochwasser-kommt-bestimmt-was-tun-9744748.html>.

[¹²] „Wärmste Jahre weltweit seit Beginn der Messung bis 2024.“ Statista, 4 February 2025, <https://de.statista.com/statistik/daten/studie/157755/umfrage/klimawandel-die-weltweit-waermersten-jahre-seit-1880/>.

[¹³] „Klimawandel hat laut Studie die Brände in Los Angeles begünstigt.“ Tagesschau, 28 Januar 2025, <https://www.tagesschau.de/wissen/klima/klimawandel-waldbraende-kalifornien-100.html>.

- [⁴] „Thome, Matthias. „Amazonas-Gebiet stößt mehr CO2 aus, als es bindet –.“ GEO, 3 May 2021, <https://www.geo.de/natur/oekologie/amazonas-gebiest-stoessst-mehr-co2-aus--als-es-bindet-30510406.html>.
- [⁵] „Jänner 2025 war der wärmste je gemessene.“ Kurier, 6 February 2025, <https://kurier.at/politik/klima/klimabericht-jaenner-monat-hitze-klimawandel-erderwaermung-temperaturen-copernicus/403007307>.
- [⁶] „Copernicus: January 2025 was the warmest on record globally, despite an emerging La Niña | Copernicus.“ Copernicus Climate Change, 6 February 2025, <https://climate.copernicus.eu/copernicus-january-2025-was-wardest-record-globally-despite-emerging-la-nina>.
- [⁷] „CO₂-Emissionen 2024: Verbrauch von Öl, Gas und Kohle so hoch wie nie zuvor.“ RND, 13 November 2024, <https://www.rnd.de/wissen/co2-emissionen-2024-verbrauch-von-oel-gas-und-kohle-so-hoch-wie-nie-zuvor-AG-WLBITZXRKPBL6TEYTHFZXQ.html>.
- [⁸] Kriegler, E., et al. „Data for Figure SPM.4 - Summary for Policymakers of the Working Group III Contribution to the IPCC Sixth Assessment Report.“ MetadataWorks, <https://doi.org/10.48490/ys3e-mq98>.
- [⁹] „Klimawandel – Ursachen und Folgen.“ Welthungerhilfe, <https://www.welthungerhilfe.de/informieren/themen/klimawandel>.
- [¹⁰] „Ursachen des Klimawandels - Europäische Kommission.“ EU Climate Action, https://climate.ec.europa.eu/climate-change/causes-climate-change_de.
- [¹¹] Bundesministerium für Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie, <https://www.bmk.gv.at/themen/energie/effizienz/rechenzentren.html>
- [¹²] Deutsche IPCC-Koordinierungsstelle - de-IPCC, <https://www.de-ipcc.de>. Katharina Aganina, „Green IT: So können Sie mithelfen, Ihr Unternehmen umweltfreundlicher zu gestalten“, Ratbacher Blog, <https://www.ratbacher.at/blog/green-it-massnahmen/#:~:text=Beispiele%20Green%20IT%20in%20der,im%20Bereich%20Green%20IT%20t%C3%A4gig>.
- Amanda McGrath, Alexandra Jonker, „Was ist nachhaltige IT?“, IBM, 5. Juli 2024, <https://www.ibm.com/de-de/think/topics/sustainable-it>



Andreas Steiner ist Consultant für Softwaretest und Requirements Engineering bei SEQIS.

Die Schwerpunkte seiner Projekterfahrung liegen in den Bereichen Testautomation, Testdurchführung, Softwarequalitäts-sicherung und Requirements Engineering.

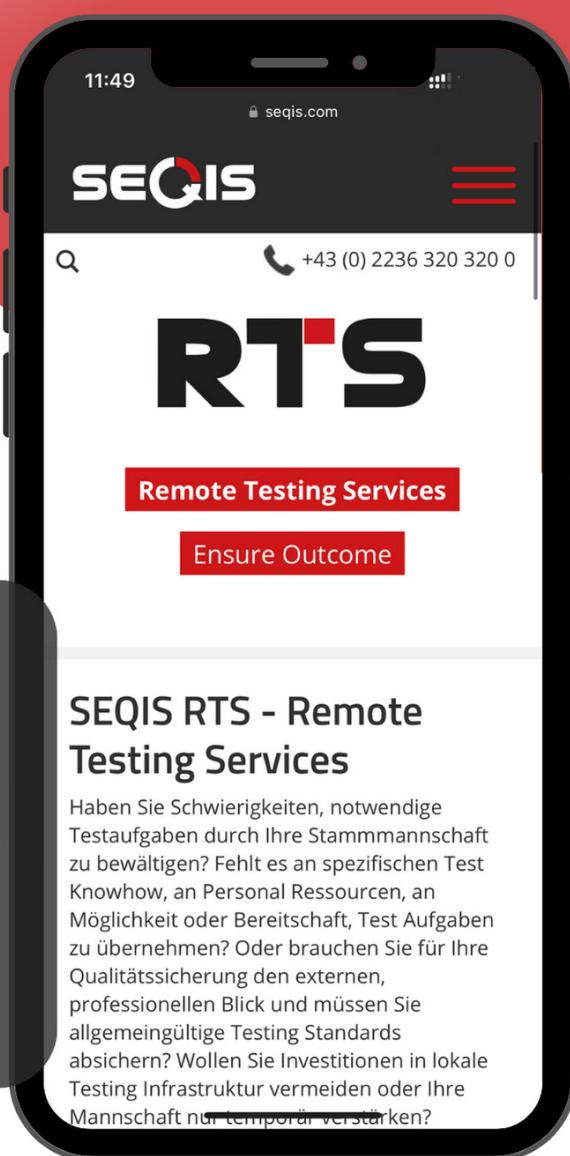
Kommunikation, Teamwork sowie vorausschauendes Arbeiten sind für ihn essenzielle Bestandteile für einen reibungslosen Projektlauf.



www.SEQIS.com

SEQIS Remote Testing Services -
Hilfe aus Österreich für Ihre spezifischen
Testaufgaben.

Kontaktieren Sie mich unter
alexander.weichselberger@SEQIS.com
und sprechen wir über Ihre
Aufgabenstellungen. Wir helfen Ihnen,
Ihr Projekt zum Erfolg zu führen.



QR Code scannen und weitere
Informationen erhalten.

Ressourcenbewusster Umgang in der IT

von Markus Schwabeneder

Steigende Energiekosten machen den effizienten, nachhaltigen Umgang mit unseren Ressourcen nicht nur zu einem ökologischen, sondern auch zu einem ökonomischen Thema, das aber bisher großteils sehr stiefmütterlich behandelt wurde und noch immer in den Kinderschuhen steckt. Eine weitere Schwierigkeit ist, dass es in dem Bereich, wie auch in vielen anderen Bereichen der IT, nur wenig allgemeingültige Paradigmen gibt, sondern viele Wege, die Vor- aber auch Nachteile haben. Daher möchte ich mit diesem Artikel vor allem Gedankenanstöße liefern. Welche Maßnahmen für verschiedene Teams, Organisationen und Unternehmen sinnvoll sind, muss man konkret und individuell bestimmen und, wie üblich, kontinuierlich verbessern und an neue Entwicklungen anpassen.

Kurzfristige und im kleinen Bereich umsetzbare Maßnahmen

Zuerst zeige ich Ideen auf, die ohne große Vorbereitungen, finanziellen Aufwand und teilweise auch vom Team oder einzelnen Teammitgliedern umgesetzt werden können.

Speicher- und/oder Rechenleistung sparernder Code

Das hört sich auf den ersten Blick nach einer Selbstverständlichkeit an, ist es aber in Wirklichkeit nicht. Ich habe es in meinem Berufsleben oft genug erlebt, dass effizienter Einsatz von Speicher und Rechenleistung zumindest in Teilbereichen keine wichtige Nicht-Funktionale-Anforderung war. Oft gibt es nur die Anforderung, dass die Benutzeroberfläche schnell reagieren muss, Back-End-Tasks werden aber oft nicht in Hinblick auf Effizienz programmiert, wenn ihre Durchlaufzeit keine Wichtigkeit hat. Man sollte aber jetzt auch nicht in

einen "Optimierungswahn" verfallen. Lesbarer und leicht verständlicher Code ist in den meisten Fällen besser als hoch optimierter Code, der z.B. exzessiv Objekte wiederverwendet, um Aufwand durch Initialisierungen und Garbage Collection zu reduzieren. Es besteht die Gefahr, dass kleine Ressourceneinsparungen hier durch erhöhten Wartungsaufwand aufgefressen werden.

Modularer Code

Per se spart modulare Programmierung jetzt keine Ressourcen, sie ermöglicht es aber erst, exakt an die Anforderungen skalierbare Software zu erstellen.

Wahl effizienter Programmiersprachen

Programmiersprachen haben stark unterschiedliche Stärken und Schwächen. Manche Aufgaben können ausgezeichnet von leichtgewichtigen Skriptsprachen erledigt werden. Schwergewichtige Allzweck-Sprachen mit großem Funktions- und Leistungsumfang zu verwenden bedeutet, öfters auch mal mit Kanonen auf Spatzen zu schießen. Sicherlich haben diese Sprachen ihre Berechtigung, der Einsatz sollte aber bewusst erfolgen und nicht als einzigen Grund haben, dass "es das eh auch kann".

Wahl der passenden IDE

Hier gibt es 2 Dinge zu beachten. Erstens gibt es IDEs, die initial schon deutlich mehr Ressourcen verbrauchen als leichtgewichtige Alternativen. Hier sollte man nicht einfach die IDE wählen, die die meisten Funktionen hat, sondern die, die die benötigten bietet. Zweitens kann man mittlerweile quasi alle IDEs hochgradig erweitern und Zusatzmodule installieren. Hier gibt es die Gefahr, dass man nicht genau überlegt, was man braucht, sondern einfach aus

Bequemlichkeit viel zu viel installiert und die IDE unnötig aufbläht.

Richtiger Einsatz von KI

KI wird mittlerweile nicht nur als Hilfe bei der Programmierung, sondern auch immer mehr in der Software selbst eingesetzt. Klarerweise benötigt dieser Einsatz Ressourcen, er wird aber in vielen Fällen sinnvoll sein. Oft gibt es hier aber noch sehr viel Optimierungspotential. Mächtige LLMs liefern mittlerweile in vielen Szenarien erstaunlich gute Ergebnisse, brauchen aber sehr viel Energie (und verursachen dann auch hohe Kosten). Viele Tasks können aber auch von gut eingestellten, speziellen Modellen ähnlich gut, aber mit viel weniger Energieverbrauch und Kosten erledigt werden.

Maßnahmen, die typischerweise aufwendiger in der Umsetzung sind und über die Teamebene hinausgehen

Die folgenden Maßnahmen müssen üblicherweise firmenweit getroffen werden, benötigen höheren finanziellen Einsatz und können oft nur langfristig umgesetzt werden.

Nutzung von Software-Virtualisierung

Software-Virtualisierung kann helfen, die Software je nach Auslastung nach unten oder oben zu skalieren und damit nur so viele Ressourcen wie wirklich aktuell gebraucht werden. Auch können so Test- und Staging-Systeme einfach nach Bedarf hoch- und nieder gefahren. Modularer Software, die auf Services aufbaut, kann hier noch gezielter benötigten Teilen mehr Ressourcen zuweisen, ohne automatisch das ganze System hochskalieren zu müssen.

Nutzung von Hardware-Virtualisierung

Hardware-Virtualisierung hat den

Vorteil, dass aus einem gemeinsamen Pool an Speicher und Rechenleistung beides nach Bedarf den laufenden Systemen zugeteilt werden kann. Ohne diese Virtualisierung müsste jedem System von vornherein so viele Ressourcen zugeteilt werden, wie es maximal benötigen darf – was in Summe deutlich mehr sein kann. Besonders sinnvoll ist dies, wenn man schon a priori weiß, dass das System zu unterschiedlichen Zeiten hohe Auslastungen haben wird. Natürlich wird durch diese Virtualisierung Overhead erzeugt, daher muss dies mit dem Einsparungspotential abgeglichen werden. Sicherheitstechnisch muss man zusätzlich bedenken, dass physische Daten von mehreren Systemen auf einer Hardware landen, was eventuell nicht gewünscht ist. Sauber konfigurierte Systeme können diese Risiken aber softwaretechnisch gut mitigieren, bei extrem sensiblen Daten mag dies aber nicht reichen. Dann stellt sich auch die Frage, ob diese Virtualisierung intern abgewickelt, von einem Rechenzentrum übernommen oder überhaupt von einem Cloud-Anbieter abgewickelt werden soll.

Das größte Ressourcen-Einsparungspotential gibt es sicherlich in der Cloud. In der Cloud können Einsparungsmaßnahmen so weit gehen, dass Rechenleistung zeitlich an unterschiedlichen Orten generiert wird, wo gerade Überschuss an Energie herrscht. Nachteil ist aber, dass man kaum Einfluss darauf hat, wie der Cloud-Anbieter dies wirklich umsetzt, ob diese Einsparungen sich ökonomisch auch für den Kunden lohnen und ob andere Nebeneffekte den ökologischen Fußabdruck nicht doch noch größer machen. All dies hat man natürlich besser im Griff, wenn Rechenleistung und Speicher intern bereitgestellt werden.

Passende Hardware für Mitarbeiter

Eine sinnvolle Auswahl der Hardware kann auch Einsparungspotential bieten. ARM-Systeme sind derzeit deutlich ressourcenschonender als

x86-Systeme.

Ebenso ist der Einsatz von Echtgeräten eine Alternative zur Verwendung von besonders rechenstarker Hardware, die die Emulation stemmen können muss.

Eine weitere Möglichkeit ist die Verwendung von virtualisierten Arbeitsumgebungen. So muss die geforderte Rechenleistung nicht von der Hardware des Users generiert werden, sondern kann von einem zentralen Server bezogen werden, der dies potenziell effizienter machen kann.

Homeoffice im Hinblick auf Ressourcenschonung

Homeoffice spart zwar Ressourcen, die bei der Anfahrt zum Arbeitsort verbraucht werden, verbraucht aber an anderer Stelle auch Ressourcen. Als einfachstes Beispiel gebe ich hier ein größeres Meeting an, bei dem einmal alle Teilnehmer ihren eigenen Computer und Bildschirm nutzen und über das Netzwerk kommunizieren müssen, während, wenn alle an einem Ort wären, keine technischen Geräte in Betrieb sein müssten. Auch andere Aspekte könnten ein großes Büro effizienter bedienen als viele einzelne verteilte Arbeitsplätze. Ein öffentlich gut erreichbares, energieeffizientes Gebäude, Mitarbeiter-Benefits wie gestützte E-Mobility und sinnvoller Einsatz von Homeof-

fice tragen sicherlich auch zu ökologischer Effizienz bei.

Abschließende Worte

Der ressourcenbewusste Umgang in der IT ist ein komplexes und vielschichtiges Feld, das von kurzfristig umsetzbaren Optimierungen im Code bis hin zu langfristigen strategischen Entscheidungen auf Unternehmensebene reicht. Dieser Artikel ist als Impulsgeber zu verstehen, die hier vorgestellten Ideen müssen in Ihrem eigenen Kontext geprüft, evaluiert und angepasst werden. Es ist klar, dass dies eine teilweise herausfordernde, aber auch lohnende Aufgabe ist. Gerne können wir Sie hierbei auch unterstützen und einen Teil des Weges gemeinsam gehen. Auch über Feedback und Erfahrungsaustausch würde ich mich freuen.



Kontaktdaten
Markus Schwabeneder



Markus Schwabeneder ist Principal Consultant und Chief Architect bei razzfazz.io.

Er begann seine Karriere in der Softwarebranche als Consultant für komplexe mathematische Themen und Optimierungsaufgaben. Seitdem ist er in den Bereichen Softwareentwicklung, Anforderungsanalyse, Testing und Softwarearchitektur tätig. Seine Fähigkeiten setzt er in unterschiedlichen Branchen ein.

Besonders begeistert er sich für das Erarbeiten von komplexen Anforderungen und spezifischen Vorgehensweisen.

Nachhaltigkeit vs. Fortschritt: Ist KI ein Umweltproblem?

von Martin Brandhuber

Künstliche Intelligenz (KI) hat sich in den letzten Jahren rasant entwickelt und durchdringt zunehmend verschiedene Bereiche unseres Lebens. Spätestens seit der Veröffentlichung des Textroboters ChatGPT von OpenAI ist KI in aller Munde. Ob in der Medizin, der Industrie oder dem Alltagsleben – KI-Technologien bieten enorme Chancen für Effizienzsteigerung und Innovation. Doch die zunehmende Verbreitung dieser Technologien hat auch eine entscheidende Kehrseite: Sie kann erhebliche Umweltbelastungen verursachen. Öffnen wir mit KI somit eine Büchse der Pandora?

Entwicklung der künstlichen Intelligenz

Die Wurzeln der künstlichen Intelligenz reichen bis in die 1950er Jahre zurück, als Wissenschaftler begannen, Algorithmen zu entwickeln, die menschliches Lernen und Problemlösungsverhalten nachahmen. In den letzten Jahrzehnten haben Fortschritte in der Rechenleistung und die Verfügbarkeit großer Datenmengen (Big Data) das maschinelle Lernen und insbesondere tiefe neuronale Netze (Deep Learning) revolutioniert. Heute sind KI-Modelle in der Lage, komplexe Aufgaben zu bewältigen, von der Bilderkennung bis hin zur Vorhersage von Krankheitsverläufen. Doch die leistungsfähigen Algorithmen erfordern immense Rechenkapazitäten, was zu steigenden Energieverbräuchen führt.

Zusätzlich zu diesen technologischen Fortschritten hat die Kommerzialisierung von KI-Technologien eine exponentielle Nachfrage nach Rechenleistung geschaffen. Unternehmen und Forschungseinrichtungen weltweit investieren in immer leistungsstarkere Hardware, um wettbewerbsfähig

zu bleiben. Dies hat zur Entstehung riesiger Rechenzentren geführt, die eine ununterbrochene Stromversorgung und Kühlung benötigen.

Warum ist KI deswegen ein Umweltproblem?

„Nachhaltigkeit“ ist – leider – zu einem Modewort verkommen. Überall findet man Nachhaltigkeit, auf Milchpackungen genauso wie auf den Hinweisen von Spielsachen, auf Werbeseiten von Fluglinien und sogar in der Softwareentwicklung. Zur Definition von Nachhaltigkeit fragen wir Wikipedia:

„Nachhaltigkeit ist ein Handlungsprinzip bei der Nutzung von Ressourcen. Hierbei soll eine dauerhafte Bedürfnisbefriedigung gewährleistet werden, indem die natürliche Regenerationsfähigkeit der beteiligten Systeme bewahrt wird, vor allem von Lebewesen und Ökosystemen. Das Handlungsprinzip der Nachhaltigkeit ist ein Zielkonflikt, ein veränderlicher bestmöglich Zustand, in dem es nicht möglich ist, die Ziel-Eigenschaft Ressourcennutzung zu verbessern, ohne zugleich die andere Eigenschaft des Ressourcen-Erhalts verschlechtern zu müssen. Im Allgemeinen wird Nachhaltigkeit heute als Form der Ressourcennutzung verstanden, die nach dem Drei-Säulen-Modell auf dem gleichzeitigen und gleichberechtigten Umsetzen von Umweltschutz, langfristigem Wirtschaften und einem fairen Miteinander beruht, damit auch zukünftige Generationen gut leben können.“^[1]

Kurz gesagt, Nachhaltigkeit soll sicherstellen, dass das ökologische Gleichgewicht erhalten bleibt. Eine steigende Nutzung von Ressourcen geht also gleichzeitig mit einer Benachteiligung des Gleichgewichts einher.

Nun fragt man sich – Rechenzentren hat es doch schon früher gegeben, warum soll das auf einmal so schädlich für die Umwelt sein?

KI ist kein „fertiges Programm“, das einmalig entwickelt wird. Sie soll sich weiterentwickeln, soll lernen, soll trainiert werden – selbstständig, durch die Erfahrungen mit InternetnutzerInnen, oder durch die gezielte Versorgung mit Testdaten. Und das benötigt Rechenleistung und somit Energie. Auch unabhängig davon verbraucht eine Anfrage etwa bei ChatGPT grob zehnmal so viel Energie wie eine Google-Suche.

„Rechenzentren verbrauchen heute vier bis fünf Prozent des weltweiten Energieverbrauchs“, sagte der Geschäftsführer des Hasso-Plattner-Instituts (HPI) in Potsdam und Leiter des Fachgebiets Künstliche Intelligenz und Nachhaltigkeit, Ralf Herbrich, der Nachrichtenagentur dpa. „Es gibt Schätzungen, dass der Verbrauch in den nächsten Jahren auf 30 Prozent ansteigen wird.“

Weitere Beiträge von dem Experten Martin Brandhuber finden Sie auf unserer Webseite.



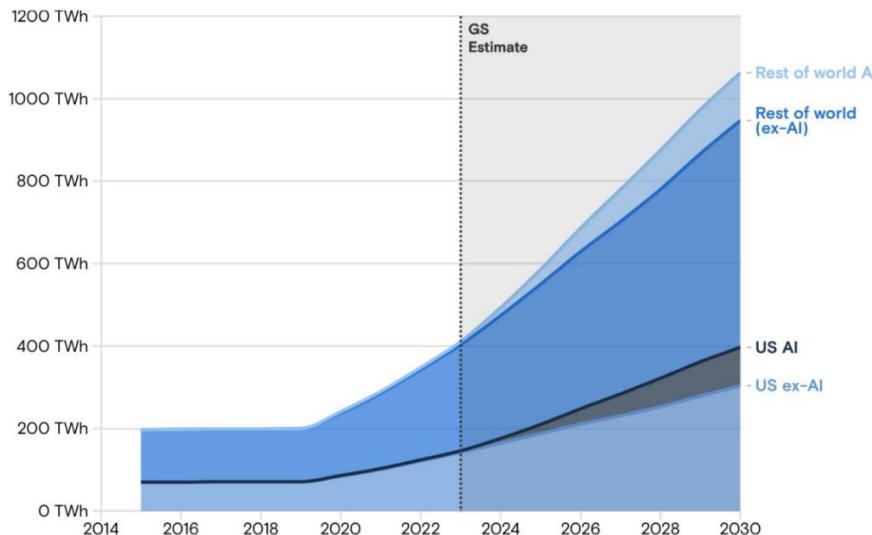


Abbildung 1: (Quelle: Masanet et al. (2020), Cisco, IEA, Goldman Sachs Research)

Die Umweltbelastung durch KI-Technologien lässt sich auf mehrere Faktoren zurückführen:

1. Energieverbrauch von Rechenzentren

Moderne KI-Anwendungen basieren auf rechenintensiven Prozessen, die in leistungsfähigen Rechenzentren durchgeführt werden. Diese Rechenzentren benötigen enorme Mengen an Energie, um sowohl die Server zu betreiben als auch die notwendige Kühlung zu gewährleisten. Das Training des „Generative Pre-trained Transformer 3“ (GPT-3) Sprachmodells von OpenAI, welches Deep Learning verwendet, benötigte beispielsweise 1.287 Megawattstunden Strom - das entspricht dem Verbrauch einer Kleinstadt über mehrere Monate (Beispiel: Klein-Neusiedl; ca. 1.000 Einwohner). Die dabei entstandenen CO₂-Emissionen entsprachen etwa 700 Hin- und Rückflügen von Wien nach Dubai. Der Nachfolger, GPT-4, benötigte 6 Monate Training und verbrauchte dabei sogar 7.200 Megawattstunden - das 5,5-fache seines Vorgängers! Google möchte bis zum Jahr 2030 unterm Strich klimaneutral sein. Aus diesem Grund sollen KI-Rechenzentren künftig mit kleinen modularen Atomkraftwerken betrieben werden, bis 2035 soll die jährliche Leistung 500 Megawatt

erreichen. Nachhaltigkeit durch Atomstrom?

2. Ressourcenverbrauch und CO₂-Emissionen

Die Herstellung der Hardware für KI-Systeme erfordert seltene Erden und andere Rohstoffe, deren Abbau oft mit erheblichen Umweltschäden verbunden ist. Zudem erzeugt der Betrieb von KI-Modellen erhebliche CO₂-Emissionen, insbesondere wenn der Strom aus fossilen Quellen stammt. Selbst in Regionen mit erneuerbarer Energie besteht das Problem, dass der steigende Strombedarf die Energiewende verlangsamen könnte. Eine Studie der Universität Massachusetts ergab, dass das Training eines einzigen großen KI-Modells mehr als 284 Tonnen CO₂ emittieren kann – das entspricht dem fünfmaligen CO₂-Ausstoß eines durchschnittlichen amerikanischen Autos über dessen gesamte Lebensdauer.

Ein weiteres Problem ist der Wasserverbrauch. Rechenzentren benötigen große Mengen an Wasser zur Kühlung, was in trockenen Regionen zu ernsthaften Wasserknappheiten führen kann. Ein aktuelles Thema dazu sind die Waldbrände in Südkalifornien, welche seit dem 7. Jänner 2025 wüten, wo die Wasserknapp-

heit zu erheblichen Problemen bei der Bekämpfung der Brände führt. Ironischerweise sind in Kalifornien auch große Rechenzentren angesiedelt, etwa von Google, die das Grundwasser zusätzlich belasten.

3. E-Waste und kurze Innovationszyklen

Technologische Fortschritte führen dazu, dass Hardware-Komponenten immer schneller ersetzt werden müssen. Alte Server, Prozessoren und GPUs werden entsorgt, was zu wachsendem Elektroschrott führt. Da viele dieser Materialien schwer zu recyceln sind, entstehen zusätzliche Umweltprobleme. Darüber hinaus sind die Lebenszyklen von KI-Technologien oft sehr kurz. Neue Modelle und Algorithmen erfordern oft spezialisierte Chips, die ältere Generationen schnell obsolet machen. Dies führt zu einem noch schnelleren Austausch der Hardware, was den Ressourcenverbrauch weiter erhöht.

Nutzen und Kosten von KI aus ökologischer Sicht

KI-Technologien können jedoch auch zur Lösung von Umweltproblemen beitragen. Daher stellt sich die Frage, ob der Nutzen den ökologischen Schaden überwiegt.

Nutzen von KI für die Umwelt

- **Optimierung von Energieverbrauch:** KI kann dabei helfen, Stromnetze effizienter zu steuern und den Energieverbrauch in Unternehmen und Haushalten zu reduzieren.
- **Nachhaltige Landwirtschaft:** Durch präzise Analysen können KI-Modelle Landwirten helfen, Wasser und Düngemittel effizienter einzusetzen.
- **Früherkennung von Umweltproblemen:** KI-gestützte Sensoren können Umweltveränderungen frühzeitig identifizieren und Katastrophen wie Waldbrände, Erdbeben oder Tsunamis vorhersagen.
- **Kreislaufwirtschaft:** KI kann Ma-

- terialströme optimieren und so Recyclingprozesse verbessern.
- KI im Umweltschutz:** Google selbst nutzt das Unternehmen DeepMind KI, um den Energieverbrauch seiner Rechenzentren um 40 % zu reduzieren.

Kosten und Risiken

- Steigender Energiehunger: Die Fortschritte in der KI führen dazu, dass immer leistungsfähigere Modelle trainiert werden müssen, was den Energiebedarf weiter erhöht.
- Mangelnde Transparenz: Es gibt bislang wenige Standards für die Nachhaltigkeit von KI-Anwendungen, was es schwierig macht, ihren ökologischen Einfluss zu bewerten.
- Abhängigkeit von Rohstoffen: Der steigende Bedarf an Hochleistungs-Hardware erhöht die Nachfrage nach seltenen Erden und anderen kritischen Rohstoffen.

Lösungsansätze für eine nachhaltige KI

Um das Spannungsfeld zwischen Fortschritt und Nachhaltigkeit zu entschärfen, sind verschiedene Maßnahmen notwendig:

1. Energieeffiziente KI-Modelle

Die Forschung an effizienteren Algorithmen kann dazu beitragen, den Rechenaufwand zu senken. Beispielsweise können prädiktive Modelle zur Energieeinsparung beitragen oder sparsamer programmierte neuronale Netze verwendet werden.

2. Einsatz erneuerbarer Energien

Rechenzentren können verstärkt auf erneuerbare Energien setzen, um ihren CO₂-Fußabdruck zu minimieren. Einige Tech-Unternehmen investieren bereits in nachhaltige Stromquellen, doch hier gibt es noch viel Potenzial.

3. Verlängerte Lebensdauer von Hardware

Durch Recycling und Wiederverwendung können wertvolle Rohstoffe

eingespart werden. Unternehmen sollten zudem auf langlebigere Hardware setzen und die Innovationszyklen verlängern.

4. Bewusste Nutzung von KI

Nicht jede Anwendung muss auf komplexen neuronalen Netzen basieren. Häufig reichen einfachere, weniger rechenintensive Modelle aus, um eine Aufgabe zu bewältigen. Unternehmen sollten prüfen, in welchen Fällen eine ressourcenintensive KI wirklich notwendig ist.

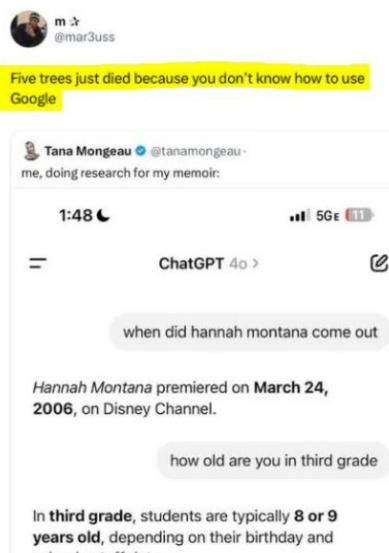


Abbildung 2: (Quelle: x.com)

Müssen wir wirklich jedes Mal eine KI befragen, wenn wir nach etwas suchen? Und auch wenn wir keine KI fragen, müssen wir wirklich eine Suchmaschine öffnen? Haben Lexika, Wörterbücher und Atlanten wirklich keine Daseinsberechtigung mehr?

5. Regulierung und Transparenz

Politische Entscheidungsträger können Nachhaltigkeitsstandards für KI entwickeln und Unternehmen dazu verpflichten, ihre Umweltauswirkungen offenzulegen. Dadurch könnten nachhaltigere Innovationen gefördert werden.

Ausblick: Die Zukunft der nachhaltigen KI

Ist KI somit gesehen ein Umweltproblem? Die Antwort lautet: ja.

Aber sie ist ein Problem, dem entgegengewirkt werden kann, und die Debatte darüber sollte intensiviert und Verbesserungen dringend erarbeitet werden. Eine nachhaltige KI ist möglich – doch sie erfordert bewusstes Handeln und eine langfristige Strategie. Nur so kann sichergestellt werden, dass der Fortschritt nicht auf Kosten der Umwelt geht, sondern im Einklang mit ihr steht. Und dazu müssen wir zuallererst unsere eigene Einstellung zu KI hinterfragen. Wollen wir ChatGPT nur benutzen, nur weil es gerade "in" ist? Müssen wir wirklich einen "smartten" Thermostat kaufen? Oder reicht es, selber die paar Schritte zum Heizkörper zu machen, und währenddessen in einem Lexikon nachzuschlagen?

Weitere spannende Blogartikel finden Sie hier:

[www.seqis.com/de/
blog-index](http://www.seqis.com/de/blog-index)



Fortschritt vs. Nachhaltigkeit: in a nutshell

Vorteile	Aufwände	Lösungen
Optimierung von Energieverbrauch	Steigender Energieaufwand	Energieeffiziente KI-Modelle
Unterstützung nachhaltiger Landwirtschaft	Mangelnde Transparenz zu ökologischem Einfluss	Einsatz erneuerbarer Energien
Früherkennung von Umweltproblemen	Rohstoffbedarf	Verlängerte Lebensdauer von Hardware
Kreislaufwirtschaft		Bewusste Nutzung von KI
KI im Umweltschutz		



Abbildungen: (Quelle: <https://undraw.co/>)

Quellen und weiterführende Informationen:

[1] <https://de.wikipedia.org>



Martin Brandhuber ist Senior Consultant und Teamlead bei SEQIS.

„Kundenzufriedenheit und Professionalität in der Planung und Durchführung des Testprozesses hat für mich höchste Priorität. Testen ist für mich ein nicht wegzudenken Bestandteil der Softwareentwicklung. Da die Anforderungen an den Vernetzungsgrad von Systemen laufend zunehmen, ist die frühe Einbindung von Testprozessen in der Softwareentwicklung unabdingbar. Nur so kann die Entwicklung kostengünstiger, qualitativ hochwertiger Software garantiert werden.“

Sind IoT und Nachhaltigkeit im Widerspruch oder eine Chance?

von Martin Wildbacher

Die rasante Entwicklung des Internet of Things (IoT) hat unsere Welt grundlegend verändert. Intelligente Sensoren, vernetzte Geräte und smarte Anwendungen optimieren Prozesse in nahezu allen Lebensbereichen.

Aktuell gibt es in Europa schätzungsweise **3,92 Milliarden IoT-Geräte** (Stand 2024). Diese Zahl verdeutlicht die wachsende Verbreitung von IoT-Technologien in verschiedenen Bereichen wie Smart Homes, Industrieautomation und Transportwesen. Prognosen zufolge wird die Anzahl der IoT-Verbindungen in Europa 2025 auf etwa 4,36 Milliarden ansteigen – weltweit auf etwa 20,1 Milliarden.^[1]

Dieses Wachstum wird durch Investitionen in Technologien wie Low-Power-Wide-Area-Netzwerke (LPWAN) und die Einführung von 5G-Netzen unterstützt, die eine kostengünstige und energieeffiziente Konnektivität ermöglichen.

Die zunehmende Verbreitung von IoT-Geräten trägt dazu bei, Prozesse zu optimieren und Kosten zu senken, was ihre Popularität in Europa weiter fördert. Es wird erwartet, dass dieser Trend in den kommenden Jahren anhält, insbesondere in Bereichen wie Energieeffizienz, Logistik und Umweltüberwachung.

Doch wie wirkt sich diese Technologie auf die Nachhaltigkeit aus? Sind IoT und Nachhaltigkeit ein Widerspruch, oder können sie gemeinsam zur Gestaltung einer umweltfreundlicheren Zukunft beitragen?

Dieser Artikel beleuchtet die Potenziale und Herausforderungen des IoT im Kontext der Nachhaltigkeit.



Abbildung 1: (Quelle: desinko - stock.adobe.com)

Chancen: IoT als Treiber für Nachhaltigkeit

IoT bietet zahlreiche Möglichkeiten, Nachhaltigkeitsziele effektiv zu unterstützen:

1. Effiziente Ressourcennutzung:

- Präzisionslandwirtschaft: IoT-Sensoren liefern in Echtzeit Daten zu Bodenfeuchtigkeit, Temperatur, Nährstoffgehalt und Wetterbedingungen. Dadurch kann die Bewässerung gezielt auf die Bedürfnisse der Pflanzen abgestimmt werden, was nicht nur Wasser spart, sondern auch die Gefahr von Bodenerosion reduziert. Auch der Dünger wird durch gezielte Anwendung effizienter genutzt, was zu einer ressourcenschonenderen Landwirtschaft beiträgt.
- Energieeffiziente Gebäude: Intelligente Heizungs-, Kühl- und Beleuchtungssysteme passen sich an die Nutzung an. So wird zum Beispiel ungenutzte Beleuchtung automatisch ausgeschaltet, und Heizsysteme werden in leeren Räumen heruntergeregt. Dies spart nicht nur Energie, sondern reduziert auch die Betriebskosten erheblich.

Stromnetze: Intelligente Stromnetze (Smart Grids) nutzen IoT zur Optimierung des Energieverbrauchs und Integration erneuerbarer Energien.

2. Optimierung der Lieferketten:

- Transparenz: IoT-Geräte wie GPS-Tracker ermöglichen die lückenlose Nachverfolgung von Waren in Echtzeit. Dadurch können ineffiziente Routen vermieden und Lieferzeiten verkürzt werden. Dies spart Kraftstoff und reduziert CO₂-Emissionen.
- Reduzierung von Lebensmittelverschwendungen: In der Lebensmittelbranche überwachen smarte Verpackungen die Temperatur verderblicher Waren während des Transports. Bei Temperaturabweichungen kann frühzeitig eingegriffen werden, wodurch Lebensmittelabfall vermieden wird.

3. Unterstützung der Kreislaufwirtschaft:

Proaktive Wartung: IoT kann den Zustand von Maschinen und Geräten überwachen, um Anzeichen von Verschleiß frühzeitig zu erkennen. So werden Reparaturen rechtzeitig durchgeführt, bevor teure und ressourcenintensive Ausfälle auftreten.

- **Langlebigkeit und Recycling:** Nutzungsdaten helfen Herstellern, Schwachstellen in ihren Produkten zu erkennen und diese langlebiger zu gestalten. Auch das Recycling wird durch eine genauere Identifikation von Materialien in IoT-Geräten vereinfacht.
- **Intelligente Abfallwirtschaft:** IoT-Sensoren in Müllcontainern optimieren die Entleerungstouren von Entsorgungsunternehmen und reduzieren damit Verkehrsaufkommen und Energieverbrauch.
- 4. Umweltüberwachung:**
- **Klimaschutz:** IoT-Sensoren erfassen Daten zu Luftqualität, Emissionen und Wasserverunreinigungen. Diese Informationen können in Echtzeit übermittelt werden, um Umweltprobleme schneller zu identifizieren und gezielte Maßnahmen einzuleiten.
- **Städtische Infrastruktur:** Smarte Verkehrsmanagementsysteme analysieren Verkehrsflüsse auf Basis von IoT-Technologie in Echtzeit und optimieren Ampelschaltungen oder leiten Fahrzeuge um, um Staus zu vermeiden. Dies reduziert sowohl die Fahrzeiten als auch den Kraftstoffverbrauch und verbessert die Luftqualität.



Abbildung 2: (Quelle: Leonid - stock.adobe.com)

Herausforderungen: Nachhaltigkeitsrisiken des IoT

Trotz der vielversprechenden Möglichkeiten birgt der Einsatz von IoT auch Herausforderungen für die Umwelt:

1. Energieverbrauch:

- **Skalierungseffekte:** Die exponentielle Verbreitung von IoT-Geräten führt zu einem steigenden Strombedarf für deren Betrieb und Datenverarbeitung.
- **Rechenzentren als Energieverbraucher:** Cloud-Infrastrukturen, die IoT-Daten speichern und analysieren, sind oft sehr energieintensiv und benötigen nachhaltige Energielösungen.
- **Optimierungspotenzial:** Ohne den Einsatz von energieeffizienten Algorithmen und Hardware könnte der Betrieb von IoT-Netzwerken den Energiebedarf der IT-Industrie erheblich steigern.

2. Elektronikschrott:

- **Kurze Lebenszyklen:** Viele IoT-Geräte haben eine begrenzte Lebensdauer. Zudem erschweren proprietäre Technologien und fehlende Standards das Recycling.
- **Wachsende Abfallberge:** Laut der UNEP („UN Environment Programme“) entsteht jährlich

mehr als 50 Millionen Tonnen Elektroschrott. IoT könnte diese Zahl weiter erhöhen, wenn keine geeigneten Maßnahmen ergriffen werden.

- **Obsoleszenz durch technologische Entwicklung:** Kurze Innovationszyklen führen dazu, dass Geräte schneller veraltet und ersetzt werden müssen, was die Umweltbelastung erhöht.

3. Datensicherheit und Datenschutz:

- **Energieverbrauch durch Datenverarbeitung:** Die Speicherung und Analyse von IoT-Daten benötigen große Rechenkapazitäten und somit Energie. Unnötige Datensammlungen könnten diesen Effekt verschärfen.
- **Ethische Fragen:** Neben den Umweltaspekten wirft IoT Fragen des Datenschutzes und der Privatsphäre auf, da immer mehr persönliche Daten von IoT-Geräten gesammelt und verarbeitet werden.

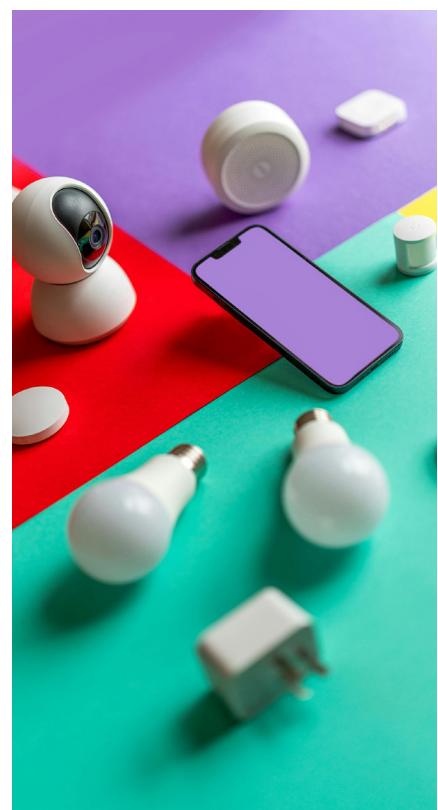


Abbildung 3: (Quelle: Foto von Jakub Zerdzicki auf Pexels)

Lösungsansätze: Nachhaltigkeit im IoT fördern

Nachhaltigkeit im IoT kann durch gezielte Maßnahmen entlang des gesamten Lebenszyklus von IoT-Geräten und -Infrastrukturen gefördert werden. Hier sind die wichtigsten Ansätze:

1. Green Design:

- Effiziente Hardware: Geräte sollten energiearme Sensoren und Chips nutzen, die minimalen Stromverbrauch garantieren und mit langlebigen Materialien ausgestattet sind.
- Modularität: Designs, die einfache Reparaturen und Upgrades erlauben, reduzieren Elektroschrott und sorgen für eine längere Nutzungsdauer.
- Einsatz recycelter Materialien: Hersteller können nachhaltige Materialien integrieren und hohe Recyclingquoten anstreben, um die Umweltbelastung zu minimieren

2. Betrieb:

- Erneuerbare Energiequellen: IoT-Geräte und die dazugehörige Infrastruktur, wie Rechenzentren, sollten auf Strom aus erneuerbaren Energien setzen. Dezentrale Energiequellen wie Solarenergie für Sensoren können ebenfalls helfen.
- Dynamische Datenverarbeitung: Edge Computing ermöglicht es, Daten näher an der Quelle zu verarbeiten, wodurch der Energieverbrauch für die Datenübertragung verringert wird. Redundante Datensammlungen sollten vermieden werden, um unnötige Rechenlasten zu minimieren.
- Nutzung von Low-Power-Netzwerken: Technologien wie LPWAN (Low Power Wide Area Networks) bieten energieeffiziente Kommunikationslösungen, besonders für Geräte mit geringer Bandbreitenanforderung

3. Standardisierung und Regulierung:

- Globale Standards: Einheitliche Protokolle und offene Standards erleichtern die Wiederverwendung und das Recycling von Geräten
- Verpflichtende Recyclingquoten: Hersteller sollten gesetzlich verpflichtet werden, hohe Anteile an recycelten Materialien in neuen IoT-Produkten zu verwenden.

4. Bewusstseinsbildung:

- Aufklärung: Unternehmen müssen Mitarbeitende und Kunden über nachhaltige Nutzungsmöglichkeiten informieren.
- Nachhaltiger Konsum: Verbraucher sollten Geräte bewusst auswählen und bei ihren Kaufentscheidungen auf Langlebigkeit und Effizienz achten.

Fazit

IoT und Nachhaltigkeit stehen nicht zwangsläufig im Widerspruch. Im Gegenteil: Richtig eingesetzt, kann IoT eine entscheidende Rolle bei der Bewältigung globaler Umweltprobleme spielen. Entscheidend ist jedoch, dass die Technologien verantwortungsvoll entwickelt und genutzt werden. Energieeffizienz, langlebige Designs und der Einsatz erneuerbarer Energien müssen dabei im Mittelpunkt stehen, um sicherzustellen, dass IoT zu einer nachhaltigen Zukunft beiträgt – statt sie zu gefährden.

Quellen und weiterführende Informationen:

Hinweis: Dieser Artikel wurde durch ChatGPT unterstützt.

[1] <https://explodingtopics.com/blog/number-of-iot-devices>

Podcast-Tipp



Martin Wildbacher ist Principal Consultant und Teamlead bei SEQIS.

Er ist in allen Testbereichen tätig: Planung und Koordination, Analyse und Testfallerstellung, Durchführung und Dokumentation sowie Problemmanagement.

Das Arbeiten in einem agilen, dynamischen Umfeld mit wechselnden Teams und Projekten sowie die Freude am Lösen von Herausforderungen sind für ihn die Essenz für den garantierten Erfolg im Softwaretest.

Nachhaltige Beziehungen bei SEQIS – mehr als nur ein kurzes Abenteuer

von Susanne Greber

Bei SEQIS stehen nachhaltige Beziehungen im Vordergrund – sowohl im Team als auch mit Kunden. Diese langfristigen Partnerschaften basieren auf Vertrauen, Respekt und gegenseitiger Wertschätzung und sind der Schlüssel zu unserem dauerhaften Erfolg.

Nachhaltigkeit bedeutet für uns mehr als umweltbewusstes Handeln – es geht um langfristige, wertschätzende Beziehungen. Denn Erfolg entsteht nicht über Nacht, sondern durch Vertrauen, Beständigkeit und gegenseitigen Respekt. Ob im Team oder mit unseren Kunden: Nur wenn wir auf verlässliche Partnerschaften setzen, können wir gemeinsam wachsen und wirklich etwas bewegen. Bei der HR Arbeit stelle ich sicher, dass Bewerber gleich zu Beginn des Prozesses verstehen, warum dieser Wert für uns so genauso wichtig ist wie Qualität, in allem, was wir tun.

Team: Vertrauen und Wertschätzung als Basis

Unser Team ist das Herzstück unseres Unternehmens – ohne die Menschen, die bei SEQIS arbeiten, wären wir nur ein schickes Logo mit einer netten Webseite. Deshalb setzen wir auf eine Unternehmenskultur, die Offenheit, Fairness und Zusammenarbeit auf Augenhöhe fördert. Wir glauben daran, dass sich Menschen dort am besten entfalten, wo sie sich geschätzt und gefördert fühlen.

Regelmäßige Feedbackgespräche, ehrliche Kommunikation und individuelle Entwicklungsmöglichkeiten sorgen dafür, dass wir uns stetig weiterentwickeln – als Einzelne und als Team. Unser Ziel ist es, ein Arbeitsumfeld zu schaffen, das nicht nur heute, sondern auch morgen und in Zukunft ein Ort ist, an dem

jeder seinen besten Job machen kann und jeden Tag mit Freude zur Arbeit kommt.

Kunden: Partnerschaft auf Augenhöhe

Auch unsere Kunden profitieren von dieser Haltung. Wir setzen auf langfristige Partnerschaften, geprägt von Vertrauen, Ehrlichkeit und Verlässlichkeit. Uns geht es nicht um den schnellen Gewinn, sondern darum, nachhaltige Lösungen zu entwickeln, die echten Mehrwert bieten.

Das Ergebnis? Zufriedene Kunden, die uns weiterempfehlen, immer wieder mit uns zusammenarbeiten und mit denen wir über Jahre hinweg Erfolge feiern. Denn wir glauben daran, dass eine gute Geschäftsbeziehung genauso wie eine starke Freundschaft funktioniert: mit Respekt, Verlässlichkeit und dem gemeinsamen Ziel, Großes zu erreichen.

Nachhaltigkeit als Erfolgsfaktor

Nachhaltige Beziehungen sind für uns kein Trend, sondern ein entscheidender Erfolgsfaktor. Ob innerhalb unseres Teams oder in der Zusammenarbeit mit unseren Kunden – durch kontinuierliche Investitionen in Beziehungen stellen wir sicher, dass SEQIS auch in Zukunft stabil, innovativ und erfolgreich bleibt.

Denn Erfolg ist keine Eintagsfliege – sondern das Ergebnis von echtem Engagement und verlässlichen Partnerschaften.

Also, auf eine lange und erfolgreiche Zusammenarbeit!



Abbildung 1: (Quelle: <https://undraw.co/>)



Susanne Greber ist HR Managerin bei SEQIS.

In ihrer Rolle ist sie für alle Personalthemen zuständig. Wichtig ist ihr dabei ein respektvoller Umgang auf Augenhöhe mit allen Mitarbeiter:innen und Bewerber:innen.

Als IT-affine nicht-ITlerin verfolgt sie gespannt, wie die technologische Weiterentwicklung, gerade im IT-Bereich, unser soziales Verhalten prägt und verändert.

Kampfansage an technische Schulden – Ein Managementproblem mit technischen Folgen

von Alexander Weichselberger

Einleitung: Das unsichtbare Erbe von IT-Projekten

Technische Schulden begegnen uns in nahezu jedem digitalen Projekt – häufig erst dann, wenn es schmerzt. Langsame Release-Zyklen, hohe Wartungskosten oder stockende Innovationen sind oft Symptome eines Problems, das nicht primär im Code, sondern im Management verankert ist. Denn was als pragmatische Abkürzung im Projektalltag beginnt, entwickelt sich schnell zu einem schwer zu steuernden Risiko. Dieser Artikel beleuchtet, wie technische Schulden entstehen, welche Auswirkungen sie auf Organisationen haben und wie Projekt- und Change Management einen entscheidenden Beitrag zur Prävention und Auflösung leisten können.

Was sind technische Schulden – und woher kommen sie wirklich?

Der Begriff „technische Schulden“ wurde ursprünglich von Ward Cunningham geprägt, um zu beschreiben, dass eine schnelle, aber unsaubere Lösung kurzfristig Vorteile bringt, langfristig jedoch „Zinsen“ in Form von Mehraufwand kostet. Heute verstehen wir darunter eine Vielzahl von Versäumnissen, die nicht nur im Code, sondern auch in Prozessen, Architekturentscheidungen oder fehlenden Dokumentationen liegen.



Abbildung 1: Technical Debt - what lies beneath the surface (Quelle: Image generated by Gemini for Workspace)

Typische Ursachen aus Managementsicht:

- **Zeit- und Budgetdruck:** Liefterminte priorisieren, kurzfristige Ergebnisse vor langfristiger Qualität
- **Unklare Anforderungen und fehlende Abstimmungen:** Was nicht klar definiert ist, wird „irgendwie“ gelöst
- **Organisatorische Silos:** Entwicklung, Betrieb und Fachbereiche agieren isoliert – technologische Entscheidungen bleiben unreflektiert
- **Fehlendes Bewusstsein:** Technische Schulden gelten oft als rein technisches Problem und bleiben deshalb lange unadressiert

Symptome und Auswirkungen: Wenn Schulden zur Hypothek werden

Technische Schulden bauen sich schlechend auf, doch ihre Effekte sind messbar:

- **Erhöhte Wartungsaufwände** und sinkende Produktivität der Teams
- **Geringere Innovationsfähigkeit**, da Ressourcen für technische Reparaturen gebunden sind
- **Demotivierte Teams**, die in einem instabilen System arbeiten müssen
- **Steigende Projekt- und Betriebskosten**, oft erst Jahre nach Projektabschluss sichtbar

Besonders gefährlich: In großen Organisationen werden technische Schulden oft nicht als Risiko geführt – sie sind unsichtbar im Portfolio- und Risikomanagement.



FROM MANAGEMENT TO DEBT LOAD – A SYSTEMIC PROBLEM

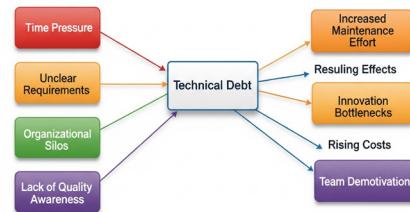


Abbildung 2: From Management to Debt Load - a systematic problem (Quelle: Image generated by Gemini for Workspace)

Die Ausgangsbasis: Technische Schulden messbar machen

Bevor man technische Schulden aktiv managen kann, muss man wissen, wo man steht. Doch wie erhebt man etwas, das häufig nicht dokumentiert ist?

Methoden zur Erhebung:

- **Technische Assessments:** Codequalität, Testabdeckung, Architektur-Reviews
- **Organisatorische Analysen:** Wo wurden pragmatische Abkürzungen genommen? Welche Entscheidungspfade gab es?
- **Befragungen & Workshops:** Das Wissen der Entwickler*innen und PMs ist oft der beste Indikator
- **Technical Debt Register:** Systematische Erfassung mit Beschreibung, Risiko, Ursache, Priorität

Distribution of Technical Debt by Area

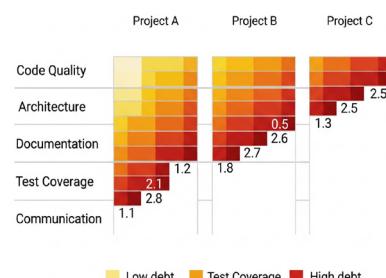


Abbildung 3: Distribution of Technical Debt by Area : Projects Quelle: Image generated by Gemini for Workspace)

Bereich	Beschreibung	Ursache	Risiko	Aufwand (Tage)	Priorität
Architektur	Monolith statt Modularisierung	Zeitdruck	Hoch	15	Hoch
Testabdeckung	Keine automatisierten Tests	Ressourcenmangel	Mittel	10	Mittel
Dokumentation	Veraltete technische Doku	Pflege vernachlässigt	Mittel	5	Niedrig
Kommunikation	Entscheidungen nicht nachvollziehbar	Fehlendes Prozessverständnis	Hoch	8	Mittel

Tabelle 1: Example for Technical Dept Register (Quelle: SEQIS GmbH)

Strategien zur Bewältigung: Change-Management statt Reparaturmodus

Der Abbau technischer Schulden ist kein IT-Fix, sondern eine Veränderung in Kultur, Prozessen und Steuerung. Projekt- und Change Management übernehmen hier eine zentrale Rolle. Erfolgreiche Strategien beinhalten:

- **Schulden als Teil der Projektplanung:** Jede Iteration kann ein kleines Refactoring enthalten – geplant, nicht nebenbei.
- **Business Case für Qualität:** Die Kosten technischer Schulden (z. B. Downtime, langsame Time-to-Market) müssen kommuniziert werden.
- **Stakeholder einbinden:** Entscheidungsträger müssen verstehen, dass Schuldenmanagement eine Investition in Zukunftsfähigkeit ist.
- **Transparente Kommunikation:** Schulden benennen, sichtbar machen – und über Fortschritte berichten.
- **Agile Einbettung:** Backlogs und Sprintpläne sollten explizit Raum für technischen Schuldenabbau lassen.

Fazit: Wer technische Schulden ignoriert, zahlt doppelt

Technische Schulden sind keine rein technische Herausforderung – sie sind das Ergebnis von Entscheidungen auf allen Ebenen des Projekts.

Projekt- und Change Manager spielen eine Schlüsselrolle dabei, sie frühzeitig zu erkennen, transparent zu machen und systematisch abzubauen.

Wer eine nachhaltige IT-Landschaft will, muss den Schulden den Kampfansagen – nicht als einmalige Aufräumaktion, sondern als integralen Bestandteil guter Projektführung.

Alexander Weichselberger: „Sie möchten mehr darüber erfahren, wie technische Schulden aktiv und pragmatisch gemanagt werden können? Dann besuchen Sie meinen Vortrag – wir gehen gemeinsam den ersten Schritt zur Entschuldung.“



Kampfansage an technische Schulden – Ursachen, Auswirkungen und Strategien im Projekt- und Change Management

am 13.11.2025



Alexander Weichselberger ist Managing Partner.

Er hat seine Einsatzschwerpunkte in den Bereichen Systemanalyse, Softwaretest, Koordination und Management von exponierten Großprojekten und kann auf jahrelange Erfahrung zurückblicken.

Dieses Wissen gibt er gerne in Form von Coachings, Methodentrainings und Fachvorträgen weiter.

Die nächsten
Termine im
Überblick:



Ten more things

Kommunikation und
Knowledge Management in
IT-Projekten

am 25.09.2025



Abbildung: (Quelle: undraw.co)

Juli		August	
1	Di	1	Fr
2	Mi	2	Sa
3	Do	3	So
4	Fr	4	Mo
5	Sa	5	Di
6	So	6	Mi
7	Mo	7	Do
8	Di	8	Fr
9	Mi	9	Sa
10	Do	10	So
11	Fr	11	Mo
12	Sa	12	Di
13	So	13	Mi
14	Mo	14	Do
15	Di	15	Fr Mariä Himmelfahrt
16	Mi	16	Sa
17	Do	17	So
18	Fr	18	Mo
19	Sa	19	Di
20	So	20	Mi
21	Mo	21	Do
22	Di	22	Fr
23	Mi	23	Sa
24	Do	24	So
25	Fr	25	Mo
26	Sa	26	Di
27	So	27	Mi
28	Mo	28	Do
29	Di	29	Fr
30	Mi	30	Sa
31	Do	31	So

September	
1	Mo
2	Di
3	Mi
4	Do
5	Fr
6	Sa
7	So
8	Mo
9	Di
10	Mi
11	Do
12	Fr
13	Sa
14	So
15	Mo
16	Di
17	Mi
18	Do
19	Fr
20	Sa
21	So
22	Mo
23	Di
24	Mi
25	Do
Ten more things	
26	Fr
27	Sa
28	So
29	Mo
30	Di

Oktober	
1	Mi
2	Do
3	Fr
4	Sa
5	So
6	Mo
7	Di
8	Mi
9	Do
10	Fr
11	Sa
12	So
13	Mo
14	Di
15	Mi
16	Do
17	Fr
18	Sa
19	So
20	Mo
21	Di
22	Mi
23	Do
24	Fr
25	Sa
26	So
Nationalfeiertag	
27	Mo
28	Di
29	Mi
30	Do
31	Fr



Ten more things

Kampfansage an technische
Schulden – Ursachen,
Auswirkungen und
Strategien im Projekt- und
Change Management

am 13.11.2025



Nachhaltigkeit in der DevOps-Pipeline: Vom Code bis zum Deployment

von Emre Kurtulush



Abbildung 1: (Quelle: Bild generiert von Gemini (Google AI))

Die IT hat sich rasant entwickelt und unser Leben tiefgreifend verändert. Doch mit den neuen Möglichkeiten rücken auch die ökologischen Folgen stärker in den Fokus. Früher lag der Nachhaltigkeitsfokus auf sichtbaren Umweltaspekten – heute erkennen wir, dass auch digitale Prozesse Ressourcen verbrauchen. Gerade in der DevOps-Pipeline hinterlassen unsere Entscheidungen Spuren. Wie können wir also Nachhaltigkeit ganzheitlich integrieren, ohne Effizienz und Agilität zu verlieren?

Warum ist Nachhaltigkeit in DevOps wichtig?

Die Informationstechnologie (IT) verursacht weltweit etwa 2 bis 4 % der CO₂-Emissionen, mit steigender Tendenz. Eine Studie von 2021 schätzt, dass der Anteil der IT an den globalen Treibhausgasemissionen zwischen 1,8 % und 2,8 % liegt, wobei die tatsächlichen Werte aufgrund unvollständiger Erfassung der gesamten Lieferkette möglicherweise bis zu 25 % höher sind, was einen Anteil von bis zu 3,5 % ergeben könnte. Eine weitere Studie aus dem Jahr 2023 gibt an, dass die Emissionen der IT-Branche fast 4 % der weltweiten Gesamtemissionen ausmachen und prognostiziert, dass dieser Anteil in

den nächsten zwei Jahrzehnten stark ansteigen wird.^[1]

Insbesondere Rechenzentren tragen erheblich zu diesen Emissionen bei. Laut einer Analyse von Morgan Stanley wird die globale Rechenzentrumsbranche bis 2030 voraussichtlich 2,5 Milliarden Tonnen CO₂-Äquivalente emittieren.^[2]

Diese Zahlen verdeutlichen die Notwendigkeit, Nachhaltigkeitspraktiken in der IT-Branche zu implementieren, um den ökologischen Fußabdruck zu reduzieren.

Nachhaltigkeit in jeder Phase der DevOps-Pipeline

1. Nachhaltiger Code

1.1 Effiziente Programmierung

Ein nachhaltiger Softwareentwicklungsprozess beginnt mit der Codierung. Effizienter Code reduziert die benötigte Rechenleistung und minimiert damit den Energieverbrauch. Dies kann durch:

- Verwendung von Algorithmen mit geringerem Rechenaufwand
- Vermeidung unnötiger Berechnungen oder Schleifen
- Verwendung von Speicher- und CPU-schonenden Datenstrukturen

Diese Maßnahmen tragen dazu bei, dass weniger Rechenzeit und Speicher benötigt werden, was wiederum den Stromverbrauch von Servern und Endgeräten senkt.^[3]

1.2 Wahl nachhaltiger Programmiersprachen

Nicht alle Programmiersprachen sind gleich effizient. Sprachen wie Rust oder Go sind oft ressourcenschonender als interpretierte Sprachen wie Python oder JavaScript. Der Grund hierfür liegt in der Art und Weise, wie

Code ausgeführt wird:

- Kompilierte Sprachen (z. B. C, Rust, Go) werden vor der Ausführung in Maschinencode übersetzt, was zu schnellerer Laufzeit und geringerem Energieverbrauch führt.
- Interpretierte Sprachen (z. B. Python, JavaScript) erfordern eine zusätzliche Verarbeitungsschicht zur Laufzeit, was den Energieverbrauch erhöht.

Laut einer Studie der Universidade do Minho in Portugal zeigt sich, dass C und Rust hinsichtlich Energieverbrauch, Laufzeit und Speicherbedarf besonders effizient sind, während Python und Perl aufgrund ihrer interpretativen Natur höhere Ressourcenanforderungen haben.^[4]

2. Nachhaltige Build-Prozesse

2.1 Minimierung von Build-Ressourcen

Die Build-Phase kann ressourcenintensiv sein. Nachhaltige Praktiken beinhalten:

- Nutzung von inkrementellen Builds, um unnötige Wiederholungen zu vermeiden
- Caching von Abhängigkeiten und bereits kompilierten Modulen
- Reduzierung von unnötigen Abhängigkeiten

Durch diese Optimierungen wird weniger Rechenleistung benötigt, was sich direkt auf die Energieeffizienz auswirkt. Beispielsweise reduziert ein optimierter Build-Prozess die Anzahl der CPU-Zyklen und damit den Stromverbrauch von Rechenzentren.^[3]

2.2 Optimierung von Container-Images

Da viele DevOps-Umgebungen auf Containern basieren, ist es wichtig,

Container-Images schlank zu halten. Dies kann durch die Verwendung von:

- Minimale Basis-Images
- Entfernen unnötiger Dateien
- Multi-Stage-Builds erreicht werden.

Kleinere Container-Images benötigen weniger Speicherplatz und führen zu schnelleren Deployments, was sowohl die Netzwerkbela stung als auch den Energieverbrauch verringert.^[5]

3. Nachhaltige Test-Strategien

3.1 Effiziente Testautomatisierung

Automatisierte Tests sind essenziell für DevOps, sollten aber nachhaltig gestaltet werden:

- Parallelisierung von Tests zur Reduzierung der Laufzeiten
- Nutzung von Mocking und Stubbing, um Tests schneller auszuführen
- Selektive Testläufe basierend auf Code-Änderungen

Durch die gezielte Ausführung nur der relevanten Tests wird unnötige Rechenleistung vermieden, was den Energieverbrauch reduziert.^[6]

4. Nachhaltiges Deployment und Betrieb

4.1 Energieeffizientes Hosting

Beim Deployment spielt die Wahl des Hostings eine große Rolle. Möglichkeiten für nachhaltiges Hosting sind:

- Nutzung von Cloud-Anbietern mit erneuerbaren Energien
- Serverless-Computing zur Optimierung der Ressourcennutzung
- Skalierung von Anwendungen nach tatsächlichem Bedarf

Rechenzentren verbrauchen weltweit enorme Mengen an Energie. Unternehmen wie Google oder Microsoft setzen daher vermehrt auf Rechenzentren, die mit erneuerbaren Energien betrieben werden, um den CO₂-Fußabdruck zu reduzieren.^[7]

4.2 Monitoring und Optimierung

Nachhaltigkeit endet nicht mit dem

Deployment. Eine kontinuierliche Überwachung hilft, unnötige Ressourcenverschwendungen zu vermeiden:

- Identifikation ineffizienter Prozesse oder überflüssiger Dienste
- Dynamische Anpassung von Ressourcen durch Autoscaling
- Nutzung von GreenOps-Strategien zur Reduzierung der CO₂-Emissionen

Durch den Einsatz von Monitoring-Tools kann festgestellt werden, welche Anwendungen oder Dienste unnötig viele Ressourcen verbrauchen, sodass gezielte Optimierungen vorgenommen werden können.^[8]

Fazit

Nachhaltigkeit in der DevOps-Pipeline erfordert eine durchgehende Optimierung des Entwicklungs-, Test- und Deployment-Prozesses. Durch effiziente Codierung, ressourcenschonende Builds, optimierte Tests und nachhaltiges Hosting können Unternehmen nicht nur ihre Umweltbilanz verbessern, sondern auch langfristig Kosten sparen. DevOps und Nachhaltigkeit sind somit keine Gegensätze, sondern können Hand in Hand gehen, um eine bessere Zukunft zu gestalten.

Quellen und weiterführende Informationen:

^[1] <https://arxiv.org/abs/2102.02622>

^[2] <https://arxiv.org/abs/2303.06642>

^[3] <https://distantjob.com/blog/sustainable-software-development/>

^[4] Pereira, Rui et al. „Energy Efficiency across Programming Languages.“ Universidade do Minho, 2017.

^[5] Hüttermann, Michael. „DevOps for Developers.“ Apress, 2012.

^[6] Fowler, Martin. „Continuous Integration.“ ThoughtWorks, 2006.

^[7] Greenpeace. „Clicking Clean: Who is Winning the Race to Build a Green Internet?“ 2017.

^[8] Bauer, Michael. „Green IT: Technologien für eine nachhaltige Zukunft.“ Springer, 2018.



Emre Kurtulush ist Consultant bei SEQIS.

Mit einem Hintergrund in Elektrotechnik und Informationstechnik fand er seinen Weg in die Welt der Softwaretests, wo ihn besonders die Verbindung von Technik, Struktur und Qualität begeistert. Erste Erfahrungen sammelte er in technologieintensiven Projekten, bei denen er sein technisches Verständnis und seine analytische Denkweise gezielt einsetzen konnte. Heute liegt sein Fokus auf den technischen Aspekten des Testens – darunter Testautomatisierung, API-Tests und die Optimierung von Testprozessen. Er verfolgt das Ziel, sich kontinuierlich weiterzuentwickeln, neue Werkzeuge und Methoden kennenzulernen und so aktiv zur Verbesserung der Softwarequalität beizutragen.

IT-Analyse III - das Modell im Überblick

von Josef Falk



Abbildung 1: (Quelle: SEQIS GmbH)

In der letzten Folge wurden verschiedene Konzepte zur IT-Analyse aus Gegenwart und Vergangenheit vorgestellt. Auf dieser Basis soll in diesem Artikel dieses Berufsbild so beschrieben werden, wie es in der Praxis tatsächlich gelebt werden kann – und auch wird.^[1]

In einem ersten Schritt wird der Inhalt des Berufsbild IT-Analyse in einer Kurz-Form dargestellt. Im zweiten Schritt wird die in den bisherigen Folgen – unhinterfragt – verwendete Bezeichnung „IT-Analytiker“ diskutiert. Darauf aufbauend werden die einzelnen Elemente des im ersten Schritt skizzierten Berufsbildes erläutert.

IT-Analyse – kurz gefasst

In der konkreten Ausprägung wird die Analyse in jedem Projekt anders gelebt. Und doch gibt es einen gemeinsamen Kern, auf den sich die Tätigkeit in allen Varianten reduzieren lässt.

Dieser gemeinsame Kern lässt sich in einem Satz ausdrücken:

IT-Analyse ist das Transformieren von Anforderungen in den Entwurf eines IT-Systems.

Diese Definition enthält drei Elemente:

- die **Anforderungen** als den Input für die Analyse-Tätigkeit.
- den **Entwurf**, der das Ziel und Ergebnis der Tätigkeit ist.
- das **Transformieren**, was aus den Anforderungen einen Entwurf macht.

Diese Elemente sollen in weiterer Folge im Detail behandelt werden. Zuvor wollen wir uns aber noch Gedanken über die Bezeichnung für die Funktion machen, deren Hauptaufgabe diese Transformation ist.

Der Job-Title – mehr als Schall und Rauch?

Namen sind Schall und Rauch. Wichtig ist der Inhalt der Tätigkeit. Wie man das dazugehörige Berufsbild dann nennt, ist zweitrangig. Aber es erleichtert doch die Kommunikation, wenn man eine gemeinsame Bezeichnung dafür hat.

Im bisherigen Verlauf dieser Artikel-Serie wurde unhinterfragt die Bezeichnung „IT-Analyse“ für das Berufsbild verwendet. Das soll nun begründet werden.

Zwei Anforderungen seien hier für einen geeigneten Job-Title genannt:

1. Der Job-Title soll den Inhalt der Tätigkeit zum Ausdruck bringen
2. Der Job-Title soll geschäftstauglich sein: d. h. er soll nicht zu lang sein und er soll keine sehr ungebräuchlichen Ausdrücke verwenden.

Würde man nur die erste Anforderung berücksichtigen, so müsste die Berufsbezeichnung „Anforderungs-Transformator“ oder so lauten. Das klingt aber erstens nicht besonders attraktiv, und würde auch nicht verstanden werden.

Sehen wir uns heute übliche Berufsbezeichnungen unter diesen Gesichtspunkten an:

- **Requirements Engineer** ist ein eingeführter Job-Title, der sich auf vielen Job-Profilen und Visitenkarten findet. Eigentlich beschreibt er die Tätigkeit aber nicht richtig, denn das „Engineering“ bezieht sich in der Realität nicht auf die Anforderungen, sondern auf die zu entwickelnde Lösung. Das kann zu falschen Erwartungen führen.
- Mehr und mehr hat sich in den letzten Jahren die Berufsbezeichnung **Business Analyst** durchgesetzt. Viele Ausübende der hier beschriebenen Funktion – so auch der Autor – haben sich gleichzeitig mit beiden Job-Titles bezeichnet: Business Analyst/ Requirements Engineer. Die Business Analyse ist jedoch ein weites Feld und wird nicht überall mit IT in Verbindung betrachtet. So ist diese Bezeichnung sehr unspezifisch.
- **Systemanalytiker** waren vor einigen Jahrzehnten ein eingeführter Begriff für die Analyse in

IT-Projekten. Heute wirkt diese Bezeichnung aber etwas verstaubt und veraltet.

Die genannten Job-Titles drücken den Inhalt der Tätigkeit nicht sehr gut aus. Teils verweisen sie lediglich auf einen Teilaspekt der Funktion, teils sind sie sehr unspezifisch.

Der Teil der Bezeichnung „Analytiker“ macht – im Allgemeinen – schon eine sehr treffende Aussage. Damit diese spezifischer wird, wird sie – im vorliegenden Vorschlag, um das Präfix „IT-“ ergänzt, sodass sich „IT-Analytiker“ als Berufsbezeichnung ergibt. Allerdings ist auch diese Bezeichnung nicht perfekt. „Analyse“ kommt aus dem Griechischen und bedeutet „Zerlegung“. Das trifft den Inhalt gut – die Anforderungen werden in ihre Elemente zerlegt. Sie werden im Anschluss jedoch auch wieder zum Entwurf zusammengesetzt – diese „Synthese“ ist in der Bezeichnung nicht enthalten. Trotzdem erscheint „IT-Analyse“ als die beste der aktuell in Verwendung stehenden Optionen und wird hier in Folge weiter verwendet.

Ein Modell der IT-Analyse

„IT-Analyse ist das Transformieren von Anforderungen in den Entwurf eines IT-Systems“. Diese Definition für das Berufsbild IT-Analyse wurde oben eingeführt. In diesem Abschnitt soll diese Definition näher betrachtet werden.

Weitere Artikel aus dieser Reihe:

IT-Analyse I: Wer braucht eigentlich IT-Analyse?

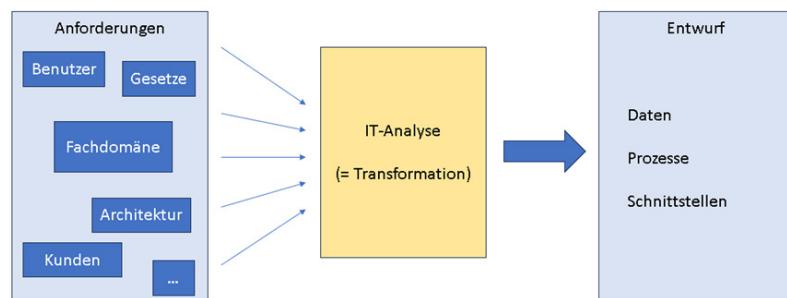


Abbildung 1: Anforderungen - Analyse - Entwurf (Quelle: SEQIS GmbH)

Die Anforderungen

Um die Anforderungen dreht sich alles. Sie sind die Existenzberechtigung für den gesamten Software-Entwicklungsprozess. Ein Fachbereich, ein Unternehmen benötigt eine bestimmte Funktionalität – und formuliert diese als Anforderung. Das Team, das diese Anforderung umsetzen soll, erhält einen Auftrag, der meistens durch ein Dokument beschrieben wird.

Dieses Dokument, das z. B. eine Ausschreibung sein könnte, ist der Ausgangspunkt. In seltenen, einfachen Fällen kann diese Anforderung unmittelbar in Software umgesetzt werden.

Meistens jedoch bestehen Fragen, Unklarheiten, Widersprüchlichkeiten, die geklärt werden müssen, damit das fertige Software-Produkt nicht ihren gewünschten Zweck verfehlt.

Die Klärung dieser Fragen, Unklarheiten, Widersprüchlichkeiten wird Anforderungserhebung genannt. Dafür gibt es eine Reihe von Methoden. Die wichtigste davon ist das Gespräch mit den sogenannten Stakeholdern, also mit allen, die von der zu entwickelnden Software betroffen sind.

Manchmal wird dieses Gespräch auch durch eine schriftliche Befragung ersetzt, etwa dann, wenn es sehr viele von diesen Stakeholdern gibt. In vielen Fällen wird auf vorhandene schriftliche Dokumente zurückgegriffen.

Um diese Anforderungen auch ver-

stehen zu können, muss der Analytiker selbst Experte in der Fachdomäne werden. Das passiert mehr oder weniger automatisch, wenn man für längere Zeit im gleichen Gebiet tätig ist. Bei einem Wechsel in ein neues Fachgebiet geht es darum, dieses Expertentum schnell zu erwerben. Nur dann ist es möglich, mit den Stakeholdern auf Augenhöhe zu sprechen. Andernfalls können die Anforderungen nur kritiklos entgegengenommen werden.

Der Entwurf

Das Ergebnis der Analyse-Tätigkeit ist ein Entwurf des geplanten Software-Systems, bzw. der Änderung an einem bestehenden System. Obwohl jedes Projekt anders ist, gibt es doch eine Systematik, wie ein derartiger Entwurf aussieht – eine Systematik, die für jedes Projekt anwendbar ist. So kann jeder Entwurf unter drei Aspekten betrachtet werden:

- Der statische Aspekt: Die Daten
- Der dynamische Aspekt: Die Prozesse
- Die Schnittstellen.

Statik – die Daten

Es geht immer um Daten. „Datenverarbeitung“ ist ein etwas verstaubter Begriff für das Fachgebiet, in dem sich auch die IT-Analyse bewegt. Der aktuelle Begriff IT („Information Technology“) sagt letztlich nichts anderes aus. Inhalt und Ziel jedes Software-Entwicklungsprozesses sind immer Daten:

- Daten, die ans Finanzamt geliefert werden müssen,
- Daten, die Maschinen steuern,
- Daten, die dazu dienen, Aufträge

- von Kunden zu bearbeiten,
- Daten, die Informationen über Lagerbestände abbilden,
- Daten, mit denen die Entwicklung von Unternehmen analysiert werden.

Das sind nur einige wenige Anwendungsfälle für die Daten, die verwaltet werden.

In jedem Fall haben diese Daten eine bestimmte Struktur, der Inhalt des Entwurfes ist, der in der Analyse entsteht. Das ist der statische Aspekt des Entwurfs.

Dynamik – die Prozesse

Diese Daten unterliegen einem Prozess. Sie entstehen irgendwo, sie wandern durch verschiedene Stellen innerhalb der Organisation, sie werden transformiert, aggregiert, angezeigt. Diese Prozesse werden im dynamischen Aspekt des Entwurfs beschrieben. Prozesse gibt es auf drei Ebenen:

- Workflow-Ebene: Ein Geschäftsfall wandert durch mehrere Stellen, bis der Prozess abgeschlossen ist. Zum Beispiel: ein Kreditantrag muss je nach Höhe von mehreren Stellen genehmigt werden, bis es zur Auszahlung kommen kann.
- Orchestrations-Ebene: Mehrere Services (oder andere Software-Bausteine) werden hintereinander – oder auch parallel – aktiv, damit ein bestimmtes Ziel erreicht wird.
- Algorithmus-Ebene: das ist der Ablauf innerhalb eines Software-Bausteines, der benötigt wird, um – zum Beispiel – eine bestimmte Berechnung durchzuführen.

Die Gestaltung jeder dieser Ebenen ist der Inhalt des Entwurfs, der durch die Analyse entsteht.

Schnittstellen

Der dritte Aspekt, der im Entwurf zu berücksichtigen ist, ist die der

Schnittstellen. In der Analyse müssen sowohl die Schnittstellen nach außen als auch innerhalb der einzelnen Bestandteile des zu entwerfenden Systems definiert werden.

Benutzer-Schnittstelle

Eine besondere Rolle unter den Schnittstellen nimmt die Benutzer-Schnittstelle ein. Vor allem dann, wenn es um die Schnittstelle zu Kunden geht, spielen hier nicht nur technische Gesichtspunkte eine Rolle. Es muss auch eine User-Experience geschaffen werden. Deshalb liegt die Aufgabe der Schaffung der Schnittstelle nach außen häufig bei spezialisierten Experten, mit denen der IT-Analyst zusammenarbeitet.

System-Schnittstellen nach außen

Wenn externe Systeme angebunden werden müssen, dann besteht häufig ein sehr geringer Gestaltungsspielraum. Das externe System bietet eine Schnittstelle an, die von dem neu zu entwickelnden System erfüllt werden muss, wenn diese mit dem externen System Daten austauschen möchte.

Schnittstellen nach innen

Auch die Bausteine innerhalb eines Systems sind durch Schnittstellen miteinander verbunden. Und auch diese werden in der Tätigkeit des Entwurfens gestaltet.

Die Analyse – von der Anforderung zum Entwurf

Zwischen Anforderung und Entwurf steht die Analyse. Analyse ist die Tätigkeit, in der die Anforderungen zum Entwurf werden.

Analyse ist ein geistiger, kreativer Prozess, der im Kopf des Analytikers stattfindet und der sich deshalb einer exakten Beschreibung entzieht.

Es gelten jedoch auch hier die allgemeinen Regeln für kreative Prozesse, wie sie unter anderem von Graham Wallas^[2] beschrieben wurden. Demgemäß besteht ein kreativer Prozess aus vier Schritten:

1. Vorbereitung: Hier wird Wissen über das zu lösende Problem aufgebaut. Im Fall der Analyse ist das die intensive Beschäftigung mit den Anforderungen und – darüber hinaus gehend – mit der Fachdomäne, in der das geplante IT-System angesiedelt ist. Das Erstellen eines Entwurfs ist keine triviale Angelegenheit. Der Entwurf ergibt sich nicht unmittelbar aus den Anforderungen. Es bedarf Überlegungen – und diese Überlegungen stoßen in vielen Fällen auf Widersprüche, die aufzulösen sind.
2. Inkubation: Wenn sich das Problem in der Vorbereitungsphase nicht lösen lässt, dann ist es zweckmäßig, sich gedanklich vom Problem zu lösen. Das Unterbewusstsein arbeitet dann weiter am Problem.
3. Illumination: Das Unterbewusstsein meldet sich, wenn es eine Lösung gefunden hat. Das kennt jeder Analytiker, wenn ihm eine Entwurfsidee in der Dusche oder beim Radfahren kommt.
4. Verifikation: Dieser Geistesblitz aus dem Unterbewusstsein muss dann aber bewusst überprüft und mit den Stakeholdern abgestimmt werden. Das passiert in der vierten Phase des Kreativitätsprozesses.

Weitere Artikel aus dieser Reihe:

IT-Analyse II: Eine Bestandsaufnahme



4-Phasen-Modell der Kreativität (nach Graham Wallas)

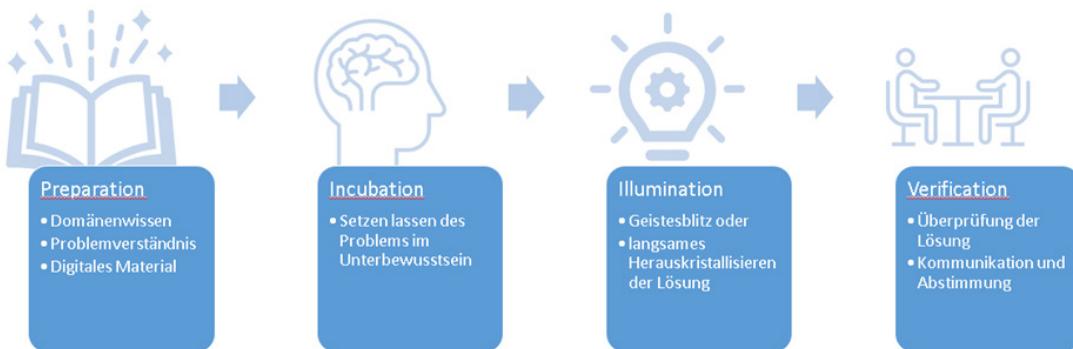


Abbildung 2: 4-Phasen-Modell der Kreativität nach Graham Wallas (Quelle Abbildung: SEQIS GmbH und Icons: PowerPoint)

Der so entstehende Entwurf ist in weiterer Folge die Basis für die Entwicklung, wo der Entwurf in die Lösung umgesetzt wird.

Dieses Modell gilt unabhängig vom Grad der Agilität, die im jeweiligen Projekt eingesetzt wird. Der Entwurf muss nicht vollständig vorliegen, wenn mit der Entwicklung begonnen wird. Es muss jedoch einen Gesamtüberblick geben und einen Detail-Entwurf jener Teile, die in die Entwicklung gehen.

Zusammenfassung und Ausblick

In diesem Artikel wurde der Überblick eines Modells für die IT-Analyse vorgestellt. Nach diesem Modell ist IT-Analyse „die Transformation von Anforderungen in einen Entwurf“.

Im nächsten Beitrag wird dieses Modell noch weiter im Detail dargestellt.



Abbildung 3: (Quelle <https://undraw.co/illustrations>)

Weitere Artikel aus dieser Reihe:

IT-Analyse I - Wer braucht eigentlich IT-Analyse?

<https://www.seqis.com/de/blog/wer-braucht-eigentlich-it-analyse>

[1] IT-Analyse II - Eine Bestandsaufnahme

<https://www.seqis.com/de/blog/it-analyse-eine-bestandsaufnahme>

Quellen:

[2] Graham Wallas: The Art of Thought



Josef Falk ist Principal Consultant bei SEQIS.

Seit dem Abschluss seines Studiums der Betriebswirtschaftslehre in Wien gestaltet er Lösungen in den unterschiedlichsten Fachbereichen – und ist dabei Mittler zwischen Fachbereich und IT-Entwicklung.

Besonderes Augenmerk legt er bei der Analyse auf den Innovationsgrad. Neben seiner Projekttätigkeit befasst er sich mit der Entwicklung der Business Analyse.



Smart Testing Strategies for Resource-Efficient Software Development

von Maged Elsherif

The Need for Sustainable Software Testing

In today's fast-paced digital world, software development is a critical driver of technological progress. However, behind every application, website, or digital service lies an amount of computational effort that consumes resources, generates heat, and requires electricity. Software testing, while essential to ensuring reliability and functionality, is one of the most resource-intensive aspects of development. As companies push for faster releases and automated workflows, testing can easily become redundant, leading to wasted time, inflated costs, and unnecessary energy consumption.

With climate change and sustainability at the forefront of global discussions, industries across all sectors are encouraged to reduce their environmental footprints. The software industry, often overlooked in conversations about sustainability, is actually a significant contributor to energy consumption. According to a study by the International Energy Agency, data centers—many of which run extensive software testing processes—account for nearly 1%^[1] of global electricity consumption. This statistic underscores the urgent need for more efficient testing methodologies that not only save time and resources but also contribute to a greener, more sustainable future.

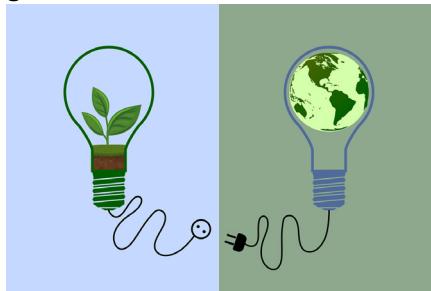


Abbildung 1: (Quelle: RosZie - Pixabay)

Understanding the Problem

Redundancy in software testing occurs when multiple test cases cover the same functionality, leading to unnecessary execution cycles. While having multiple layers of verification can be beneficial, excessive duplication does not necessarily lead to better software quality; instead, it often results in wasted computational power and increased test execution times. Studies have shown that between 30-50% of test cases in automated test suites are redundant, meaning they do not provide additional value beyond what has already been tested.

For instance, in large-scale enterprise applications with complex Continuous Integration/Continuous Deployment (CI/CD) pipelines, developers might run thousands of test cases per commit, many of which are either irrelevant or duplicate. Over time, this creates inefficiencies that slow down software delivery and consume excessive resources. Companies looking to optimize their testing efforts must therefore shift towards more strategic and intelligent testing approaches.

Smart Testing Strategies to Optimize Efficiency

Risk-Based Testing

Risk-based testing prioritizes test cases based on the likelihood and impact of defects, allowing teams to focus on the areas most likely to fail or those critical to business operations. Instead of testing every part of an application, testing efforts are concentrated on the areas that carry the highest risk.



How to Implement Risk-Based Testing in Your Project

- Identify Critical Areas Early in the Development Process
- Leverage Historical Data for Better Risk Assessment
- Continuously Reassess Risk Throughout the Development Cycle
- Adopt a Risk Scoring System
- Automate Testing for High-Risk Areas
- Use Tools Like the Journey to Rome Jira Extension to Identify Risk-Exposed Areas

Journey to Rome is a lightweight, questionnaire-based risk assessment method integrated into Jira. It helps identify the level and type of risk for new or existing functionality without needing input from developers. The results guide test intensity and focus, making it easier to prioritize where testing is most needed.

A real-world example of the impact of risk-based testing is discussed in the study „SUPERNOVA: Automating Test Selection and Defect Prevention in AAA Video Games Using Risk-Based Testing and Machine Learning“. This paper presents a system that leverages risk-based testing and machine learning to optimize test selection and defect prevention in video game development. The authors report a reduction of over 55% in testing hours for a sports game title, achieved without compromising defect detection effectiveness. This approach not only improved testing efficiency but also optimized resource allocation. The risk-selection criteria in SUPERNOVA are based on factors such as historical defect density, code complexity, and recent code changes. These criteria help prioritize tests toward areas most likely to introduce

bugs, ensuring maximum coverage with minimal effort.^[2]

Test Case Deduplication

As test suites grow, redundant test cases often accumulate. This redundancy leads to inefficiencies and unnecessary resource consumption. AI-driven tools that perform test case deduplication can identify and remove overlapping tests, making the suite more efficient and manageable. Even traditional methodologies such as pairwise test case reduction and combinatorial techniques have been used to tackle redundancy by minimizing the number of test cases while preserving coverage.

How to Implement Test Case Deduplication in Your Project

- Analyze Your Test Suite for Redundancies
- Leverage AI and Machine Learning Tools
- Integrate Deduplication into Your CI/CD Pipeline
- Establish Deduplication Policies
- Monitor and Continuously Improve

In a study by Alastair F. Donaldson, techniques for test-case reduction and deduplication were examined through transformation-based compiler testing. The research highlighted how automation in deduplication effectively eliminated redundant test cases, boosting the efficiency of the software testing process. By integrating AI-powered solutions, organizations can streamline their test suites, reducing unnecessary resource usage while maintaining high standards of software quality.^[3]

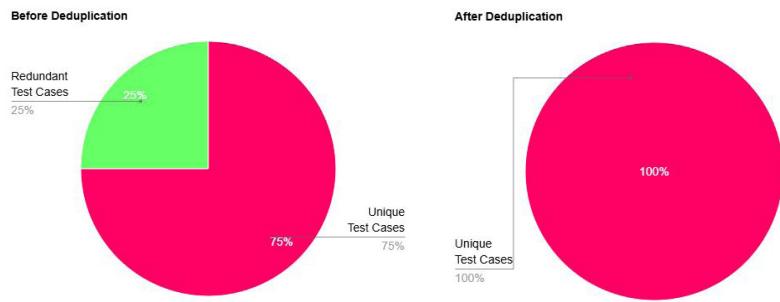


Abbildung 2 und 3: This pie chart is for illustrative purposes only and does not represent real data. It is intended to demonstrate the concept visually. (Quelle: <https://piechartmaker.co/>)

Selective and Incremental Testing

Instead of running the entire test suite after every change, selective testing focuses on only the tests relevant to the code that was modified. Incremental testing further optimizes this by ensuring that only the affected components are tested, reducing unnecessary tests on stable parts of the system.

How to Implement Selective and Incremental Testing in Your Project

- Identify Test Dependencies
- Integrate with Continuous Integration (CI)
- Use Change Detection Tools
- Focus on Key Components and High-Risk Areas
- Optimize Test Suites for Incremental Testing
- Monitor and Adjust Testing Strategies

In a study titled "Testing at the Speed and Scale of Google," Google explains how they use an in-memory graph to efficiently track dependencies between tests and build rules across their entire codebase. This graph is updated with each code change, enabling the company to pinpoint and run only the tests directly impacted by those modifications. By avoiding unnecessary tests, this targeted approach reduces computational resource consumption and speeds up development cycles. Not only does this method improve testing efficiency, but it also makes the testing process more sustainable by minimizing the environmental impact associated

with excessive resource usage in large-scale testing environments.^[4]

Parallel and Cloud-Based Testing

Parallel testing enables multiple tests to run simultaneously across different machines or cloud-based platforms. This reduces testing time significantly by leveraging distributed resources. Cloud platforms also allow for scalability, where additional resources can be provisioned as needed.

How to Implement Parallel and Cloud-Based Testing in Your Project

- Choose the Right Testing Framework
- Use Cloud-Based Testing Platforms
- Integrate Parallel Testing into Your CI/CD Pipeline
- Optimize Test Distribution for Maximum Efficiency
- Monitor and Optimize Test Execution

In a study titled "Cost-benefit evaluation on parallel execution for improving test efficiency in cloud computing," the authors propose a cloud-based approach that partitions large test jobs into smaller tasks and executes them concurrently across multiple virtual machines (VMs). This parallel execution method significantly improves testing efficiency by speeding up test execution times while reducing the overall costs associated with the testing process. Furthermore, by leveraging cloud resources, this approach contributes to

sustainability by optimizing resource usage and reducing the environmental impact of traditional, resource-heavy testing methods.^[5]

Energy-Aware Test Execution

Scheduling tests during off-peak hours or utilizing energy-efficient hardware can significantly lower energy consumption and reduce the environmental footprint of test execution.

How to Implement Energy-Aware Test Execution in Your Project

- Schedule Tests During Low-Demand Periods
- Optimize Test Environments for Energy Efficiency
- Reduce Test Execution Time with Smarter Testing Strategies
- Monitor and Optimize Energy Usage



Abbildung 4: (Quelle: Mariana M Caceres - pixels)

In a study titled „GreenScale: Carbon-Aware Systems for Edge Computing,” researchers developed a framework that schedules applications based on the carbon intensity of energy sources. The study demonstrated that this approach reduced carbon emissions by up to 29.1% for applications in categories such as AI, gaming, and AR/VR.^[6]

Automated Test Optimization

Traditional automated testing often runs predefined test cases without considering their relevance. Modern automated testing tools use Test Impact Analysis (TIA), which assesses code changes and determines the minimal set of tests to execute, thereby avoiding redundant tests.

How to Implement Automated Test Optimization in Your Project

- Integrate AI-Powered Test Impact Analysis (TIA) Tools
- Optimize Test Execution with Code Coverage Analysis
- Automate Test Prioritization with Machine Learning
- Implement Dynamic Test Selection in CI/CD Pipelines

A study titled “SEFGAST: Step-Up to Environment Friendly Green Automated Software Testing” explored how energy consumption in automated testing can be optimized. By utilizing energy measurement tools alongside test case coverage analysis, the study demonstrated how identifying the energy usage of specific hardware components during test execution led to significant reductions in resource consumption. This method not only minimized the energy footprint but also improved testing efficiency by targeting energy-intensive operations.^[7]

Performance and Load Testing Adjustments

While performance and load testing ensure that an application can handle high traffic, full-scale tests can be resource-intensive. Instead, AI-based predictive modeling can simulate user behavior and estimate system performance without running full-scale load tests.

How to Implement Smarter Load Testing in Your Project

- Use AI-Driven Predictive Modeling for Traffic Simulations
- Apply Adaptive Load Testing Strategies
- Implement Machine Learning-Based Anomaly Detection
- Optimize Load Testing in CI/CD Pipelines

A study titled “Performance Testing Using a Smart Reinforcement Learning-Driven Test Agent” developed an intelligent agent using reinforcement learning to optimize load testing. By

automatically generating efficient test workloads, the approach significantly reduced testing costs and improved efficiency by 25% compared to traditional methods. This AI-driven strategy not only enhances testing performance but also aligns with sustainability goals by minimizing resource consumption and reducing the environmental impact of extensive testing.^[8]

The Tangible Benefits of Smart Testing

Adopting smarter testing strategies results in significant benefits for software development teams and businesses alike:

- Reduced computational costs: Eliminating unnecessary test cases leads to lower infrastructure expenses.
- Faster development cycles: Optimized testing reduces execution time, allowing for quicker software releases.
- Lower environmental impact: By minimizing resource consumption, companies contribute to a more sustainable technology ecosystem.
- Improved software quality: With a focus on essential and high-risk test cases, software reliability is maintained or even enhanced.

Conclusion

Software testing is essential, but the way we approach it needs to evolve. By adopting smarter strategies like risk-based testing, AI-driven optimization, and cloud-based execution. Companies can improve efficiency while reducing costs and environmental impact.

Testing isn't just about catching bugs; it's about building sustainable, high-quality software. These modern testing practices allow businesses to prioritize what truly matters, focus resources where they'll have the most impact, and cut down on unnecessary testing and waste. By

integrating these strategies, organizations can not only speed up development cycles but also reduce their carbon footprint and resource consumption.

Businesses that embrace these resource-conscious methods will not only streamline their testing processes but also position themselves as leaders in innovation and sustainability. In the long run, the combination of smarter testing and environmental responsibility will shape the future of the tech industry, helping build software that is both reliable and eco-friendly.

Quellen und weiterführende Informationen:

- [¹] International Energy Agency. (2024). Data Centres and Data Transmission Networks. <https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks>
- [²] SUPERNOVA: Automating Test Selection and Defect Prevention in AAA Video Games Using Risk-Based Testing and Machine Learning" by Senchenko et al., arXiv, 2022. <https://arxiv.org/pdf/2203.05566>

- [³] Donaldson, A. F., et al. (2021). Test-Case Reduction and Deduplication Almost for Free with Transformation-Based Compiler Testing. <https://www.doc.ic.ac.uk/~afd/homepages/papers/pdfs/2021/PLDI.pdf>

- [⁴] Google Testing Blog. (2011). Testing at the Speed and Scale of Google. <https://testing.googleblog.com/2011/06/testing-at-speed-and-scale-of-google.html>

[⁵] Cost-benefit evaluation on parallel execution for improving test efficiency in cloud computing. Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2017. <https://ieeexplore.ieee.org/document/7988384>

[⁶] Kim, Y. G., Gupta, U., McCrabb, A., Son, Y., Bertacco, V., Brooks, D., & Wu, C.-J. (2023). GreenScale: Carbon-Aware Systems for Edge Computing. arXiv preprint arXiv:2304.00404. <https://arxiv.org/abs/2304.00404>

[⁷] SEFGAST: Step-Up to Environment Friendly Green Automated Software Testing. International Journal of Engineering Trends and Technology, 2023. Retrieved from <https://ijettjournal.org/assets/Volume-70/Issue-3/IJETT-V70I3P218.pdf>

[⁸] Helali Moghadam, M., et al. (2021). Performance Testing Using a Smart Reinforcement Learning-Driven Test Agent. arxiv.org/pdf/2104.12893

<https://huddle.eurostarsoftwaretesting.com/common-problems-of-test-redundancy/>

<https://alife-robotics.org/article/vol8issue1/125957076.pdf>



Maged Elsherif ist Consultant bei SEQIS.

Mit einem Hintergrund im Maschinenbau wechselte er in das Gebiet der Softwaretests und begann als Tester für ein Videospieleunternehmen. Diese erste Erfahrung weckte eine Leidenschaft für Softwaretests, die ihn dazu brachte, sie als Vollzeitkarriere zu verfolgen. Er spezialisiert sich auf die technischen Aspekte von Softwaretests, einschließlich Testautomatisierung, API-Tests und Leistungstests. Mit dem Ziel, an der Spitze der Technologie zu bleiben, strebt er kontinuierlich danach, sein Fachwissen zu erweitern und zur Weiterentwicklung des Bereichs der Softwaretests beizutragen.

Test Data Management Meets Sustainability

von Fatemeh Ghaffaari



Abbildung 1: (Quelle: KI-generiert mit Gemini)

The Intersection of Test Data Management and Sustainability

Sustainability—which means „meeting the needs of the present without compromising the ability of future generations to meet their own needs“^[2]—applies directly to software testing. Large-scale software testing consumes significant resources, and without efficient management, it leads to wasted storage, increased energy consumption, and unnecessary duplication of data.

The sustainability dimensions in software testing include:

- Environmental Sustainability: Reducing energy consumption through optimized storage and data lifecycle management.
- Economic Sustainability: Reducing costs through automated test execution and maintenance.
- Social Sustainability: Ensuring ethical data use, employee well-being, and social responsibility.
- Technical Sustainability: Using long-term test automation solutions that evolve with technological advancements.

Common Challenges:

- Over-provisioning: Businesses give test environments excessive storage, the majority of which is never utilized. The result is a waste of energy and higher hardware demand.
- Data Waste: Ineffective data-aging procedures result in the buildup of test data that is

redundant, irrelevant, or out-of-date, which clogs storage systems and increases maintenance costs.

- Manual Processes: In the absence of automation, test data generation procedures are not always consistent, which can result in inefficient replication of previously produced content.

Best Practices for Sustainable Test Data Management

Smart Generation of Test Data

- Synthetic test data: Synthetic test data can be used rather than sensitive production data. This is privacy-compliant and even more resource-efficient. Various synthetic data generation tools exist that create quite realistic datasets for specific testing scenarios without exposing real user data. Examples of such tools include:
 - > MOSTLY AI: Known for generating highly accurate synthetic data that preserves the statistical properties of the original data.
 - > Tonic.ai: Offers solutions for generating realistic, de-identified test data.
 - > Gretel.ai: Provides a platform and APIs for creating privacy-preserving synthetic data.
 - > Synthetic Data Vault (SDV): An open-source library for generating synthetic data for various data types.
- Needs-Based Generation: Test data is generated precisely for the needs of the test scenarios without generating and storing any data that may not be required. Identifying key test scenarios and aligning data generation efforts with them reduces overhead.

- Data Masking: When production-like data is needed, anonymization or pseudonymization ensures that data privacy is

maintained. Masked data still retains the core characteristics necessary for testing while removing sensitive information.

Efficient Usage of Storage

- Data compression: Compressed test data requires less storage and reduces data transfer times. Modern compression algorithms can significantly reduce file sizes without compromising data integrity. Examples of lossless compression algorithms commonly used in data storage include LZ77 and LZ78 (which are the basis for formats like ZIP and GZIP) and Run-Length Encoding (RLE), which is effective for data with repeating sequences. Techniques like dictionary encoding and columnar storage, often implemented in modern databases, also contribute to efficient compression by reducing redundancy within data sets.
- Deduplication: Duplicates in data should be identified and removed to save storage space. Duplication tools study stored data for finding identical segments so that no duplicate copies of data are created.
- Intelligent Archiving: If test results are no longer relevant, they can be archived or removed. Such archiving solutions, integrated with data lifecycle management frameworks, can automate this process and ensure compliance with data retention policies and industry regulations.

Automation and Tools

This is how automated TDM solutions can enable better utilization of resources. Test data generation and management tools can optimize processes, reduce manual effort,

and ensure consistency of data. For example, AI-powered solutions can predict data needs and adjust storage allocation dynamically for even higher efficiency.

Regulatory and Ethical Considerations
TDM practices should be aligned with regulatory and ethical requirements. Data privacy laws like GDPR and HIPAA have put immense pressure on handling sensitive information with care. Integrating compliance into the workflow of TDM will prevent penalties while fostering sustainability.

GDPR (General Data Protection Regulation): A regulation enforced in the European Union that governs data privacy and security. It mandates strict guidelines on collecting, storing, and processing personal data, ensuring that organizations maintain transparency and accountability. Under GDPR, individuals have rights over their data, including the right to access, rectify, and erase their personal information.^[5]

HIPAA (Health Insurance Portability and Accountability Act): An U.S. law that establishes national standards for protecting sensitive patient health information. HIPAA ensures that healthcare providers, insurers, and business associates maintain the confidentiality and security of medical data. It requires safeguards for electronic health records and imposes strict penalties for breaches.^[6]

In addition, test data creation and usage should be based on ethical principles such as fairness, transparency, and accountability. Organizations should avoid letting synthetic data introduce bias or inaccuracies that may affect the performance of the software.

Sustainability Data Strategy: Key Insights

A well-defined sustainability data strategy involves^[3]:

- Factual Data, Not Estimates: Relying on precise, audit-ready sustainability data rather than approximations ensures accuracy and reliability. Many organizations currently rely on sector averages, which are often imprecise and fail to provide actionable insights for reducing environmental impact.
- Transparency in Net Zero Motives: Organizations must disclose actual emissions and offsets purchased, aligning financial data with sustainability goals. The rising costs of carbon offsets make it crucial to provide clear, data-backed sustainability strategies.
- Operational Improvements and Long-Term Value Creation: High-quality sustainability data helps reduce carbon emissions, minimize back-office expenses, and improve decision-making processes. By integrating sustainability metrics with financial data, companies can track the effectiveness of green initiatives.
- Audit-Ready and Proficient Data: The increasing scrutiny on ESG data means organizations must ensure their sustainability reporting aligns with regulatory expectations and can withstand audits. Consistent and structured data management supports compliance and enhances credibility.

ESG (Environmental, Social, and Governance) refers to a set of criteria used by organizations, investors, and regulators to assess a company's commitment to sustainability and ethical business practices. ESG data plays a crucial role in sustainability reporting and decision-making. Companies are expected to disclose their environmental impact, social responsibility efforts, and governance practices. However, challenges such as data standardization, regulatory compliance, and audit readiness make ESG reporting

complex. A robust sustainability data strategy ensures accurate, transparent, and audit-ready ESG reporting, helping organizations align their sustainability initiatives with business goals.^[7]

Organizations must also master challenges in ESG data reporting, standardization, and regulatory compliance to enhance sustainability initiatives.

Sustainable Software Testing in Practice

Software testing has a significant impact on sustainability, with long-term benefits often overlooked^[2]. The key aspects of sustainable software testing include:

- Minimizing Environmental Impact: Sustainable software testing reduces unnecessary computing resource consumption, leading to lower energy use and reduced carbon footprints. This includes optimizing test execution schedules to run during off-peak hours and leveraging cloud environments with renewable energy sources.
- Enhancing Economic Sustainability: Test automation reduces operational costs by streamlining repetitive testing tasks. Companies implementing automated, energy-efficient testing strategies benefit from reduced hardware and cloud expenditures.
- Improving Technical Sustainability: Implementing scalable and maintainable test frameworks ensures that software testing processes remain efficient and adaptable to future technological advancements.
- Encouraging Knowledge-Sharing and Collaboration: Sustainable testing practices involve knowledge transfer and continuous improvements, helping organizations develop resilient and future-ready test strategies.

Benefits of Sustainable TDM

The advantages of a sustainable approach to TDM include the following:

- **Cost Savings:** Reduced storage needs lower hardware and cloud costs. Optimized test data usage decreases infrastructure expenses while improving efficiency.
- **Eco-Friendliness:** Optimized data management results in lower energy consumption and a reduced carbon footprint. Green IT initiatives such as efficient data storage and retrieval mechanisms contribute to environmental sustainability.
- **Data Security:** Synthetic and well-managed test data minimizes the risk of data privacy breaches. Organizations can maintain compliance with data protection regulations while reducing redundant storage.
- **Operational Efficiency:** Streamlined TDM processes improve overall development efficiency, allowing teams to focus on innovation. Sustainable TDM supports faster software release cycles and enhances quality assurance practices.

The Future of Test Data Management in a Greener IT Landscape

Emerging trends such as green computing and carbon-neutral cloud services are reshaping the IT industry. Companies are increasingly adopting environmentally friendly technologies and practices, including renewable energy-powered data centers and energy-efficient storage solutions. AI and machine learning developments really drive innovation in the field of TDM. These technologies enable predictive analytics, real-time optimization, and adaptive resource allocation, adding to sustainability. For example, AI can find patterns in test data usage and suggest optimizations that reduce waste. However, it is important to acknowledge that these technologies, particularly in their training and operation, can also

be a source of significant energy consumption.

Conclusion

Test data management and sustainability go hand in hand. Optimizing TDM processes enables companies to save resources and, at the same time, make their contribution to environmental protection. This is the ideal moment to adopt more sustainable solutions and take a new look at your current procedures. Seize this

opportunity to implement efficient generation and management strategies for test data now—for a greener future in IT.

Sustainable TDM practices will bring balance to the needs of an organization between sustainability and environmental responsibility. It all begins with a few simple, achievable actions that add up to a significant impact. We should together build up the path towards sustainability for the IT industry and beyond.

Quellen und weiterführende Informationen:

- [¹] Dror Etzion and J. Alberto Aragon-Correa, „Big Data, Management, and Sustainability: Strategic Opportunities Ahead,” May 10, 2016, Volume 29, Issue 2.
- [²] Beer, A., Felderer, M., Lorey, T., & Mohacsi, S. (2021). „Sustainability in the Test Process.” 1st International Workshop on the Body of Knowledge for Software Sustainability (BoKSS), IEEE ICSE Conference.
- [³] Clusters Media Technology. (2022). „Sustainability Data Strategy: Top Key Components for a Positive Impact.”
- [⁴] Jim Soos, Planckton Data Technologies. (2023). „Sustainability Data Management.”
- [⁵] Official text of the General Data Protection Regulation (GDPR): <https://gdpr-info.eu/>
- [⁶] More information about the Health Insurance Portability and Accountability Act (HIPAA): <https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/combined-regulation-text/index.html>
- [⁷] Learn more about Environmental, Social, and Governance (ESG): <https://www.investopedia.com/terms/e/environmental-social-and-governance-esg-criteria.asp>



Fatemeh Ghaffaari ist Consultant bei SEQIS.

Mit über einem Jahrzehnt Erfahrung in der Qualitätssicherung, insbesondere im Bank- und IT-Dienstleistungsbereich, bringt sie beeindruckende Expertise mit. Ihre Leidenschaft gilt der Sicherstellung der Qualität von Webanwendungen und APIs, wo sie täglich komplexe Herausforderungen meistert. Mit einem Masterabschluss in Physik und einer Spezialisierung auf Testplanung, -analyse, funktionale Tests und Defect-Management zeichnet sie sich durch analytisches Denken und die Fähigkeit aus, innovative Lösungen im Team einzubringen.

Es ist Webinar-Donnerstag!



Ten more **things**



- Praxisbezogene Fachvorträge zu aktuellen IT-Trendthemen
- Kompetentes, fachliches Know-how aus der Praxis
- 10 Tipps & Tricks für Ihren Arbeitsalltag
- Kostenlose Teilnahme
- Online

Termine

- 25. September 2025: Kommunikation und Knowledge Management in IT-Projekten
- 13. November 2025: Kampfansage an technische Schulden – Ursachen, Auswirkungen und Strategien im Projekt- und Change Management
- 16. April 2026: Thema tbd.
- 11. Juni 2026: Thema tbd.
- 24. September 2026: Thema tbd.
- 12. November 2026: Thema tbd.

Informiert bleiben



www.seqis.com/events-index

Referenzstory: INVARIS Informationssysteme GmbH

mit Martin Schöllerl



INVARIS
LEADING DOCUMENT SOLUTIONS

„Durch die Empfehlungen von SEQIS zu Methoden und Werkzeugen konnten wir unserem Ziel, die kontinuierliche Qualität unserer Software in einem agilen Umfeld sicherzustellen, deutlich näherkommen.“

Martin Schöllerl, Head of Development,
INVARIS Informationssysteme GmbH

INVARIS Informationssysteme GmbH ist ein international führender Hersteller von Software-Plattformen für interaktive Formular- bzw. Dokumentenerstellung und -bearbeitung. Die Spezialität von INVARIS sind hochvolumige Dokumentenerzeugung, automatisierte Textverarbeitung und intelligentes Output Management. Seit 1986 setzt das Unternehmen Maßstäbe im Bereich Document Intelligence. Mit über 30 Jahren Erfahrung und stetigem Blick in die Zukunft kann INVARIS seinen Kunden genau die Produkte anbieten, die sie benötigen.

Qualität: Höchste Qualität bei jedem Arbeitsschritt - das gilt nicht nur für die Produkte, sondern auch für die Implementierung, den Service und die Wartung der Lösungen.

Das Gespräch

- Martin Schöllerl, Head of Development
- Sandra Benseler, SEQIS Sales Managerin

Individuell abgestimmte Lösungen statt allgemeiner Tipps

Sandra Benseler: Wie dringend war es, in Ihrem Unternehmen Veränderungen am Entwicklungsprozess vorzunehmen? Oder anders gefragt: Gab es einen Zeitpunkt, an dem deutlich wurde, dass Handlungsbedarf besteht?

Martin Schöllerl: Ja, die Notwendigkeit einer gezielten Optimierung war sehr deutlich geworden. Die Komplexität unserer Softwareprodukte der goSmart®-Suite hat in den letzten Jahren stark zugenommen. Dies hatte zur Folge, dass unsere Software sehr präzise weiterentwickelt werden musste, um sicherzustellen, dass sie auch nach Anpassungen und Erweiterungen noch wie gewünscht funktioniert.

Zum anderen ist unser Team, das für die Installation und Betreuung der Software beim Kunden verantwortlich ist, gewachsen. Auch damit wurde die Zuverlässigkeit der Funktionalitäten auf die Probe gestellt. Insbesondere bei individuellen Systemanbindungen. Die größte Herausforderung war dabei das manuelle Testen bei gleichzeitig steigender Komplexität.

Es kam dazu, dass Fehler erst spät entdeckt wurden. Zudem hatten wir es mit Legacy-Software zu tun, die teilweise 20 Jahre alt und monolithisch gewachsen ist.

In der Vergangenheit zögerten Entwicklerinnen und Entwickler oft, Änderungen am Code vorzunehmen, weil sie sich unsicher fühlten. Interne Abhängigkeiten und die Komplexität der Produkte erhöhten das Risiko, unbeabsichtigte Probleme zu

verursachen. Diese Unsicherheit machte die Entwicklungsarbeit weniger angenehm, da jede Änderung mit einem gewissen Bauchweh verbunden war.

Der Wunsch, von manuellen zu automatisierten Testverfahren überzugehen, zeigte uns, dass wir nicht genau wussten, wie und wo wir anfangen sollten. Unser Ziel war es daher, ein Sicherheitsnetz zu schaffen - sowohl für die Entwicklerinnen und Entwickler als auch für die Teams, die die Produkte beim Kunden betreuen. Damit einhergehend wird auch die Kundenzufriedenheit erhöht.

Sandra Benseler: Wie konnte unser Agile Quality Coach Sie dabei am besten unterstützen?

Martin Schöllerl: Auf vielfältige Weise. Zunächst wurden umfassende Analysen durchgeführt, die durch persönliche Interviews mit allen relevanten Stakeholdern ergänzt wurden. Dazu gehörte auch eine sorgfältige Auswahl der Interviewpartner, um alle wichtigen Perspektiven abzudecken. Die gewonnenen Erkenntnisse wurden anschließend zusammengefasst und bildeten die Grundlage für die Ableitung von Maßnahmen. Diese Maßnahmen zielen darauf ab, Optimierungen umzusetzen und

die Strategie zu verbessern. Die Ergebnisse und Empfehlungen wurden uns vor Ort vom Agile Quality Coach präsentiert, um alle Beteiligten auf den gleichen Stand zu bringen. Ein besonderer Fokus lag darauf, das Thema Unit Tests zu stärken und weiter voranzutreiben.

Sandra Benseler: Welche konkreten Verbesserungen konnten Sie nach Umsetzung der empfohlenen Maßnahmen feststellen? Können Sie ein konkretes Beispiel nennen, das diese Veränderungen verdeutlicht?

Martin Schöllerl: Ja, die Maßnahmen, die wir ergriffen haben, zeigten bereits sehr rasch positive Ergebnisse. Ein Beispiel ist ein Kunde, bei dem wir in letzter Zeit mehrere Updates durchgeführt haben. Vor Einbindung des Agile Quality Coaches mussten wir den Update-Termin mehrfach verschieben, da die Qualität nicht ausreichte, und die Zeitpläne durcheinander geraten sind.

Mit Beginn der Umsetzung von Maßnahmen, die uns der Agile Quality Coach aufgezeigt hatte, verliefen Update-Termine kontinuierlich besser und beim letzten Update verlief alles termingerecht und ohne Verzögerungen. Und auch die Anzahl der Fehler, die während der Testphase entdeckt wurden, ist kontinuierlich zurückgegangen.

Diese Entwicklungen sind ein klares Zeichen dafür, dass unsere Maßnahmen Wirkung zeigen. Der Prozess ist zwar langwierig und es gibt noch viele Baustellen, aber die Fortschritte sind sichtbar. Es geht definitiv voran.

Sandra Benseler: Aus welchen Gründen haben Sie sich für die Unterstützung durch SEQIS entschieden?

Martin Schöllerl: Wir haben am Anfang auch andere Anbieter kontaktiert. Bereits in den ersten

Gesprächen wurde uns bei den meisten Anbietern klar, dass keine individuellen Lösungen zu erwarten sind. Zudem fehlte der Workshopcharakter. Wir wollten eine für uns maßgeschneiderte Lösung, erstellt anhand einer detaillierten Analyse und der Möglichkeit, dass genau auf unsere speziellen Problemstellungen eingegangen wird. Von einer 0815 Lösung, mit der wir am Ende bei der Umsetzung wieder da stehen, wo wir am Anfang standen, wollten wir ja weg kommen.

Wir wollten gezielte Maßnahmen vorgeschlagen bekommen, bei deren Umsetzung wir auch Schritt für Schritt begleitet werden, falls nötig. Bei SEQIS haben wir genau das erhalten, was unseren Erwartungen weit besser entsprach. Daher haben wir uns relativ schnell intern für SEQIS entschieden.

Sandra Benseler: Gab es neben der Unterstützung durch den Agile Quality Coach weiteren Bedarf, den SEQIS abdecken konnte?

Martin Schöllerl: Ja, im Bereich Unit Tests, als eine vorgeschlagene Maßnahme. Nach dem Analyse-Workshop und den Präsentationen, in denen auch erklärt wurde, wie man Unit Tests erstellt, kam diese zweite Phase, die gezielte Begleitung durch euch. Ein SEQIS Experte unterstützte unsere Kollegen und Kolleginnen dabei, notwendige Maßnahmen selbst umzusetzen. Anschließend wurden die Ergebnisse von euch wiederum überprüft.

Bis auf ein paar kleine Punkte hatte der SEQIS Experte dann keine Beanstandungen oder weitere Verbesserungsvorschläge. Das Coaching hat also offenbar genau das bewirkt, was wir uns erhofft hatten: Die Mitarbeiter:innen haben die Konzepte verstanden und wenden die Best Practices erfolgreich an. Insgesamt zeigt dies, dass wir mit SEQIS die richtige Wahl getroffen haben.

Sandra Benseler: Mit welchem Argument würden Sie IT-Verantwortliche davon überzeugen, dass ein Agile Quality Coaching eine gewinnbringende Unterstützung ist?

Martin Schöllerl: Der entscheidende Vorteil von einem Agile Quality Coaching ist der individuelle Ansatz. Anstatt nach einem standardisierten Schema vorzugehen, haben wir maßgeschneiderte Lösungsvorschläge und konkrete Maßnahmen erhalten, die direkt auf unser Unternehmen und unsere Prozesse abgestimmt waren. Das war für uns besonders wertvoll, weil wir so schnell Lösungen umsetzen konnten.

Natürlich kann man sich auch im Internet viele Informationen und allgemeine Ratschläge holen. Doch die wirkliche Herausforderung liegt darin, diese Informationen gezielt auf die spezifischen Bedürfnisse und Prozesse unseres Unternehmens anzuwenden. Das ist genau das, was wir durch eure Unterstützung erhalten haben: maßgeschneiderte Lösungen statt allgemeiner Tipps.

Zusätzlich hatten wir das Glück, zur gleichen Zeit einen neuen Mitarbeiter einzustellen, der sich ausschließlich mit diesen Themen beschäftigt. Das ist ein wichtiger Punkt: Man muss ausreichend Zeit und Ressourcen einplanen, um solche Projekte erfolgreich umzusetzen. Es ist nicht möglich, dies einfach nebenbei zu erledigen.

Was uns besonders gefallen hat, war das Gefühl, dass ihr euch intensiv mit unseren spezifischen Bedürfnissen auseinandergesetzt und uns nicht nur allgemeine Ratschläge gegeben habt. Das war der Grund, warum wir uns für euch entschieden haben.

Sandra Benseler: Vielen Dank für das schöne Feedback und die Gelegenheit der erfolgreichen Zusammenarbeit mit Ihnen und Ihrem tollen Team!

razzfazz.io pflanzt Bäume

von Timo Keber



Abbildung 1: Unser erster Baum (Quelle: <https://tree-nation.com/de/profil/razzfazzio>)

Als Entwickler:in verwendet man täglich unterschiedlichste Software, wahrscheinlich mehr als die meisten anderen Nutzer:innen. Durch das Erleben zweier Perspektiven – Nutzung und Entwicklung – erlebt man auch mehr als bei dem Rest der Verwender, das Stoßen auf Probleme oder schlicht Ärgernisse, die einem die Benutzung erschweren.

Die Kehrseite davon ist, dass man auch aktiv Einfluss darauf nehmen kann und die Software, die man verwendet, analysieren und verbessern kann. Nichts ist ein besseres Beispiel davon, als unser hauseigener SEQITracker*, den wir alle verwenden und für dessen Umsetzung und Wartung wir verantwortlich sind. Wenn wir Schwierigkeiten und Unstimmigkeiten bei der Nutzung entdecken, oder rückgemeldet bekommen, setzen wir uns zusammen und arbeiten daran.

Ähnlich ist es für die VS Code Extension: GitLab Workflow^[1] abgelaufen. Die Integration von GitLab Merge Requests^[2] in unseren Standardablauf hat bedeutet, dass wir auch ohne Context-Switching aus der IDE weiter Code Reviews durchführen wollten. Dem hat GitLab mit der Entwicklung der eigenen VS Code Extension abgeholfen, die einen Großteil von Features des Web-Uls abdeckt. Leider wurde aber nie eine vollständige Fea-

ture-Parity erreicht und besonders das Verfassen von Reviews aus den VS Code Webviews ist nicht vergleichbar mit dem Komfort und den Möglichkeiten des Web Äquivalents.

Die Recherche zu Alternativen oder Lösungen blieb fruchtlos; so entstand die Idee, eine zeitliche begrenzte Entwicklung eines der fehlenden Features selber umzusetzen zu versuchen. Die Einsicht, wie stark die andauern-de Nachfrage nach spezifisch dieser Funktionalität ist, hat dieses Commitment nur bestärkt.

Problem

Die Ausgangslage ist in bestehenden Issues bereits ausführlich diskutiert, mein Fokus war beschränkt auf:

Review und Draft-Notes Feature für GL Workflow

- Unterstützung für Draft Notes, die in GitLab als Entwürfe gespeichert werden können
- Darstellung und visuelle Unterscheidung dieser in der Code-Review Ansicht

Für die Umsetzung war es erforderlich, den bislang ausschließlich auf die Verarbeitung von GraphQL-Objekten ausgelegten Code zu erweitern – wie in den folgenden Punkten näher beschrieben.

Optimierung des API-Handleings

- Einführung einer diskriminierten Union zwischen Nachrichten der API, die als GQL und JSON übermittelt werden
- Konditionelles Abhandeln dieser in UI

Die genauen Anforderungen und Schwierigkeiten haben sich im Laufe der Diskussion mit den Maintainern ergeben.

Strategie

Der Großteil des Entwicklungsauf-wands bestand aus der Analyse der bestehenden Codebasis, sowie der GitLab-API und der Anbindung an diese. GitLab verfolgt dabei eine hybride Architektur und bietet eine klassische nach OpenAPI spezifizierte REST API, sowie auch eine versionslose GRAPHQL API, an. Für die Umsetzung ist diese Unterscheidung relevant ge-worden: Die Draft Notes werden nur von der REST API unterstützt.

Initial war mein Ansatz, nicht abge-gene Kommentare (= Draft Notes) lokal, in Memory zu halten und nach entsprechender Interaktion des Users zusammen (= in Batch) an die API zu posten. Der erste Stand meiner Entwicklung, der noch diese Strate-gie verfolgte und als PoC von dem verantwortlichen GitLab Maintainer unter die Lupe genommen wurde, war zu komplex und zu umständlich, um so übernommen zu werden. Nach Rücksprache wurde meine andere Konzeption bevorzugt: einen „API-First“ Approach. Die Verwendung der Draft Notes API hilft dabei, nicht für jeden einzeln veröffentlichten Kommentar Benachrichtigungen für alle an der Review beteiligten Nutzer auszulösen.

Wie bereits oben bei der Problem-stellung erwähnt, hat die Lösung mit Draft Notes zu Folge, dass der gesamte Code für das Fetchen, Kon-vertieren und Darstellen von Kom-men-taren und Diskussion für Draft Notes angepasst werden muss. So ist nun der jetzige Status der Um-setzung. Nach dem Fertigstellen der Unterscheidung und meiner Anfrage wird ein Ablauf losgetreten, in dem mein Beitrag aus Code und UX Pers-pektiven evaluiert wird.

Kai Armstrong @phikai · 2 days ago

That's unfortunate. @francoisrose @marc_shaw I assume we are not planning on bringing the draft notes API to GraphQL soon right?

@viktoras @timokz Probably not... at least not with the current work we're doing. It may be something that arises out of the Rapid Diff efforts, but I think for compatibility reasons we're trying to keep the bulk of this the same.

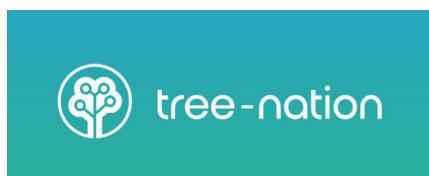
1 1

Abbildung 2: Auch ist es nicht geplant Draft Notes über die GraphQL API anzubieten (Quelle: https://gitlab.com/gitlab-org/gitlab-vscode-extension/-/merge_requests/2390)

Code, der wächst

Zum springenden Punkt, wie das bisher diskutierte nun Nachhaltigkeit bedeutet und was es mit den Bäumen auf sich hat:

Indem man Features verbessert, Bugs fixed oder an neuen Funktionen arbeitet, sammelt man automatisch Punkte. Anstatt sie für Merchandise oder andere Prämien zu nutzen, gibt es die Möglichkeit, sie in Nachhaltigkeit zu investieren – konkret in das Pflanzen von Bäumen.



Home > Plant a Tree with Tree-Nation

Plant a Tree with Tree-Nation

With this donation GitLab will plant a tree in their growing Tree-Nation Forest. You will receive a tracking link to keep track of your Tree and watch it grow.

5 points

Quantity

Abbildung 3: Kaufoption der Bäume mit Contributor Points (Quelle: <https://rewards.gitlab.com/shop/37706-tree-nation-donation>)

Zum jetzigen Zeitpunkt ist nur ein einzelner Baum in unseren Namen bestellt, aber mit der Fertigstellung und Mergen der finalen Kontribution sollten ein Minimum von 20 Bäumen mehr durch uns gedeihen dürfen. Weitere Updates dazu gerne in Kürze!

Nachhaltige Softwareentwicklung?

Besonders durch den enormen Energieverbrauch, der mit der zunehmenden Verwendung von generativer AI verbunden zu sein wirkt^[3] stellt sich

die Frage, welche Verantwortung wir als Entwickler:innen sowie Unternehmen im Technologiesektor haben. Bereits früher in meiner Karriere durfte ich mich mit der Energienutzung verschiedener Programmiersprachen auseinandersetzen.^[4]

Auch, wenn die Methodik dieser Messungen umstritten ist, ist es nicht gerade erfreulich, die eigenen Lieblingssprachen in den oberen Rängen des Verbrauchs zu sehen.

Fazit & Caveat

Spannend wird die Weiterentwicklung und der Verlauf des Merge Request, ich freue mich schon sehr auf den Zeitpunkt der Fertigstellung und besonders auf die Verwendung des neuen Features.

Natürlich kann der tatsächliche Energieverbrauch, der durch die Entwicklung und den Betrieb dieser Features entsteht, nicht durch das Pflanzen weniger Bäume aufgewogen werden. Doch es geht darum, Bewusstsein zu schaffen, nachhaltige Praktiken zu



fördern und zumindest einen kleinen Beitrag zur Kompensation zu leisten.

Vielleicht ist es Zeit für mehr Entwickler:innen, Unternehmen und andere Organisationen, Open Source nicht nur als technischen Vorteil zu sehen, sondern auch als Möglichkeit, aktiv zu einer besseren Welt beizutragen. Eine Merge Request nach dem anderen – und ein Baum nach dem anderen.

Quellen und weiterführende Informationen:

[1] GitLab Workflow Extension https://docs.gitlab.com/editor_extensions/visual_studio_code/

[2] Merge Requests https://docs.gitlab.com/user/project/merge_requests/

[3] <https://www.sciencedirect.com/science/article/pii/S2542435123003653>

[4] <https://dl.acm.org/doi/10.1145/3136014.3136031>

Relevante Issues zu dem Batch-Feature <https://gitlab.com/groups/gitlab-org/-/epics/7362>; <https://gitlab.com/gitlab-org/gitlab-foss/-/issues/3364>; <https://gitlab.com/gitlab-org/gitlab-vscode-extension/-/issues/526>

*SEQITracker: <https://www.seqis.com/de/blog/moco-und-seqitracker>

Timo Keber ist Agile Development Wizard.

Er kann auf mehrjährige Erfahrung in der Entwicklung von Multiplatform-Apps mit Dart und Flutter sowie in der Systemadministration zurückblicken. Zu seinen fachlichen Schwerpunkten zählen außerdem die Full-Stack-Entwicklung komplexer Anwendungen und die Arbeit in agilen Softwareprojekten. In Ausbildung und Freizeit hat er dabei einen besonderen Fokus auf REST-Alternativen und Open-Source-Software gelegt.

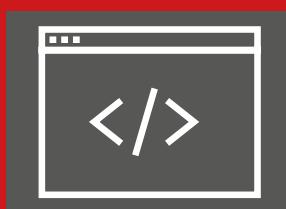
SEQIS ist der führende österreichische Anbieter in den Spezialbereichen
IT Analyse, Development, Softwaretest und Projektmanagement.

Beratung, Verstärkung und Ausbildung:
Ihr Partner für hochwertige IT-Qualitätssicherung.



IT ANALYSE

Notwendige Änderungen analysieren und IT-gerecht aufbereiten



DEVELOPMENT

Agil, individuell und qualitätsgesichert



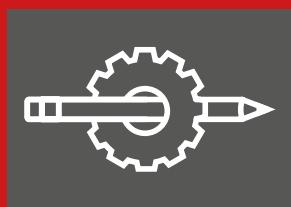
TESTING

Probleme durch methodischen Soll-Ist-Vergleich erkennen



RELEASE & OPERATE

Reibungsloser Go Live und Betrieb der IT-Lösungen



DEVOPS

Neuerungen abgestimmt mit Entwicklung und Betrieb live setzen



METHODOLOGY & TOOLS

Vorgehensweisen optimieren und auf die richtigen Tools setzen



TRAINING & WORKSHOPS

Mitarbeiter Know-how stärken – standardisiert oder maßgeschneidert



PROJEKT- MANAGEMENT

verantwortlich, zielorientiert und pragmatisch