

# SE IS

## QualityNews

Analysis. Development. Test. Management

Ausgabe H1/2024

## IT-Modernisierung

**Shift Left Testing - oder wie das Testen endlich wieder dort angelangt ist, wo es eigentlich herkam**

Seite 4

**Container based hosting: sicher nicht ohne LPT!**

Seite 18

**Test Automation: Paving the Way for IT Modernization**

Seite 36

Bild: Lebenshilfe Niederösterreich

**Analysis. Development. Test. Management. Better Results.**

# Lesen Sie in dieser Ausgabe:

|  |           |  |           |
|--|-----------|--|-----------|
| <b>Editorial.....</b>  | <b>3</b>  | <b>Cheat Sheet - Knowhow für Meetings .....</b>  | <b>20</b> |
|  |           | SEQIS  |           |
| <b>Shift Left Testing - oder wie das Testen endlich wieder dort angekommen ist, wo es eigentlich herkam.....</b> | <b>4</b>  | <b>SEQIS Kalender.....</b>   | <b>22</b> |
| Alexander Vukovic  |           | Alle Termine auf einem Blick   |           |
| <b>Referenzstory: ZETA GmbH.....</b>   | <b>7</b>  | <b>IT-Analyse II - Eine Bestandsaufnahme.....</b>  | <b>24</b> |
| mit Georg Kotsis   |           | Josef Falk   |           |
| <b>Nachhaltigkeit in der Software Entwicklung.....</b>   | <b>8</b>  | <b>Der EU AI Act - die KI-Verordnung der Europäischen Union zur Regulierung von künstlicher Intelligenz.....</b> | <b>28</b> |
| Klemens Loschy   |           | Alexander Lewisch  |           |
| <b>Codeium - der einzig sinnvolle Code Assistent?.....</b>   | <b>10</b> | <b>5 Fragen an Hansjörg Münster ....</b>   | <b>32</b> |
| Daniel Kleissl   |           | Hansjörg Münster   |           |
| <b>Referenzstory: JAGGAER Austria GmbH.....</b>  | <b>16</b> | <b>Test Automation: Paving the Way for IT Modernization .....</b>  | <b>36</b> |
| mit Martin Kersch  |           | Maged Elsharif   |           |
| <b>Container based hosting: sicher nicht ohne LPT!.....</b>  | <b>18</b> |  |           |
| Klemens Loschy   |           |  |           |

## 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](http://www.SEQIS.com/de/blog-index)) oder als Leserbrief. Wenn Sie dieses Magazin abbestellen möchten, senden Sie bitte ein Mail an [marketing@SEQIS.com](mailto:marketing@SEQIS.com).

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

Herausgeber: SEQIS GmbH,  
Neusiedler Straße 36, A-2340 Mödling  
Tel: +43 2236 320 320 0  
[info@SEQIS.com](mailto:info@SEQIS.com), [www.SEQIS.com](http://www.SEQIS.com)  
Gericht: Bezirksgericht Mödling  
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.

### Ihre Meinung ist gefragt!

Nach den QualityNews ist bekanntlich vor den QualityNews! Schon bald arbeiten wir wieder auf Hochtouren an der nächsten, spannenden Ausgabe. Lesen Sie nur das, was Sie wirklich interessiert! Sagen Sie uns, welche Themen Sie spannend finden.

Kontaktieren Sie uns: [marketing@SEQIS.com](mailto:marketing@SEQIS.com)  
Wir freuen uns auf Ihre Vorschläge und Wünsche!



DI Reinhard Salomon

Mag. (FH) Alexander Vukovic

Mag. (FH) Alexander Weichselberger

# Editorial

Sehr geehrte Leserin,  
sehr geehrter Leser,

wir freuen uns, Ihnen die Ausgabe  
für das erste Halbjahr 2024 zu  
präsentieren.

Vielen Dank für das positive  
Feedback zu unserer letzten Ausgabe  
zu dem Themenbereich „Artificial  
Intelligence & Augmented Reality“.  
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 „IT-Modernisierung“ und  
bietet eine Auswahl von Fachartikeln  
mit diesem Schwerpunkt, die wir  
aus unserem [blog.seqis.com](https://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



# Shift Left Testing - oder wie das Testen endlich wieder dort angelangt ist, wo es eigentlich herkam...

von Alexander Vukovic

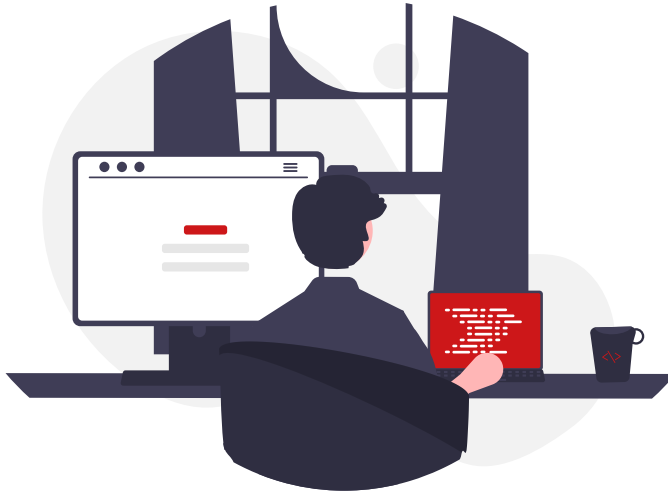


Abbildung 1: (Quelle: undraw.co)

## Die Kunst des Softwaretestens

1979 veröffentlichte ein gewisser Glenford J. Meyers ein Buch, das die Welt der Softwareentwicklung langfristig revolutionieren sollte: „The Art of Software Testing“. Bis zu diesem Zeitpunkt war Softwareentwicklung einfach Software Entwicklung. Testen war Teil der Softwareentwicklung, genauso wie das Stanzen von Lochkarten, Schreiben von Code, Zeichnen von Architekturdiagrammen etc. Doch Meyers stellt einige für damals sehr kontroverse Thesen auf, wie z. B.:

1. Testen ist das Ausführen der Software mit dem Ziel, Fehler zu finden
2. Testen kann daher nur von Personen ordentlich gemacht werden, die nicht selbst der/die Entwickler sind
3. Testen ist daher eine eigene wichtige Disziplin, gleichwertig auf der Ebene der Softwareentwicklung und bedarf eigener Methoden und Experten

Interessanterweise unterstreicht Meyers in der dritten Auflage von 2011 diese Thesen. Sie haben in den 1990er und 2000er Jahren dazu

geführt, dass einem ordentlichen Entwicklungsteam ein ordentliches dediziertes Testteam gegenübergestellt wurde. Das Berufsbild des Testers hat sich entwickelt, die Zusammenarbeit zwischen Entwicklungsteam und Testteam musste unter Zuhilfenahme von Süßigkeiten immer wieder neu aufgebaut werden und weil es keine andere Möglichkeit zur Automatisierung gab, wurde alles über die Oberfläche automatisiert.

Ohne diese bewegten Zeiten wüssten wir heute nicht: „Das Konzept hat nicht funktioniert“!

- Die Zusammenarbeit zwischen Entwicklungs- und Testteam war immer schwierig
- Entweder testeten die Entwickler gar nicht mehr selbst, „das macht ja eh das Testteam“
- Oder die Entwickler fühlten sich kontrolliert/bevormundet von den „Profi“-Testern, die vom Entwickeln im Regenfall keine Ahnung hatten
- Das Testteam fungierte oft als Projektbremse: „so lange wir das nicht „fertig“ getestet haben, können wir nicht ausliefern“

- Oder das Testteam erstellte Unmengen von Bugs - ohne dabei die wirklich kritischen Bugs zu finden

Diese Erfahrung mussten viele Unternehmen machen, die z. B. ihr Testing an Crowdfundingplattformen auslagerten und ihre Tester pro Bug bezahlten. Sie bekamen viele Bugs. 😊

Wobei, es lag nicht nur am getrennten Testteam. Die ganze Softwareentwicklung hat sich in den 1990ern und 2000ern in die falsche Richtung entwickelt. Alle propagierten die industrielle Softwareentwicklung in Anlehnung an Auto-Fertigungsstraßen. Modellgetriebene Softwareentwicklung, CASE-Tools, Software-„Generatoren“, es gab unzählige Ansätze, in diese Richtung zu gehen. Das Wasserfallmodell tat sein übriges und es kam wie es kommen musste, unzählige Softwareentwicklungsprojekte scheiterten. Sie scheiterten bei der Zeit, sie scheiterten beim Scope, sie scheiterten beim Budget, aber, und das spricht nicht für Meyers Ansatz, sie scheiterten vor allem bei der Qualität.

## Die agile Revolution

Die Welt der Softwareentwicklung konnte so nicht weitermachen und immer mehr alternative Methoden und Entwicklungsprozesse erblickten die Welt. Extreme Programming (XP) brachte viele neue Konzepte mit sich, wie z. B. Unit Tests, Test Driven Development, User Stories, Story Points, etc. Feature Driven Development, DSDM, Kanban, Scrum, alle diese Methoden entstanden zu dieser Zeit aus dem Fakt heraus, dass so viele Projekte scheiterten.

Was danach kam, ist Geschichte: Die agilen Methoden wurden unter dem „Agile Manifesto“ zusammengebracht

und traten ihren Siegeszug bis heute an.

Was blieb dabei auf der Strecke? Sie haben es sicher erraten: die Qualität und das Testen

Ein großer Teil des Problems ist dabei die ultrakonservative und traditionelle Sicht des Softwaretestens in diversen Zertifikaten wie z. B. dem ISTQB Certified Tester Foundation Level. Z. B. die beim Foundation Level zugrunde liegenden Prozessmodelle wie z. B. der „fundamentale Testprozess“ erheben universalen Gültigkeitsanspruch unabhängig vom Entwicklungsprozess.

Die Gegenbewegung zu den traditionellen Zertifikaten waren Bücher wie „Agile Testing“ oder „More Agile Testing“ von Janette Gregory und Lisa Crispin oder auch die Certified Agile Tester Zertifizierung, die insbesondere von 2011 an den Markt dominierte.

All diese Entwicklungen zeigten die Schwächen von Meyers Konzepten auf. Das Testing wanderte wieder zurück in das agile Team und auch unser Produktportfolio entwickelte sich weiter: Mit unserem Agile Quality Coaching<sup>[1]</sup> helfen wir agilen Teams die gesamte Qualitätssicherung besser in ihren Entwicklungslifecycle zu integrieren und mit unseren Remote Testing Services<sup>[2]</sup> erweitern wir die Testkapazitäten unseres Kunden direkt in den agilen Prozess eingebettet.

### **Nun, was hat das alles mit Shift Left Testing zu tun?**

Shift Left Testing ist die konsequente Weiterentwicklung der agile testing Konzepte, die sich in den letzten 10 Jahren entwickelt haben.



#### **Eine Auswahl der wichtigsten Eckpunkte des Shift Left Testings sind:**

1. Mache Qualität zu einem Querschnittsmindset. Qualität betrifft jede einzelne Aktivität im gesamten Softwareentwicklungsprozess
2. Nutze Quality Gates wie Definition of Ready und Definition of Done im Prozess, um den Level der notwendigen Qualität zu fixieren und abzusichern
3. Sichere die Qualität bei der Anforderungserhebung durch bessere Kommunikation
4. Sichere die Qualität auf Ebene der Anforderungsbeschreibung durch Reviews
5. Sorge für Wartbarkeit und Lesbarkeit und damit für langfristige Werthaltigkeit des Codes bereits in der Entwicklungsumgebung durch Einsatz entsprechender Tools (Beautifier, Coding Style Guide, etc.)
6. Sichere die Qualität durch statische Code Analyse bereits in der Entwicklungsumgebung ab
7. Tracke technische Schulden, am besten toolgestützt
8. Erhebe und beobachte Codequalitätsmetriken wie Duplication, Cyclo-matic Complexity oder Code Smells
9. Sichere die Qualität auf Codeebene mit Unit Tests
10. Strebe die richtigen 100% an Unittestabdeckung an
11. Nutze alle Möglichkeiten ohne UI zu testen, z. B. durch Consumer Driven Contract Tests oder API-Tests
12. Vermeide so weit wie möglich automatisierte UI-Tests und nutze sie nur für die wichtigsten End 2 End Regressionstests
13. Führe alle weiteren regressiven Qualitätssicherungsschritte automatisiert über die Continuous Integration/Delivery/Deployment Pipeline aus
14. Konzentriere manuelle Testaktivitäten auf die kreativen und wichtigen Tests
15. Nutze Session Based Testing und Timeboxing so früh wie möglich im Prozess
16. Wende die Grundsätze der Softwareentwicklung auch auf Testartefakte an (Versionierung, keine Redundanzen, Modularisierung, Klasse statt Masse, usw.)
17. Dokumentiere alle Testaktivitäten automatisiert durch Testevidenzen
18. Nutze Testmethoden wie z. B. Grenzwertanalyse, Äquivalenzklassen oder Pairwise Testfallreduktion auf allen Ebenen (was bedingt, das alle im Team diese Techniken kennen)
19. Reporte den Qualitätszustand automatisiert in Echtzeit
20. Mache alle Testaktivitäten transparent und auch für den Kunden einsehbar



Die Auswahl zeigt, Shift Left Testing führt das Testen wieder dorthin zurück, wo es einst war: zu den Anforderungen und in das Umsetzungsteam. Testing ist immer und überall und die Teammitglieder sollten dazu in der Lage sein, jederzeit in die Testrolle zu schlüpfen, genauso wie in die einer Programmiererin, des Architekten oder eines Requirement Engineers. AI wird jeden einzelnen der oben genannten Punkte in Zukunft erleichtern und unterstützen können.

### Schlussfolgerung: Alles wird technisch

Shift Left Testing bildet einen Trend ab, der sich in den letzten Jahren immer mehr verstärkt hat und nun durch AI nochmals einen großen Boost bekommen hat: Testen wird immer technischer und ein guter Tester zu sein, ist ohne Entwicklungsskills nur noch im sehr eingeschränkten Umfeld des Abnahmetestens z. B. für den Fachbereich des Kunden möglich. Die Mitarbeit in einem agilen Team erfordert in jedem Fall die entsprechenden technischen Skills um den Weg des Shift Lefts mitgehen zu können. Das hat natürlich entscheidende Auswirkungen auf das Berufsbild des Softwaretesters. Die veraltete Form, die z. B. Meyers propagiert hat, wird mittelfristig keinerlei Relevanz mehr haben.

Dann ist wieder vereint, was nicht getrennt hätte werden dürfen. Doch ein großer Vorteil der temporären Trennung von Entwicklung und Test ist, dass sich Testing damit trotzdem stark weiterentwickelt hat und das neue gemeinsame Ganze „Shift Left“ ist damit deutlich stärker, als es die Softwareentwicklung 1979 je hätte sein können.

#### Quellen und weiterführende Informationen:

Meyers 1979, 2004, 2011  
<https://malenezi.github.io/malenezi/SE401/Books/114-the-art-of-software-testing-3-edition.pdf>  
<https://agilemanifesto.org/>

<sup>[1]</sup>Agile Quality Coaching: <https://www.seqis.com/de/leistungen/methodology-tools>

<sup>[2]</sup>Remote Testing Services: <https://www.seqis.com/de/leistungen/remote-testing-services>



Was verbirgt sich hinter diesem QR Code? Seien Sie gespannt!



**Alexander Vukovic** ist SEQIS Gründer und Chief Evangelist.

Er ist erster Ansprechpartner für alle agilen, testmethodischen und test-technischen Anfragen. In der Praxis arbeitet er als Agile Quality Coach, Berater, Interims-Testmanager, CI-Experte und Lasttester. Mehr als 25 Jahre Beratertätigkeit führten ihn während seiner zahlreichen Projekte in die unterschiedlichsten Branchen und Länder.

Sein persönliches Motto „Es gibt keine Probleme, sondern nur nicht gefundenen Lösungen“ spiegelt sich in jedem Projekt wider.

## Referenzstory: ZETA GmbH

mit Georg Kotsis



Die ZETA Gruppe ist ein weltweit agierender End-to-End-Lösungsanbieter für die Biotech- und Pharmaindustrie. ZETA bietet umfassende Lösungen und Services entlang des gesamten Wirkstoffentwicklungs- und Herstellungsprozesses.

Georg Kotsis: „**razzfazz.io** wurde von uns mit der Erstellung von browser-basierten Lebensläufen und Mitarbeiter\*innenprofilen auf Basis einer OpenSource Software beauftragt. Das Ergebnis kann sich sehen lassen: Das Team punktete dank perfektem Requirements-Engineering mit effizienter Software-Anpassung, sowie exakter Testung und Qualitätssicherung. Die Anpassungen beinhalteten neben der Umstellung der Datenbank und des Authentication Providers, auch die fachliche Unterstützung bei der Optimierung der CV Sections, der Layoutstandardisierung uvm. Zudem gab es technische Unterstützung beim Hosting der Software. Der gesamte Entwicklungsprozess erfolgte sehr professionell und effizient. Wir sind mit der erhaltenen Lösung äußerst zufrieden.“



Georg Kotsis, Corporate Head of Enterprise Management  
ZETA GmbH



## Nachhaltigkeit in der Software Entwicklung

von Klemens Loschy

### Was versteht man eigentlich unter Nachhaltigkeit?

Wie immer hat Wikipedia eine (oder sogar mehr, vielleicht sogar richtige oder zumindest passende) Antwort parat: **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 liest sich etwas sperrig und wenig IT bezogen... Wenn man dann weiterliest: Im entsprechenden englischen Wort **sustainable** ist dieses Prinzip wörtlich erkennbar: to sustain im Sinne von „aushalten“ bzw. „ertragen“. Mit anderen Worten: Die beteiligten Systeme können ein bestimmtes Maß an Ressourcennutzung „dauerhaft aushalten“, ohne Schaden zu nehmen. Der Definition kann ich was abgewinnen, aber aus meiner Sicht ist das für die IT nicht (ganz) zutreffend: in der IT geht es nicht darum bzw. darf es nicht **nur** darum gehen, ein „Maß an Ressourcennutzung dauerhaft auszuhalten“, das bekommen wir in der IT mittlerweile gut hin: „das erschlagen wir mit Blech“ ist seit mittlerweile vielen Jahren eine gängige „Strategie“, wenn Systeme dieses Maß an Ressourcennutzung eben **nicht** aushalten. Danach folgen eine Reihe von Interpretationen und Definitionen aus verschiedenen Kreisen der Gesellschaft, Politik und Wirtschaft. Später folgen dann noch drei Schlagwörter, die aus meiner Sicht dann aber gut passen (wenn auch nicht unbedingt deren Erklärung): **Suffizienz, Effizienz, Konsistenz** (wobei sich mir „Konsistenz“ und dessen Erklärung nicht wirklich erläutern).

OK, Wikipedia wie es lebt und lebt.

Aber wir sind ja im AI Zeitalter angekommen, was behauptet ChatGPT (V4), wenn man konkret nach „Nachhaltigkeit in der IT“ fragt? ChatGPT ist der Meinung, dass es dabei um **Energieeffizienz, Ressourcenschonung, Grüne Softwareentwicklung** (wobei das ebenfalls wieder Richtung Ressourcenschonung geht), **umweltfreundliche IT-Infrastruktur** (quasi wieder Ressourcenschonung), **nachhaltige Lieferketten** (in denen Ressourcen geschont werden), **soziale Nachhaltigkeit**. Immerhin ein neuer Aspekt: **soziale Nachhaltigkeit**, sonst mehr oder weniger doch auch wieder dasselbe...

### Nachhaltigkeit @ razzfazz.io

Vielleicht ist es zu viel verlangt, auf eine anerkannte und allgemein gültige Definition von Nachhaltigkeit aufbauen zu können. Es fühlt sich in etwa so an wie Diskussionen zu Softwaretesting vor ~20 Jahren: da gabs ein paar Typen, die irgendwas von „Qualität“, „Testen ist wichtig“ oder sogar „Softwaretest ist cool“ erzählt haben und nur wenige haben das so richtig verstanden, was das bringen soll. Viel Arbeit, Zeit und Überzeugungsaufwand war notwendig, um die Wichtigkeit dieser Themen nachhaltig in der IT zu verankern, und sogar heute, 20 Jahre später, trifft man immer noch Menschen und Unternehmen, die mit Themen wie Qualität und Testen so überhaupt nichts oder nur wenig anfangen können.



Abbildung 1: (SEQIS Messestand aus dem Jahr 2007 - SEQIS GmbH)

Also beschreibe ich meine bzw. die Sicht von razzfazz.io zum Thema Nachhaltigkeit in der Software Entwicklung: **Suffizienz** subsumiert das als Überbegriff aus meiner Sicht sehr gut und kann man auf sehr viele Bereiche und Tätigkeiten in der Software Entwicklung anwenden. Suffizienz heißt, **Material und Energie zu sparen** - das hört sich doch schon mal nachhaltig an. Dabei geht es gar nicht nur um technische Ressourcen, auch der Mensch/Mitarbeiter ist Teil dieses Suffizienz Prinzips und der richtige Einsatz dieser Ressourcen dementsprechend wichtig und bildet die Brücke hin zu soziale Nachhaltigkeit. **Soziale Nachhaltigkeit** hat aber auch noch andere Gesichtspunkte: Nicht jeder User läuft mit der neuesten Generation an Handys oder Notebooks herum und kommt in den Genuss von beinahe unbegrenzten und super schnellen Jederzeit- und Überall-Internet und ist obendrein noch Technik affin. Diese eingeschränkte Personengruppe ist natürlich der Hauptumsatztreiber unserer Produkte und dementsprechend stark im Fokus - doch haben wir diese Gruppe mit unserer Art von Software nicht überhaupt erst erschaffen? Ist das nicht eher ein Henne-Ei-Problem? Durch die Fokussierung unserer Software auf diese Zielgruppe schließen wir einen beträchtlichen Teil von Menschen einfach aus oder verringern zumindest deren User Experience zum Teil stark.

Nachhaltigkeit heißt also für uns bewusst etablierte Prozesse und de facto Standards zu hinterfragen, ob diese noch zeitgemäß im Sinne des Nachhaltigkeitsgedankens sind, weil wir davon überzeugt sind, dass die IT seinen Teil dazu beitragen muss und auch kann, die Welt ein Stück weit besser zu machen.



### Wieso hat es Nachhaltigkeit in der IT so schwer?

„Weils wurscht ist“ - oder zumindest „war“. Viele Dinge werden (leider) erst dann wichtig, wenn deren nicht Einhaltung zu (monetären) Verlusten führt. Bevor es nicht einmal so richtig in der Produktion gekracht hat, ist Qualitätssicherung maximal eine Vorgabe im Prozess. Erst, wer einmal ein Problem in Produktion miterlebt hat und deswegen weiß, welche Kosten und Schäden damit verbunden sind, der wird in Zukunft alles tun, damit diese Situation nie, nie wieder eintritt. Mit der Nachhaltigkeit ist es (leider) ähnlich: bis vor Kurzem (Corona, Energiekrise, Inflation, Klimakrise) gab es einfach nur wenig unmittelbaren Nutzen auf Nachhaltigkeit zu achten, es hat sich einfach nicht „ausgezahlt“. Erst in den letzten Jahren steigt langsam das Bewusstsein, dass „mehr Blech“ sicher nicht die beste Lösung ist, und auch nicht mehr die günstigste. Nachhaltige Lösungen sind gefragt, um Software ressourcenschonend betreiben zu können, denn Hosting und Stromkosten sind keine kleinen Posten mehr im Budget. Steigende Mitarbeitergehälter, sowohl in AT aber auch bei Off- und Nearshoring Partnern, gepaart mit dem Trend zu weniger Arbeitszeit, und der immer weiter steigende Wettbewerb am Markt bedingen den gezielten und richtigen Einsatz der Mitarbeiter. Slack oder falsch eingesetzte Ressourcen sind mittlerweile einfach zu teuer. Veraltete, starre Prozesse, die Change weitestgehend verhindern und den Status Quo sprichwörtlich einzubetonieren, und die weit verbreitete Eigenschaft von Menschen „Veränderung ist allem voran erstmal was Negatives“ arbeiten dem Thema Nachhaltigkeit aktiv entgegen.

### Was also tun?

Nachhaltigkeit muss Einzug in unseren IT-Alltag halten, so wie vor vielen Jahren die Qualitätssicherung langsam aber stetig an Einzug gehalten hat. Noch bedeutet das, dass wir aktiv daran arbeiten müssen, um

das Thema nicht aus den Augen zu verlieren, bis es sich so weit gefestigt hat, dass es „normal“ geworden ist. Es braucht diese enthusiastischen Typen, die immer wieder Fragen „ist das eigentlich Nachhaltig, was wir da machen?“ oder „wie können wir das nachhaltiger gestalten?“ und mit jeder Fragen kommen wir dem Ziel einen Schritt näher: Nachhaltigkeit ist weder „wurscht“ noch ist es „Wahnsinn“, es ist integraler und nicht weg diskutierbarer Bestandteil der professionellen Softwareentwicklung!

#### Quellen und weiterführende Informationen:

<https://de.wikipedia.org/wiki/Nachhaltigkeit>

<https://chatgpt.com>



## Power-Webinar garantiert!

# Nachhaltigkeit in der Softwareentwicklung: Wurscht oder Wahnsinn?

  
WEBINAR

  
26. SEPT.

  
16:00 Uhr

Jetzt kostenlos anmelden!



  
**Klemens Loschy**  
„Nachhaltigkeit in der Softwareentwicklung“: Ist das absurd?



[www.SEQIS.com/events-index](http://www.SEQIS.com/events-index)



**Klemens Loschy** ist Mitglied des SEQIS Geschäftsleitungsteams und Principal Consultant.

Er kann auf jahrelange Erfahrung in den Bereichen Testautomation, Last-Tests und Performance Engineering, funktionale Tests, Testen in agilen Teams, Anwendungsentwicklung von Testsoftware sowie Beratung und Unterstützung in zahlreichen Projekten unterschiedlichster Branchen zurückblicken.

## Codeium - der einzig sinnvolle Code Assistent?

von Daniel Kleissl



Abbildung 1: (Quelle: Bild erstellt mit ChatGPT 4, OpenAI)

### Codeium - der einzig sinnvolle Code Assistent?

Jeder Programmierer und jede Programmiererin kennt mindestens eine Tätigkeit in der Softwareentwicklung, die er oder sie lieber delegieren oder automatisieren würde. Für die einen sind es Unit-Tests, für die anderen ist es Code-Cleanup.

Gerade bezogen auf repetitive Tasks versprechen AI-Sprachmodelle die Befreiung und mehr Zeit für jene Aufgaben, die Erfahrung oder unsere Kreativität erfordern. Doch können die derzeit verfügbaren Assistenten hier wirklich eine Unterstützung bieten?

Mir war es ein Anliegen, die Fähigkeiten eines Code-Assistenten mit Integration in VS Code auszuloten. Selbst, wenn dieser nur eine kleine Beschleunigung für von mir unlieb-samen Tasks bedeuten würde, wäre das schon ein großer Gewinn. Doch bevor ich mich für eine Integration entscheiden konnte, war es nötig zu beleuchten, wie die verschiedenen Anbieter das Thema Datenschutz behandeln und welche Bedeutung das für uns als Programmierer im Allgemeinen und für Auftragsarbeiten

für Dritte im Besonderen hat.

### Copy Right vs Copy Wrong

Dass die Copyright-Problematik gerade für kommerzielle User äußerst relevant ist, hat mittlerweile auch die Akteure im Feld der AI-Sprachmodelle erreicht. Aus diesem Grund, sind von den gängigsten AI-Assistenten zumindest gewisse Versionen verfügbar, die laut Endnutzervereinbarung die Daten nicht zum Training von neuen AIs einsetzen. Zum Beispiel hat Microsoft Copilot, ehemals Bing Enterprise, einen solchen Passus in seinen Verträgen enthalten. Allerdings ist dieser Assistent nur mit bestimmten Enterprise- oder Universitätslizenzen verfügbar<sup>[1]</sup>. Das Gleiche gilt für ChatGPT, jedoch verbirgt sich diese Version hinter deren Enterprise Version, für die nicht einmal ein öffentlich bekannter Preis existiert.

Codeium, ein AI-Coding-Assistent mit Plugins bzw. Extensions unter anderem für JetBrains IDEs und VS Code, erlaubt im Kontrast dazu Nutzern auch bei deren kostenfreien Version mittels einfachem Opt-Out die Verwendung der eigenen Telemetrie (z.B. lokale Latenz, welche Features verwendet wurden, wie viel

Vorschläge akzeptiert oder abgelehnt wurden etc.) zu unterbinden. Darüber hinaus verwendet Codeium generell keinen User-Code zum Training von deren AIs. Codeium hat allerdings laut deren Marketingmaterial noch ein anderes Alleinstellungsmerkmal, das vor allem für die Verwendung von AIs als Programmierassistent sehr relevant ist, worauf ich im nächsten Kapitel eingehe<sup>[2]</sup>.

### Illegalität per Assistenz - Ein Lösungsansatz

Es ist eine Sache, wenn User durch ihre Fragen an die AI dem Unternehmen dahinter mitunter geheime Daten liefern, die entweder nicht nach außen dringen sollten oder die das eigene Intellectual Property in Gefahr bringen. Es ist aber eine ganz andere Thematik, wenn das Unternehmen, das die AI zur Verfügung stellt, diese mit einer unglaublichen Menge an Daten füttert, ohne hundertprozentig sicherzustellen, dass keine geheimen oder patentierten Code-Beispiele in den Trainingsdaten vorkommen. Spuckt nun der jeweilige Coding-Assistent geschützten Code aus und der Programmierer übernimmt diesen, dann stellt sich für mich die Frage, wie die Situation hier aus rechtlicher Sicht zu betrachten ist<sup>[3]</sup>. Es wird mitunter im Nachhinein sehr schwierig nachzuweisen sein, dass ein fremder, nicht offener Algorithmus in der eigenen Codebasis durch eine solche AI-Assistenz hineingelangt ist.

Nun ist es aber keinem User zumutbar, jeden generierten Code auf das Vorhandensein von patentierten Code-Teilen zu überprüfen.

In genau diese Kerbe schlägt Codeium mit deren Integration. Laut deren Website hat Codeium die Trainings-

daten aufbereitet, damit sichergestellt wird, dass alle nicht öffentlichen Code-Teile daraus entfernt werden. Dass dafür die alleinige Verwendung von Open Source Datensätzen nicht ausreicht, stellt Codeium in einem spannenden Blog Artikel zur Schau<sup>[4]</sup>. Kurz zusammengefasst, ist die Problematik jene, dass Open Source nicht automatisch bedeutet, dass darauf aufbauend auch kommerzielle Projekte entwickelt werden dürfen. Diese Art von Verwendung von Open Source Projekten fällt unter ein spezielles Subset von Lizenzen, womit es nicht ausreicht darauf zu achten, nur Open Source Daten zum Training zu verwenden. Codeium behauptet hier von sich selbst, mit Hilfe von Filtern und sogar manuellen Modifikationen alle problematischen Daten zu entfernen. Selbst, wenn diese Behauptung als User schwer zu überprüfen ist, allein, dass sich das Unternehmen darüber Gedanken macht, zeigt für mich die richtige Herangehensweise bei der Bereitstellung solcher Services. Dadurch war Codeium der einzige Assistent mit VS Code Integration, der für eine nähere Betrachtung in Frage gekommen ist.

Doch wie ist nun das Fazit nach mehr als 6 Monaten, in denen ich Codeium verwendet habe?

### Noch ein weiter Weg zum perfekten Assistenten

Wir verwenden bei razzfazz.io TypeScript für die Programmierung unseres aktuellen Projekts. Für all jene, die keine Erfahrung mit TypeScript haben: Die Typen von TypeScript sind "nur" ein Compile-Time Feature, d.h. zur Laufzeit sind die Typen und die dadurch bereitgestellte Informationen nicht mehr vorhanden. Das bedeutet für den oder die Programmierer oder die Programmiererin einerseits ein gewisses Umdenken bei der Verwendung von Typen, andererseits bietet die Sprache dadurch auch nicht dieselben Run-Time Features wie z.B. Java. Diese Komplexität ist nicht nur für Developer herausfor-

dernd, sondern auch für die Codeiums "Fill-in-the-middle"-Funktion.

Die Schwierigkeit besteht unter anderem darin, dass es in TypeScript, wie in jeder anderen Sprache auch, eine Menge Wege gibt, um dasselbe Ziel zu erreichen - allerdings gekoppelt mit der zusätzlichen Erschwernis eines komplexen, optionalen Typensystems. Hat man sich zusätzlich in einer Codebase auf gewisse Coding-Conventions geeinigt und eine Menge an Typen angelegt, dann werden die Vorschläge der AI immer weniger wertvoll, da die Menge an "Meta-Information" schlicht zu groß wird. Häufig schlägt diese dann generische Lösungen vor, die zwar funktionieren, aber streng genommen falsch sind, da diese eben einen anderen Stil verwenden oder Typen, die an einer anderen Stelle in der Codebase definiert sind, nicht anwenden. Damit befindet man sich schnell in einer Situation, die einem stetigen Code-Review von Fremddate gleich. Dies stellt eine komplett andere Herausforderung zum "einfachen" Programmieren dar und erfordert weit mehr Konzentration. Diese Problematik lässt sich zwar von Seiten der AI-Entwickler verbessern, indem der AI mehr Kontext mitgegeben wird. Damit befindet man sich allerdings mitten in der Context vs. Latency Problematik, die derzeit die größte Herausforderung für cloudbasierte AI Modelle darstellt.

### Latency vs Context-Länge - Vom Regen in die Traufe

Will man von einem Large Language Model (LLM) bessere Antworten bekommen, dann benötigt dieses unter anderem auch mehr Input-Daten über den Prompt, den sogenannten Context. Allerdings haben cloudbasierte Systeme hier eine doppelte Herausforderung: Erstens müssen sie den Context über die Internetverbindung übertragen, was eine gewisse Zeit benötigt. Andererseits dauert die Inference des Models, also das Erstellen der Antwort, auch eine gewisse Zeit, und die Antwort muss

über dieselbe Internetverbindung an den User zurückgeliefert werden. Erschwert wird das ganze davon, dass mit anwachsender Context-Länge die Antwort des LLMs auch mehr Zeit benötigt. Hier müssen also die Entwickler dieser Integrationen einen Sweet-Spot finden. Wenn die Latenz zu groß ist, dann kommt die AI selbst mittelmäßig schnellen Schreibern nicht hinterher. Wird allerdings der Context zu kurz, dann werden die Antworten der AI immer weniger spezifisch, und dafür für den User weitestgehend unbrauchbar<sup>[5]</sup>.

Genau in diesem Spannungsfeld befindet sich auch Codeium mit seiner VSCode Integration. Selbst als nicht sehr schneller Schreiber merke ich, dass die Integration eine gewisse Zeit benötigt, um einen Vorschlag zu unterbreiten. Das Feeling ist dabei nicht vergleichbar mit einer normalen Auto-Completion. Bei mir bürgerte sich mit der Zeit eine Art "Stop-and-Go" Programmieren ein, da ich unbewusst auf den Vorschlag der AI warte. Wenn dieser dann noch dazu nicht passt, ist das umso ärgerlicher. Zumindest kann damit gerechnet werden, dass gerade hier neuere AI-Modelle eine Beschleunigung liefern und die Qualität der Antworten auch mit weniger Context besser werden wird. Doch Codeium hat sich auch eine Möglichkeit einfallen lassen, um das Wissen des Users über die eigene Codebase zu nutzen und für die Erstellung eines sinnvollen Contexts einzusetzen. Dieses Feature wurde als Prototyp im März 2024 unter dem Namen "Context Pinning"<sup>[6]</sup> vorgestellt. Hierbei handelt es sich um eine Möglichkeit, in der Konfiguration des Assistenten Files oder Verzeichnisse zu hinterlegen, die besonders relevante Informationen beinhalten. Für TypeScript wären das z.B. Files mit Typendefinitionen oder Helper-Klassen. Bei den ersten Tests von Codeium unter Verwendung dieses Features haben sich noch keine massiven Sprünge in der Code-Qualität gezeigt, allerdings ist das auch nicht die Erwartungshal-

tung an ein Prototype-Feature.

Es zeigt jedoch einmal mehr die richtige Herangehensweise von Codeium: Das Ziel soll eine echte Kollaboration zwischen AI und Programmierer sein. Und damit ist es nötig, dem User niederschwellige Tools in die Hand zu geben, womit die AI auf die eigenen Bedürfnisse zugeschnitten werden kann. Jenseits eines Modells, das mit der kompletten eigenen Codebasis trainiert wurde, wird jeder Algorithmus es schwer haben, genau zu erkennen, wo die wichtigen Stellen im Code zu finden sind. Die Idee des "Context Pinning" ist somit eine Win-Win Situation, da Codeium sich einen Teil der Entwicklung eines Modells zur Context-Erkennung erspart, und der Programmierer die Möglichkeit bekommt, den Assistenten genau zu fokussieren und personalisieren. Es ist mit dem derzeitigen UI sogar möglich, den Context vor einer Chat-Anfrage zu modifizieren, falls nur ein Subset der markierten Files relevant ist.

Der Vollständigkeit halber sei auch gesagt, dass Codeium die Möglichkeit bietet, einen kurzen "Default-Prompt" zu hinterlegen, der jeder Anfrage angehängt wird. Dieses ermöglicht eine weitere Personalisierung des Assistenten, z.B. wenn ich ein bestimmtes Framework verwende, und möchte das Codeium seine Antworten darauf bezieht. Ähnliche Möglichkeiten bieten auch andere AI Assistenten wie ChatGPT, daher gehe ich hier nicht genauer darauf ein.

### Kollaboration zwischen Mensch und Maschine

Es gibt Aufgaben beim Programmieren, für die weder Fill-in-the-Middle noch ein separates Chat-Fenster der richtige Ansatz sind. So kann man sich in der Situation befinden, dass man zwar in einfachen Worten beschreiben kann, was man mit einer Codestelle erreichen will, aber man weiß nicht genau, wie man diese Aufgabe am besten angehen soll. Oder

man hat eine Codestelle, die man auf eine gewisse Art refaktorisieren will, die zwar gut beschreibbar, aber schwer automatisierbar ist.

Hier zeigt sich der große Vorteil eines weiteren Moduls von Codeium: Codeium Command. Dieses Modul ermöglicht direkt im Editor der IDE ein Fenster zu öffnen, über das dann dem AI-Assistenten Anweisungen gegeben werden können. Dieser erstellt dann ein Vorschlags-Diff, das entweder direkt akzeptiert, abgelehnt oder mittels der "Follow-Up"-Funktion modifiziert werden kann. Letztere ist hilfreich, wenn der Vorschlag zwar nahezu dem gewünschten Ergebnis entspricht, aber eben kleinere Anpassungen benötigt. Wie immer bei der Konversation mit einem AI Assistenten lernt man mit der Zeit, wie man den "Prompt" strukturieren muss und welche Informationen wichtig sind. Z.B. kann man bei

sich wiederholenden Refactorings einen schon umgeschriebenen Block gemeinsam mit dem zu refaktorisierenden Block markieren, und somit der AI ein Beispiel geben, an welche Struktur sie sich halten soll. Oder man kopiert diesen Codeblock in das Command-Fenster und gibt so der AI einen Anhaltspunkt zur gewünschten Struktur.

Ein interessanter psychologischer Effekt bei der Verwendung von Codeium Command war, dass es weit weniger kognitive Anstrengung erforderte, den so generierten Code zu überprüfen. Das mag einerseits daran liegen, dass man sich durch die Verwendung des Command-Fensters bewusst in eine "Code-Review" Situation begibt, andererseits ist die Formatierung des Vorschlags als Diff gut lesbar, und ein Format, das jeder Programmierer vom Lesen von Git Commits gut kennt.

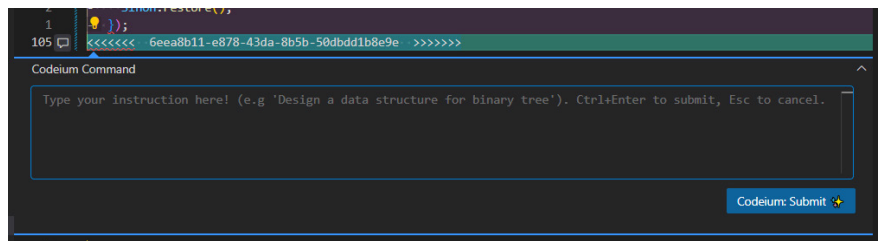


Abbildung 2: Codeium Command Window (Quelle:SEQIS GmbH)

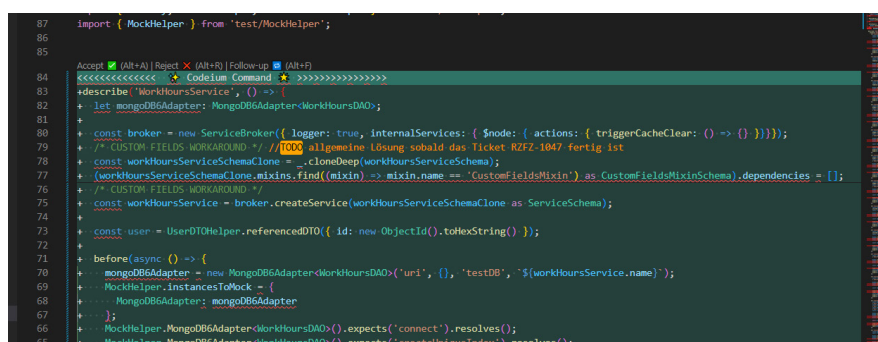


Abbildung 3: Codeium Command in Action (Quelle:SEQIS GmbH)





## Fazit

Für mich hat sich Codeium Command in der normalen Programmierung als das hilfreichste Modul herausgestellt. Fill-in-the-Middle hat aufgrund der Latenz-Problematik und Qualität der Codevorschläge noch einiges an Weg vor sich, bis es eine ähnlich störungsfreie Unterstützung wie z.B. Intellisense bietet. Codeium Chat wiederum mag für die Exploration von unbekanntem Repositories interessant sein, oder um sich das Öffnen eines Browserfensters und die Internetsuche zu ersparen, aber auch hier zeigt sich, dass die Modelle, vor allem GPT 3.5, noch nicht ganz dort sind, wo man sie sich für eine gute Unterstützung wünschen würde. Allerdings kann damit gerechnet werden, dass vor allem hier neuere Modelle weit besser performen und sukzessive auch für den Free-Tier ausgerollt werden. Für Nutzer, die ein bezahltes Abo abgeschlossen haben, stellt Codeium schon jetzt einen auf GPT 4 basierenden Chat zur Verfügung.

Codeium Command allerdings bietet in einem relativ engen Anwendungsbereich eine gute Unterstützung und erreicht dies wiederum einerseits durch Einschränken dessen, was die AI als Aufgabe bekommt, und andererseits dadurch, dass sie dem Entwickler eine einfache Möglichkeit bietet, die AI anzuleiten. Und gerade für mich bietet Codeium hier bei Aufgaben, die durch ihre repetitive Natur anstrengend sind, eine echte Verbesserung.

Auch, wenn Codeium noch am Anfang steht und daher voraussichtlich noch nicht für jeden Programmierer oder jede Programmiererin eine große Unterstützung darstellt, möchte ich dennoch eine Empfehlung dafür aussprechen, in der Hoffnung, dass möglichst viele Personen zumindest das Experiment wagen und die Integration installieren. Wir als Nutzer solcher AI-Assistenten haben hier nicht nur die Möglichkeit, sondern in meinen Augen auch die Verpflichtung durch

die Wahl unserer Tools, die Entwicklung in diesem Feld mitzugestalten.

Und da vor allem Codeium in meinen Augen so viel bei der Entwicklung eines Code Assistenten richtig macht, hoffe ich, dass diese Integration erfolgreich ist und somit auch als gutes Beispiel gelten kann, worauf es bei einer AI Integration ankommt: Kolla-

boration zwischen AI und Entwickler, leicht bedienbares UI und gute, einfach verwendbare Anpassungsmöglichkeiten.

Da Codeium in der Basis-Variante kostenlos verfügbar ist, handelt es sich hier um ein Experiment, das hoffentlich viele wagen werden.

### Downloads

<https://marketplace.visualstudio.com/items?itemName=Codeium.codeium>

### Quellen

<sup>[1]</sup> <https://learn.microsoft.com/en-us/copilot/manage#commercial-data-protection-eligibility>

<sup>[2]</sup> <https://codeium.com/faq>

<sup>[3]</sup> <https://x.com/DocSparse/status/1581461734665367554>

<sup>[4]</sup> <https://codeium.com/blog/copilot-trains-on-gpl-codeium-does-not>

<sup>[5]</sup> <https://codeium.com/blog/latency-the-ultimate-constraint>

<sup>[6]</sup> <https://codeium.com/blog/codeium-context-pinning-launch>

### Weitere Information

<https://razzfazz.io/>

<https://codeium.com/>



**Daniel Kleissl** ist Agile Development Wizard.

In der Weiterentwicklung des SEQITrackers konnte er wertvolle Erfahrung im Bereich Full Stack Development sammeln und sein Wissen aus seinem Fachhochschulstudium vertiefen. Zusätzlich konnte er sich durch die Jira-Cloud-Plugin Entwicklung wichtiges Know-How im Cloud-Development aufbauen.

Für ihn sind Clean Code und eine gute Testabdeckung die wichtigsten Säulen eines guten Entwicklungsprozesses.

# razzfazz.io

## boost results

### Wie wir sind

Unser Anspruch ist es, Ihre Entwicklungsaufgaben erstklassig umzusetzen.

Wir schaffen gemeinsam mit Ihnen die richtige Basis für Ihre Aufgabe: Wir setzen auf eine fundierte IT Analyse und erarbeiten bereits hier gemeinsam die Lösung, die Sie sich wirklich wünschen. Dabei ist Ihnen von Anfang an klar, welche Lösungsteile wir in den kommenden Iterationen an Sie übergeben werden.

Und wenn sich Ihre Wünsche im Projektverlauf ändern: Durch unsere agile Vorgehensweise adressieren wir diese Änderungen - somit werden letztlich nur jene Anforderungen realisiert und geliefert, die Sie aktuell auch wirklich benötigen.

Analyse - Entwicklung - Test - Übergabe: professionelle und zuverlässige Begleitung für Ihr Projekt.

**Weitere Informationen finden Sie auch auf unserer Webseite:**  
[www.razzfazz.io](http://www.razzfazz.io)

### Was uns ausmacht

Unsere Wurzeln reichen zurück auf Qualitätssicherung und Systemanalyse, durch die Unterstützung vieler Entwicklerteams in den letzten 20+ Jahren.

Wenn wir uns darauf geeinigt haben, was Ihre Anforderungen sind, freuen Sie sich auf unsere Ergebnisse: Wir wissen, wie man richtig Qualität von Anfang an in die Software verankert. Unit Tests, Test Automation, Last- und Performancetests, Usability und Securitytests sind integrale Bestandteile unserer Entwicklungsprojekte.

Darüber hinaus sichern wir ein perfektes Alignment mit Ihrem Wertschöpfungsprozess: Wir stehen dafür, das Richtige richtig zu tun.

### Unsere Leistungen



Führende Technologien



Moderne Oberflächen



Individuelle Lösungen



Fundiertes Requirements Engineering



Perfekte Qualitätssicherung



Agiler Prozess



**„Das Leben ist zu kurz, um keine Unit Tests zu schreiben.“**

Alexander Vukovic  
Gründer und Chief Evangelist



**„razzfazz.io is faster than you can say performance!“**

Klemens Loschy  
Geschäftsleitung & Principal Consultant

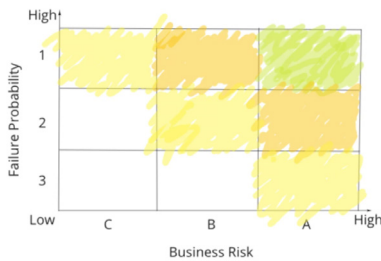


**„Software. Agile. Developed and tested by experts - razzfazz.io!“**

Alexander Weichselberger  
Managing Partner

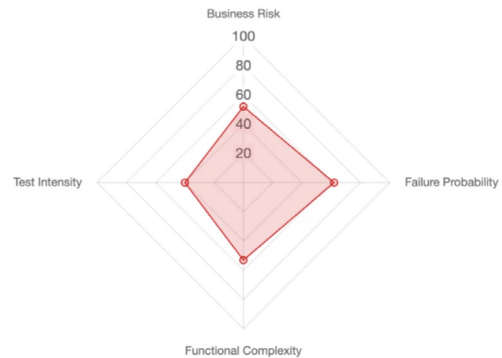
# Journey to Rome

Entdecken Sie intelligentes Risk Assessment für Jira mit unserem Plugin Journey to Rome! Berechnen Sie das Anforderungsrisiko und die Testintensität basierend auf 12 simplen Fragen und Antworten.

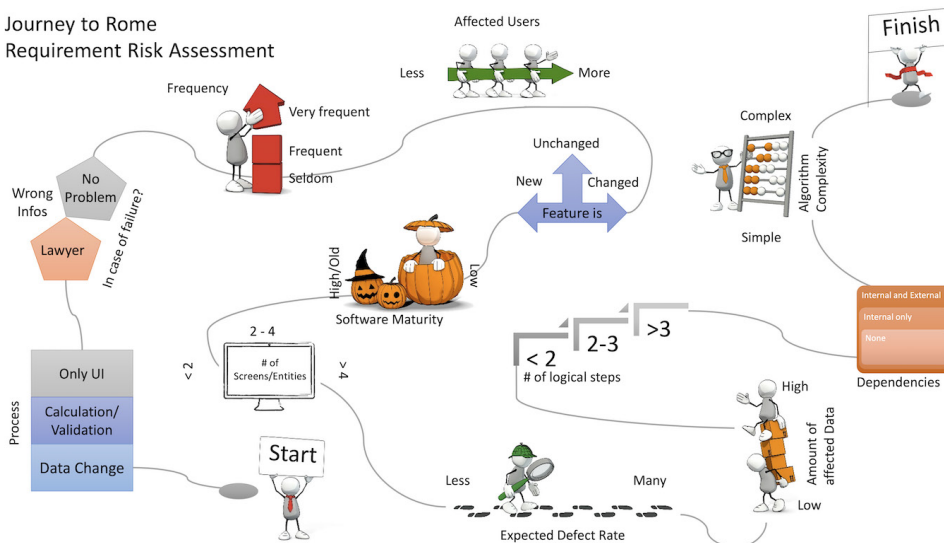


Journey to Rome zeigt die Fehlerwahrscheinlichkeit und den möglichen Impact auf Ihre Organisation in Form von Fehlerfolgekosten. Das Berechnungsergebnis ist unterteilt in die Risikokategorien Business Risk, Failure Probability und Functional Complexity. Daraus wird ein Risikoindex von A1 - C3 errechnet und die notwendige Testintensität von 1 bis 100 bewertet.

Die Testintensität berechnet sich basierend auf Risiko und Komplexität und ist ein Key Indicator für das risikobasierte Testen und das Testmanagement. Für das Testfalldesign und die Testplanung lassen sich ebenfalls wertvolle Entscheidungen aus der Testintensität ableiten.



## Journey to Rome Requirement Risk Assessment



### Get it now!

Weitere Informationen und die Möglichkeit, die Journey to Rome Basisversion in Ihrer Jira Cloud Instanz kostenlos zu nutzen, finden Sie auf unserer Webseite [www.razzfazz.io](http://www.razzfazz.io)

Intelligent Risk Assessment  
Journey to Rome



## Referenzstory: JAGGAER Austria GmbH

mit Martin Kersch



„SEQIS ist schon seit Jahrzehnten als Softwaretester im deutschsprachigen Raum etabliert und verfügt nachweislich über eine hohe Expertise. Ich persönlich kenne kaum ein Unternehmen, das Remote Testing Services in diesem Umfang und vor allem mit dieser Skalierbarkeit anbietet.“

Martin Kersch, Chief Technology Officer, JAGGAER Austria GmbH

### Martin Kersch, Chief Technology Officer

Martin Kersch verfügt über mehr als 20 Jahre IT-Erfahrung in Unternehmen aller Größenordnungen und war in der Vergangenheit u. a. als Geschäftsführer und CIO tätig. Während seines Studiums der Betriebswirtschaftslehre an der Wirtschaftsuniversität Wien arbeitete er mehrere Jahre an globalen Projekten für ein international renommiertes Beratungsunternehmen.

Seit April 2020 ist Martin als Chief Technology Officer für JAGGAER tätig. In dieser Funktion ist er für die technische Konzeption, Implementierung und den Betrieb der Plattformen verantwortlich. JAGGAER ist der weltweit führende Anbieter von Beschaffungs-Lösungen & Software für digitale Source-to-Pay-Prozesse.



### Das Gespräch

- Martin Kersch, Chief Technology Officer, JAGGAER Austria GmbH
- Alexander Weichselberger, Managing Partner, SEQIS GmbH

**Weichselberger:** Lieber Martin, ich freue mich sehr, heute mit dir über unsere Zusammenarbeit und die gemeinsamen Projekte zu sprechen. Um für unsere Leserinnen und Leser etwas Licht ins Dunkel zu bringen, starten wir am besten mit einer kurzen Erklärung, was JAGGAER überhaupt macht.

**Kersch:** Sehr gerne! Hinter JAGGAER verbirgt sich das weltweit führende Unternehmen im Bereich Spend Management. Wir sind seit über 28 Jahren am Markt und bieten SaaS-basierte Beschaffungslösungen, die den gesamten Source-to-Pay-Prozess abdecken – angefangen bei Spend Analytics und Category Management über Sourcing, Lieferanten- und Vertragsmanagement bis hin zu eProcurement und mehr. Mit unserer umfassenden Software-Suite können die JAGGAER-User alle Ausgaben und Prozesse des direkten und indirekten Einkaufs über eine einzige Plattform steuern. So helfen wir unseren Kunden – zu denen internationale Größen aus diversen Industriezweigen und dem öffentlichen Sektor zählen – dabei, ihre Einkaufsprozesse zu optimieren, Kosteneinsparungen zu erzielen, Risiken zu minimieren und ihre Zusammenarbeit mit den Lieferanten zu stärken.

**Weichselberger:** Danke, Martin. Das klingt auf alle Fälle spannend, bringt aber sicherlich auch einige Herausforderungen in Sachen Qualitätssicherung mit sich, oder? Kannst du uns diesbezüglich ein paar Einblicke geben?

**Kersch:** Natürlich. Grundsätzlich streben wir mit unseren SaaS-Lösungen einen hohen Standardisierungsgrad an und entwickeln die gesamte Produktpalette kontinuierlich weiter, damit unsere User stets „state-of-the-art“ bleiben. All diese Funktionen werden selbstverständlich vor dem Rollout unserer Releases im Rahmen unserer internen Qualitätssicherung ausgiebig getestet. Aber natürlich sind in vielen unserer Projekte auch kundenspezifische Anpassungen notwendig, durch die sich die Anzahl der abzudeckenden Testszenarien sowie deren Fluktuation deutlich erhöhen. Dies international mit unserer eigenen Testabteilung abzudecken, ist nahezu unmöglich. Daher sind wir auf externe Unterstützung angewiesen, die es uns ermöglicht, die Individualisierungen entsprechend in unserer Qualitätssicherung zu berücksichtigen.

**Weichselberger:** Verstehe, und bei der Suche nach einem geeigneten Partner ist eure Wahl letztlich auf



SEQIS gefallen. Was war aus deiner Sicht ausschlaggebend für die Entscheidung?

**Kersch:** SEQIS ist schon seit Jahrzehnten als Softwaretester im deutschsprachigen Raum etabliert und verfügt nachweislich über eine hohe Expertise. Ich persönlich kenne kaum ein Unternehmen, das Remote Testing Services in einem solchen Umfang anbietet, die vor allem auch so gut skalierbar sind. Dazu kommt, dass ich in meinen letzten Positionen vor JAGGAER auch schon mit SEQIS gearbeitet habe und es immer sehr gut funktioniert hat. Diese Referenzen waren somit auch ausschlaggebend dafür, dass wir uns für SEQIS als Testing-Partner entschieden haben. Anschließend hat alles seinen Lauf genommen und wir haben Anfang 2022 zunächst mit der generellen Testkonzeptberatung und anschließend im Sommer mit den Remote Testing Services begonnen.

**Weichselberger:** Lass uns daran gerne anknüpfen und den Fokus nun darauf legen, wie die Zusammenarbeit zwischen JAGGAER und SEQIS aussieht und in welchen Bereichen wir euch konkret unterstützen.

**Kersch:** Im Allgemeinen erbringt SEQIS für JAGGAER fortschrittliche Qualitätssicherungs-Services, die sich bei uns konkret in zwei verschiedene Bereiche unterteilen. Auf der einen Seite haben wir ein Projekt zur grundsätzlichen Optimierung unserer internen QA gestartet, um der starken Fragmentierung entgegenzuwirken und die gesamten Prozesse effizienter zu gestalten. In diesem Zuge hat uns SEQIS ausführlich beraten und gemeinsam mit unserem Global Corporate Head of QS sowie unserem internen Expertenteam ein neues Testframework konzipiert. Auf der anderen Seite unterstützt uns SEQIS mit Remote Testing Services dabei, die spezifischen Anpassungen, die für einzelne Kunden umgesetzt werden, vor jedem Release entsprechend zu

testen. Das umfasst auch die vollständige Dokumentation und das Reporting, damit sowohl unser QA-Team als auch unsere Kunden einen transparenten Einblick in die Tests sowie Ergebnisse erhalten. Und das bringt uns wiederum auch Fortschritte für unser Standardverfahren: Denn Erkenntnisse aus kundenspezifischen Tests, die einen Mehrwert für den Gesamtprozess darstellen, werden durchaus mit dem Standard abgeglichen und im Falle eines positiven Ergebnisses eingeführt.

**Weichselberger:** Bei dem Stichwort „Mehrwert“ würde ich gerne einhaken: Worin siehst du generell die größten Vorteile, die JAGGAER durch die Zusammenarbeit mit SEQIS erzielt?

**Kersch:** Ein großer Pluspunkt war auf jeden Fall, dass alles super schnell aufgesetzt wurde und von Anfang an problemlos geklappt hat. Dadurch mussten wir uns nicht um eigene Ressourcen kümmern und haben auch keinerlei Ressourcen bei uns im Team gebunden. Das ist auch insofern praktisch, da wir durch die externe Unterstützung von SEQIS für jedwede Entwicklung gewappnet sind und die Skalierbarkeit auch bei einem stark wachsenden Bedarf im Remote Testing Bereich gesichert ist. Darüber hinaus verfügt SEQIS über eine sehr gute Testexpertise, die Mitarbeitenden kennen sich mit allen gängigen Test- und QA-Frameworks aus, verstehen, wie Testfälle geschrieben werden, und behalten stets den notwendigen Qualitätsstandard für QA bei. Davon profitieren wir enorm! Nicht zuletzt, da wir keine eigene Team-Infrastruktur für die kundenspezifischen Tests aufbauen müssen und gleichzeitig auch das Onboarding wegfällt. Das stellt eine große Erleichterung dar.

**Weichselberger:** Und wie würdest du die Zusammenarbeit, gemessen an den generellen Herausforderungen von Remote Testing Services, bewerten?

**Kersch:** Aus meiner Sicht funktioniert die Zusammenarbeit trotz Remote-Collaboration einwandfrei und wir haben alle Herausforderungen bisher gemeinsam erfolgreich gemeistert. Als globales Unternehmen sind wir es bei JAGGAER ohnehin gewohnt, dass mehr als 1.200 Mitarbeitende in Distributed Teams über verschiedene Kontinente hinweg verteilt sind. Natürlich ist das, auch durch die kulturellen Unterschiede bedingt, eine Challenge, aber am Ende des Tages klappt alles problemlos, da wir uns gemeinsam darauf einlassen und unsere Prozesse entsprechend organisiert haben. Wir leben die Digitalisierung in all ihren Facetten und jeder kann jeden gut erreichen. So ist es letztlich auch mit SEQIS: Obwohl die Tester nicht vor Ort an einem unserer Standorte tätig sind, sondern uns von außerhalb unterstützen, läuft die Zusammenarbeit so gut, als würden wir gemeinsam in einem Office sitzen.

**Weichselberger:** Das freut uns auf jeden Fall zu hören. Vielen Dank für die interessanten Einblicke. Abschließend würde mich noch interessieren, was die nächsten Steps sind, die bei JAGGAER in Sachen Qualitätssicherung anstehen?

**Kersch:** Unser Ziel ist es natürlich, die Qualität der Lösungen, die wir unseren Kunden zur Verfügung stellen, kontinuierlich zu verbessern, und entsprechend leistet die QA dazu einen wesentlichen Beitrag. Deshalb arbeiten wir daran zum einen im Rahmen der internen Prozessverbesserung, zum anderen haben wir aber auch das Bestreben, die Testabdeckung für unsere Kunden stetig zu erhöhen. Davon ausgehend, wird sich die Zusammenarbeit mit SEQIS sicherlich in Zukunft weiter vertiefen und auf ein breiteres Einsatzgebiet ausdehnen.

**Weichselberger:** Das hören wir gerne! Vielen lieben Dank für deine Zeit und den spannenden Input, Martin!

# Container based hosting: sicher nicht ohne LPT!

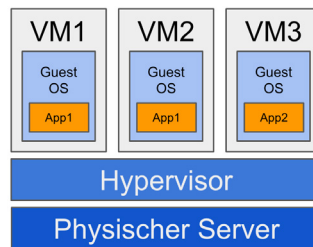
von Klemens Loschy

Ob AWS, GCP, Azure Cloud oder doch selbst gehostet mit Kubernetes oder OpenShift: Container based Hosting ist mittlerweile sehr etabliert oder ein Umstieg darauf zu meist bereits geplant.

### CBH: Virtualization on Steroids

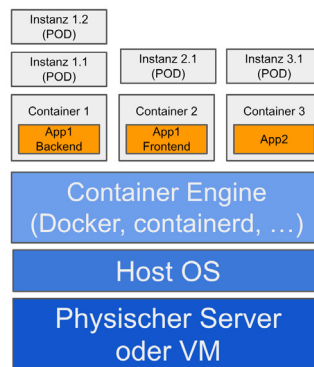
CBH bietet die Möglichkeit, Application Container (meist auf Docker Basis) zu hosten und darüber hinaus viele Enterprise Features wie z.B. das Zusammenfassen von mehreren Applikations-Instanzen zu logischen Services oder die Aufteilung dieser Instanzen auf verschiedene physische Nodes, automatische Restarts im Fehlerfall oder Skalieren dieser Instanzen aufgrund bestimmter Kennzahlen wie CPU Load: wertvolle Features, die moderne und robuste Infrastrukturen ermöglichen. Ein großer und stark unterschätzter Unterschied zu Standard Virtualisierung ist die „Größe“ der virtuellen Maschinen bzw. Container: wurde früher das ganze Betriebssystem inkl. Web- und Applicationserver virtualisiert, teilt man die Applikation bei CBH in viele kleine(re) Komponenten (Container): da gibt es Container für die Datenbank, einen für das Backend der Hauptanwendung, einen für das Admin-Interface, einen für das Customer-Interface... Das bringt Flexibilität, aber auch Komplexität mit sich. Damit einher verändert sich die Art der Skalierung: bei klassischer Virtualisierung hat man eher auf starke virtuelle Maschine gesetzt und bei Bedarf noch mehr RAM oder CPU zugewiesen, horizontale Skalierung war eher die Ausnahme und fast immer mit zusätzlicher (und zum Teil aufwendiger) Konfiguration verbunden. Mit CBH wird vorwiegend horizontal skaliert: sind zu viele Anfragen am Backend der Hauptanwendung? Man startet einfach eine zweite Instanz

derselben Komponente, die wird automatisch in den Service integriert und die Anfragen auf beide Instanzen verteilt.



### klassische Virtualisierung

Abbildung 1: (Quelle: klemens loschy - SEQIS GmbH)



### container based hosting

Abbildung 2: (Quelle: klemens loschy - SEQIS GmbH)

### Der Umstieg: Ein steiniger Weg

Auch wenn CBH eine Vielzahl von Vorteilen mit sich bringt, birgt der Umstieg auch viele Risiken: Die korrekte Funktionsweise der Konfiguration muss getestet werden! Das ist an sich nichts Neues und darauf habe ich schon in älteren Blog-Beiträgen hingewiesen (immer wieder sind und

waren Performance Probleme auf falsche Konfiguration zurückzuführen, auch lange vor CBH), nur mit CBH erreicht der Konfigurationsaufwand bzw. die Menge an Konfigurationsmöglichkeiten und deren Komplexität ein ganz neues Level. Zusätzlich sollte die Applikation in kleine Komponenten zerlegt werden, was oft mit einer nicht trivialen Anpassung der ganzen Applikation einhergeht. Zu diesem Zeitpunkt muss verstärkt durch fachliche Tests die korrekte Funktionsweise sichergestellt werden, aber schon jetzt kann man mit Hilfe von punktuellen und gezielten Lasttests erste aussagekräftige Ergebnisse erzielen. Im Normalfall ist man hierbei noch weit weg von einer produktionsnahen Simulation und die Ergebnisse sind somit nicht 1:1 auf den Produktivbetrieb anwendbar, dafür ist der eigentliche Aufwand für den Lasttest viel geringer und man bekommt schon sehr frühe erste belastbare Ergebnisse.

### Podcast-Tipp



### LPT for the win!

Ein Last- und Performancetest hilft dabei, diverse Fragestellungen im simulierten Produktionsbetrieb zu beantworten: werden die Instanzen bei hoher Last automatisch skaliert? Wie sehr profitiert meine Applikation von mehreren Instanzen? Werden diese Instanzen bei geringer Last auch wieder entfernt? Funktioniert das Skalieren unterbrechungsfrei oder kann das doch zu unerwarteten Fehlern führen?

Aber auch betriebsrelevante Fragen können geklärt werden: Was passiert, wenn während des Betriebs eine physische Node einfach wegbricht? Wie kurz (oder lange) dauert es, bis alle Instanzen auf anderen Nodes wieder gestartet und in die Services integriert und damit betriebsbereit sind? Oder aber welche Schritte sind notwendig, um eine Node plangemäß zu servicieren? Kann man das gefahrlos unter Tags machen oder muss es dafür eigene kommunizierte Wartungsfenster spät Abends, am Wochenende oder mit Downtime geben? Last- und Performancetests helfen dabei, das Operations-Team sich mit der neuen Infrastruktur vertraut zu machen, das Monitoring zu testen und die Arbeitsschritte in Ausnahmesituationen zu proben.

Die Empfehlung ist aber ganz klar, bereits den eigentlichen Umstieg von klassischer Virtualisierung auf CBH mit einem LPT zu begleiten, denn die neue Infrastruktur soll ja auch im besten Fall die Performance steigern und keinesfalls negative Auswirkungen haben. Dafür ist es notwendig, eine Baseline noch vor dem Umstieg zu machen, der dann als Vergleichsgrundlage dient. Selten wird nach einem Umstieg auf Anhieb dieselbe Performance wie davor erreicht und eine Reihe an Tests ist notwendig, um die Root Cause zu finden und die Probleme zu lösen. Ein produktiver Einsatz ohne vorherige Lasttests hätte zu massiven Problemen und Ausfällen geführt!

### Vertrauen ist gut, Gewissheit ist besser!

Cloud Anbieter wie AWS, GCP oder Azure Cloud bieten für viele der genannten Features einfache Konfigurationsmöglichkeiten. Auch die Notwendigkeiten, den Master und alle Nodes von 0 auf bereitzustellen und zu servicieren fällt mit dem Einsatz eines Cloud Dienstes weg und damit verringert sich automatisch die Anzahl an Fehlerquellen. Auch die Möglichkeit der Skalierung ist in der Cloud einfacher: Zum Teil ist es gar nicht notwendig, sich mit einzelnen Nodes auseinanderzusetzen, die Cloud bietet auf Knopfdruck so gesehen „unendliche“ Ressourcen (gegen Einwurf von kleinen oder großen Münzen versteht sich). Unternehmen tendieren dann dazu, den Versprechungen und theoretischen Möglichkeiten dieser Cloud Anbieter blind zu vertrauen. Letztendlich ist aber immer noch die richtige Verwendung der Features und das Zusammenspiel der Cloud Ressourcen mit der Applikation ausschlaggebend für den Gesamterfolg, Gewissheit erlangt man also im besten Fall durch Last- und Performancetests, oder aber durch Probleme im Produktivbetrieb.

Container based hosting ist aus der heutigen IT Landschaft nicht mehr wegzudenken und mehr und mehr Unternehmen setzen zu Recht darauf, denn die Vorteile gegenüber klassischer Virtualisierung sind deutlich. Komplexität bedeutet aber immer gleichzeitig erhöhte Risiko und höhere Fehlerwahrscheinlichkeit und damit höheren Testaufwand. Mit Hilfe von Last- und Performancetests wird das Risiko minimiert und der Aufwand insgesamt reduziert, denn sobald ein Problem in Produktion auftritt, sind die Folgekosten jedenfalls höher als jeder Last- und Performancetest.

Weitere Beiträge von dem Experten **Klemens Loschy** finden Sie auf unserer Webseite.



**Klemens Loschy** ist Mitglied des SEQIS Geschäftsleitungsteams und Principal Consultant.

Er kann auf jahrelange Erfahrung in den Bereichen Testautomation, Last-Tests und Performance Engineering, funktionale Tests, Testen in agilen Teams, Anwendungsentwicklung von Testsoftware sowie Beratung und Unterstützung in zahlreichen Projekten unterschiedlichster Branchen zurückblicken.

# Cheat Sheet - Knowhow für Meetings

von SEQIS GmbH

## Im Vorfeld

### Kein Treffen ohne „Arbeitsfrage“

#### Agenda

- Im Vorfeld ausschicken
- Alternativ: LeanCoffee
- Protokoll (inkl. Agenda) erstellen

#### Grund des Treffens

- Sozialer Austausch
- Informationsaustausch
- Entscheidungsmeeting
- Workshop

### Inhaltliche Vorbereitung

#### Spielraum

- Was ist im Scope, was nicht?
- Welches Budget haben wir?

#### Erwartungshaltung und Ziele

- Welche Artefakte und Ergebnisse wollen wir?

#### Stakeholder und Rollen

- Zielgruppe einladen

## Wichtige Softskills

### Kein Top-Down Approach

- Dozieren verboten!

### Interaktion

- Beteiligung schaffen - Alle sind für das Ergebnis verantwortlich

### Auflockerung

- Medien- oder Ortswechsel (Cafeteria)

### Agiler Footprint

- CI
- Stop starting - Start finishing
- Review
- Retro
- Velocity

### Arbeitsfortschritt monitoren

- Visualisierung

### Stimmen hörbar machen

- Zuhören
- Hinhören
- Nachfragen
- Ausreden lassen

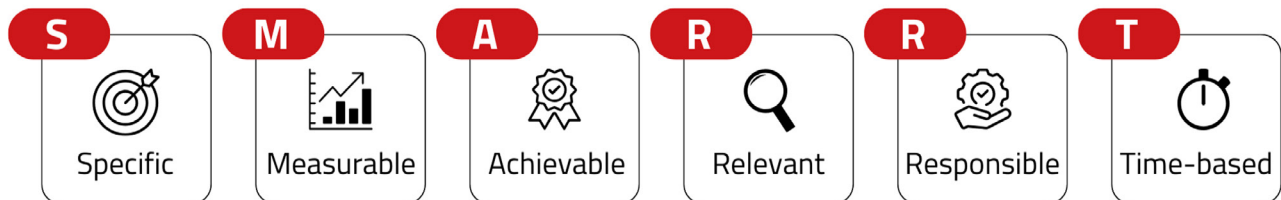


Abbildung: SEQIS GmbH



## Hier treffen sich verschiedene Menschen

- Die Überaktiven
- Die Laberer & Energievampire
- Die Schläfer:innen
- Die Nicht-Vorbereiteten
- Die Abgelenkten
- Die Entschleunigten
- Die Nicht-Zuständigen
- Die Schuldsucher(zuweiser):innen



## Im Meeting

### Orientierung

- Spielregeln - gemeinsam festlegen
- Überblick - auf Folie oder Flipchart
- Inhalte - kurz vorstellen: Ok für alle?
- Ablauf - Begrüßung + Inhalt + Wrap-Up

### Inhalte umsetzen

- Commitment - Termin Engagement
- Timeboxing - Themen (laut Agenda) zeitlich absichern
- Interaktion - Kontakt mit Teilnehmer:innen halten
- Motivation - Take-Aways für jede Person
- Dokumentation - am besten live...

### Meetingziele umsetzen

- Priorisierung - Was ist am wichtigsten?
- Rollendefinition & Ownership - Wer kümmert sich um was?
- Toolchain - Welche Prozesse nutzen wir?
- Kommunikation - Austausch über Fortschritt und mögliche Issues
- Feedback - zum Termin und „Lessions Learned“ Session



Abbildung: (Quelle: undraw.co)

## ... und Lösungstipps

- Positive Feedback-Kultur statt Fingerpointing
- Lösungsorientiert nach vorne
- Solide Vorbereitung (inhaltlich & technisch!)
- Action Points, Ownerships und Deadlines festlegen
- Fokussiert bleiben
- Ausartende Gespräche einfangen



Abbildung: (Quelle: canva.com)

## Tools

- Wecker
- LeanCoffee
- Minutes Template
- Inhalte aus SEQIScience
- Feedback - All-Purpose Single Mentimeter Fragebogen
- Flipchart

## Moderation

- -> siehe „Im Meeting“
- Struktur vorgeben
- Meinungs- und Gedankenaustausch fördern
- Atmosphäre schaffen und/oder absichern
- Visualisierung von Ergebnissen / Artefakten
- nicht: Werten / Meinung vertreten
- nicht unbedingt: Fachwissen haben

Die nächsten Termine im Überblick:



Ten more things

Nachhaltigkeit in der Softwareentwicklung:  
Wurst oder Wahnsinn?

am 26.09.2024



Abbildung: (Quelle: undraw.co)

## September

|  |
|--|
| 1 So   |
| 2 Mo   |
| 3 Di   |
| 4 Mi   |
| 5 Do   |
| 6 Fr   |
| 7 Sa   |
| 8 So   |
| 9 Mo   |
| 10 Di  |
| 11 Mi  |
| 12 Do  |
| 13 Fr  |
| 14 Sa  |
| 15 So  |
| 16 Mo  |
| 17 Di  |
| 18 Mi  |
| 19 Do  |
| 20 Fr  |
| 21 Sa  |
| 22 So  |
| 23 Mo  |
| 24 Di  |
| 25 Mi  |
| 26 Do <small>Ten more things Webinar<br/>Nachhaltigkeit in der Softwareentwicklung:<br/>Wurst oder Wahnsinn?</small> |
| 27 Fr  |
| 28 Sa  |
| 29 So  |
| 30 Mo  |

## Oktober

|   |
|---|
| 1 Di  |
| 2 Mi  |
| 3 Do  |
| 4 Fr  |
| 5 Sa  |
| 6 So  |
| 7 Mo  |
| 8 Di  |
| 9 Mi  |
| 10 Do   |
| 11 Fr   |
| 12 Sa   |
| 13 So   |
| 14 Mo   |
| 15 Di   |
| 16 Mi   |
| 17 Do   |
| 18 Fr   |
| 19 Sa   |
| 20 So   |
| 21 Mo   |
| 22 Di   |
| 23 Mi   |
| 24 Do   |
| 25 Fr   |
| 26 Sa <span style="float: right;">Nationalfeiertag</span> |
| 27 So   |
| 28 Mo   |
| 29 Di   |
| 30 Mi   |
| 31 Do   |

| November |  |
|----------|--|
| 1 Fr     | Allerheiligen  |
| 2 Sa     |  |
| 3 So     |  |
| 4 Mo     |  |
| 5 Di     |  |
| 6 Mi     |  |
| 7 Do     |  |
| 8 Fr     |  |
| 9 Sa     |  |
| 10 So    |  |
| 11 Mo    |  |
| 12 Di    |  |
| 13 Mi    |  |
| 14 Do    | Ten more things Webinar<br>Projekt & Change Management –<br>mit Support durch AI |
| 15 Fr    |  |
| 16 Sa    |  |
| 17 So    |  |
| 18 Mo    |  |
| 19 Di    |  |
| 20 Mi    |  |
| 21 Do    |  |
| 22 Fr    |  |
| 23 Sa    |  |
| 24 So    |  |
| 25 Mo    |  |
| 26 Di    |  |
| 27 Mi    |  |
| 28 Do    |  |
| 29 Fr    |  |
| 30 Sa    |  |

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



Ten more things

Projekt & Change  
Management – mit Support  
durch AI

am 14.11.2024



## IT-Analyse II - Eine Bestandsaufnahme

von Josef Falk



Abbildung 1: (Quelle: SEQIS GmbH)

Im ersten Artikel<sup>[1]</sup> dieser Serie haben wir diskutiert, inwieweit IT-Analyse als eigene Rolle in einem Software-Entwicklungsprojekts sinnvoll ist – und festgehalten, dass das eine Frage der Arbeitsteilung ist.

Darauf aufbauend wollen wir uns in diesem Artikel dem Wesenskern der IT-Analyse von verschiedenen Seiten nähern. Wir werden uns einerseits die zeitliche Komponente ansehen. Seit wann gibt es IT-Analyse? Und wie hat sich ihr Inhalt im Lauf der Zeit geändert? Andererseits wollen wir verschiedene moderne Ansätze über dieses Berufsbild betrachten. Worin unterscheiden sie sich? Wo liegen die jeweiligen Schwerpunkte?

Aus diesen Komponenten soll in weiterer Folge die Beschreibung eines praxistauglichen Berufsbilds der IT-Analyse entstehen.

### IT-Analyse im Lauf der Zeit

Nicht nur im Bereich der IT-Analyse behaupten neue Ansätze von sich, dass sie etwas bieten, was es nie zuvor gab. Als Beispiel sei der Ansatz des „Digital Design“ erwähnt, von dem dessen Autoren sagen, dass die moderne Digitalisierung dieses – bisher nicht vorhandene – Berufsbild erforderlich macht. Stellen wir diese

Aussagen auf den Prüfstand. Zu diesem Zweck blicken wir einmal fast 50 Jahre zurück.

Aus dem Jahr 1976 stammt das Buch „Systemanalyse“ von Hartmut Wedekind<sup>[2]</sup>. Der sehr altmodisch anmutende Untertitel dieses Buches lautet „Die Entwicklung von Anwendungssystemen für Datenverarbeitungsanlagen“.

Hier wird folgender Prozess für den IT-Entwicklungsprozess vorgeschlagen:

1. Ist-Analyse
2. Zielsetzung, Sollkonzept und Rechtfertigung eines neuen Anwendungssystems
3. Systementwurf
4. Systemimplementierung
5. Systembetrieb.

Alle fünf Phasen dieses Modells gemeinsam werden als „Systemanalyse“ verstanden. Das Berufsbild, das dafür zuständig ist, wird „Systemanalytiker“ genannt. Wobei auch hier schon Arbeitsteilung zwischen Entwurf und Schreiben des Codes vorgesehen ist. Es gibt jedoch eine ganz andere Gewichtung als sie heute üblich ist. Die Systemanalyse geht sehr weit ins Detail; der Algorithmus wird so weit spezifiziert, dass „das

Codieren ein einfacher Vorgang“ ist, den „oftmals Hilfskräfte verrichten können“<sup>[3]</sup>. Entwicklung ist in dieser Betrachtungsweise das einfache Umsetzen des vom Systemanalytiker erstellten Entwurfs in Programmcode. Entsprechend detailliert muss der Entwurf sein.

Die Arbeitsteilung zwischen Analyse und Development ist also schon damals angelegt, wenn auch die Grenzen der Aufgaben anders verlaufen als heute.

Man hat auch im Jahr 1976 schon gesehen, dass die einzelnen Phasen im Software-Entwicklungsprozess nicht strikt – im Sinne eines Wasserfalls – voneinander getrennt werden dürfen: „Die Entwicklung eines Systems ist ein iterativer Prozess, in dem sehr viele Rücksprünge auf schon entwickelte Teile und deren Korrektur erforderlich sind“<sup>[4]</sup>.

Im Buch Wirtschaftsinformatik I von Hans Robert Hansen aus dem Jahr 1983 werden einzelne Berufsbilder beschrieben, darunter auch das des Systemanalytikers. Dessen Aufgaben werden unter anderem folgendermaßen zusammengefasst<sup>[5]</sup>:

- Analyse des Istzustandes bestehender Systeme
- Entwicklung von Lösungsvorschlägen und von Sollkonzepten für neue Informationssysteme
- Entwurf der Ausgaben, Eingaben, Dateien und Verarbeitungsalgorithmen für neue Systeme.

Auch hier gibt es also ein Berufsbild, das sich auf den Entwurf und das Erstellen eines Sollkonzeptes spezialisiert – die Aufgaben der Systemanalyse sind von denen der Entwickler personell getrennt. In der Entwicklung wird außerdem noch zwischen



„Anwendungsprogrammierer“ und „Systemprogrammierer“ unterschieden. Die Aufgaben der Anwendungsprogrammierer werden angegeben mit:

- Analyse zu programmierender, vorgegebener, anwendungsbezogener Aufgaben
- Entwicklung einer programmiertechnischen Lösung mit Leistungsspezifikationen wie Speicherbedarf, Maschinenzeit, usw.

Den beiden genannten Ansätzen ist gemeinsam, dass es mit der Systemanalyse ein Berufsbild gibt, das für den Entwurf eines IT-Systems zuständig ist. Der Unterschied zwischen beiden Ansätzen ist, dass der erste den Entwurf deutlich weiter gehend sieht, sodass dieser auch weit in den technischen Bereich hineinreicht, die Aufgabe der Entwicklung ist es nur noch, den Entwurf 1:1 in Code umzusetzen. Der zweite Ansatz beschränkt den Entwurf auf den fachlichen Bereich. Bei der Entwicklung bleibt die Expertise für die technischen Algorithmen und Strukturen.

### IT-Analyse in aktuellen Ansätzen

Im Folgenden werden drei aktuelle Ansätze, die für die IT-Analyse relevant sind, beschrieben:

- Requirements Engineering
- Business Analyse
- Digital Design.

### Requirements Engineering

Der Ansatz des Requirements Engineering stellt einen Begriff in den Vordergrund, der für die IT-Analyse von grundlegender Bedeutung ist: die Anforderung. Die Anforderung ist gewissermaßen Ausgangspunkt und Rohstoff für jede Software-Entwicklung. Es beschreibt das, was das entstehende System können muss. Große Bekanntheit und Verbreitung hat der Ansatz des Requirements Engineering durch die Zertifizierungsprüfungen zum Certified Professional for Requirements Engineering (CPRE) des IREB (International Requirements

Engineering Board) gefunden.

So hat die Berufsbezeichnung „Requirements Engineer“ die frühere Bezeichnung „Systemanalytiker“ nahezu vollständig abgelöst.

Requirements Engineering nach dem CPRE-Lehrplan des IREB kennt vier Haupttätigkeiten:

- Ermitteln von Anforderungen
- Dokumentieren von Anforderungen
- Prüfen/Abstimmen von Anforderungen
- Verwalten von Anforderungen.

Es ist das Verdienst des Requirements Engineering, den Begriff der „Anforderung“, der zuvor nur implizit im IT-Entwicklungsprozess präsent war, in den Fokus gerückt zu haben. Im Gegenzug ist jedoch in diesem Ansatz vom Entwurf des Systems, das in der Systemanalyse noch Hauptzweck der Tätigkeit war, nicht mehr die Rede. Es wird strikt zwischen Anforderung und Lösung entschieden. Wer aus den sauber dokumentierten Anforderungen des Requirements Engineering den Entwurf des Systems macht, bleibt offen. So war es zumindest bis zur Version 2 des Lehrplans.

Die Version 3 des CPRE-Lehrplans<sup>[6]</sup>, die 2020 veröffentlicht wurde, behebt diesen Mangel zum Teil. Hier spielt die Lösung eine wesentlich größere Rolle, als es noch in der Version 2 der Fall war. So wird etwa im fünften Prinzip des Requirements Engineering anerkannt, dass die Begriffe, „Problem“, „Anforderung“, „Lösung“ untrennbar ineinander verwoben sind. Manche Anforderungen treten erst durch die Entwurfsentscheidung für eine bestimmte Lösung auf – andererseits kann eine von einem Stakeholder formulierte Anforderung bei der Entscheidung für eine bestimmte Lösungsalternative überhaupt keine Rolle mehr spielen.

Diese neue Sichtweise kommt auch im Prinzip 8 zum Ausdruck: „Gute Requirements Engineers gehen über das hinaus, was ihre Stakeholder Ihnen sagen“. Es kommt nicht nur auf das genaue Erheben der Anforderungen an, sondern auch um das kreative Finden von neuen – nicht geäußerten – Anforderungen.

Dennoch gibt es auch in der Version 3 nicht die Tätigkeit des Entwurfs in dem Sinne, dass die Anforderungen in ein Lösungskonzept transformiert werden. Es wird in der Begrifflichkeit

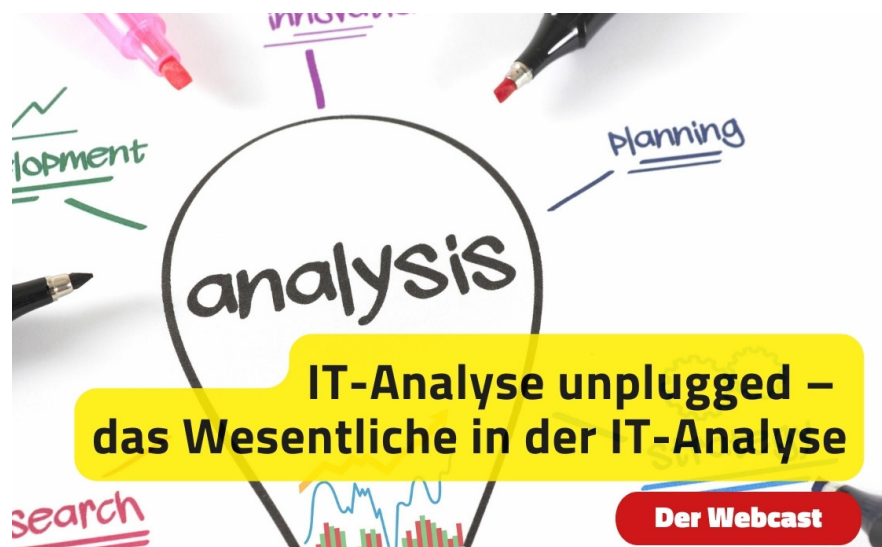


Abbildung 2: analysis Bild (Quelle: EhdäsS - stock.adobe.com)  
Ten more things Webcast mit und von Josef Falk: [www.seqis.com/de/events-index](http://www.seqis.com/de/events-index)

nicht unterschieden, was ist „Anforderung“, die von einem Stakeholder kommt, und was ist eine Entwurfsentscheidung des Analytikers – in beiden Fällen wird von „Anforderung“ gesprochen.

### Business Analyse

Ein weiterer Ansatz, der großen Einfluss auf das Berufsbild der IT-Analyse erlangt hat, ist die Business Analyse. Insbesondere als Job-Title für IT-Analysten hat „Business Analyst“ große Beliebtheit gewonnen – und wird häufig als Synonym für „Requirements Engineer“ verwendet.

Der Inhalt dieses Ansatzes wurde insbesondere vom „International Institute of Business Analysis“ (IIBA) und deren Publikation „A Guide to the Business Analysis Body of Knowledge“ (BABOK) geprägt<sup>[7]</sup>. Der BABOK unterteilt die Tätigkeiten eines Business Analysten in sechs so genannte Knowledge Areas:

1. Business Analysis Planning and Monitoring
2. Elicitation and Collaboration
3. Requirements Life Cycle Management
4. Strategy Analysis
5. Requirements Analysis and Design Definition
6. Solution Evaluation.

Business Analyse ist nicht auf den Bereich der IT beschränkt. Der BABOK definiert Business Analyse als „the practice of enabling change in an enterprise by defining needs and recommending solutions that deliver value to stakeholders“. Diese Definition ist sehr allgemein und ermöglicht so ein sehr weites Anwendungsfeld. Der Nachteil ist aber, dass die beschriebenen Vorgangsweisen und Methoden nicht auf die IT-Analyse zugeschnitten sind und so die Aussagen im BABOK häufig schwammig – „nicht Fisch und nicht Fleisch“ – sind.

Ähnlich wie das Requirements Engineering hat auch die Business Analyse die Anforderungen im Fokus.

Die Tätigkeit des Entwerfens wird auch hier kaum beschrieben, auch wenn mit der Version 3 des BABOK erstmals der Begriff des Designs eingeführt wurde.

### Digital Design

Digital Design ist ein Ansatz, der – wie das Requirements Engineering (RE) – im Umfeld des IREB entstanden ist – gewissermaßen als Reaktion auf die Anforderungs-Zentrierung im Requirements Engineering. Als Begründung dafür werden die neuen Herausforderungen der Digitalisierung angegeben. Tatsächlich ist es aber gleichsam eine Rückkehr zu den Anfängen der IT-Analyse, als auch schon der Entwurf (das Design) im Zentrum der Tätigkeit stand.

Der Inhalt des Digital Design ist ebenso wie das Requirements Engineering in einem Handbuch und einem Lehrplan zu einer Zertifizierungsprüfung festgelegt<sup>[8]</sup>.

Digital Design sieht die erforderlichen Kompetenzen für einen Digital Designer in zwei Bereichen:

**Material-Kompetenz:** Analog zum Bauwesen wird die zur Verfügung stehende digitale Technologie als „Material“ angesehen, das im Design gestaltet wird. Zu diesem Zweck muss der Designer dieses Material kennen.

**Design-Kompetenz:** Um das digitale Material gestalten zu können, muss Know-How über den Design-Prozess vorhanden sein. Der Designer muss die Methoden und Techniken kennen, um ein Design erstellen zu können.

Es sind drei Schritte im Design-Prozess vorgesehen:

**Auftragsklärung:** es soll beim Auftraggeber und unter allen relevanten Stakeholdern ein klares und gemeinsames Verständnis über den Auftrag erzielt werden.



Beschäftigen Sie sich mit dem Thema IT-Analyse? Dann empfehlen wir Ihnen den Vortrag „IT-Analyse unplugged – das Wesentliche in der IT-Analyse“ von Josef Falk. Jetzt ansehen!



**Konzeptarbeit:** Ziel ist, unter allen relevanten Stakeholdern ein ausreichendes Verständnis der Lösung und des zugrundeliegenden technischen Systems zu erarbeiten. Auf Grundlage dieses Verständnisses kann entschieden werden, ob das Risiko einer Realisierung der Lösung eingegangen werden soll oder nicht.

**Entwicklung und Betrieb:** Entwicklung und Betrieb werden im Digital Design als Einheit betrachtet, weil die Lösung laufend weiterentwickelt wird. Hier erfolgt der eigentliche Entwurf. Erst hier werden die Konzepte in einem Detailgrad ausgearbeitet, der die Realisierung der digitalen Lösung ermöglicht.

### IT-Analyse - Grundsätze

Jeder der genannten Ansätze hat einen speziellen Fokus. Im Folgenden sollen die wesentlichen Punkte

zusammengefasst werden:

- Die Basis jeder Software-Entwicklung sind Anforderungen
- Anforderungen können nicht unmittelbar in Software umgesetzt werden
- Im Entwurfsprozess werden die Anforderungen transformiert
- Ziel dieses Prozesses ist ein Entwurf, in dem die Anforderungen konsolidiert und Widersprüche aufgelöst werden und der in Software umgesetzt werden kann
- Ein Entwurf beschreibt Aufbau, Prozesse und Schnittstellen des IT-Systems
- Technische Entscheidungen, wie Auswahl der Programmiersprache, anderer Entwicklungswerkzeuge und deren Anwendung fallen nicht in den Kompetenzbereich der IT-Analyse.

#### Ausblick

In dieser Folge wurden verschiedene Ansätze zu Beschreibung des Berufsbildes IT-Analyse aus Gegenwart und Vergangenheit betrachtet. Auf dieser Basis und ergänzt um eigene langjährige Erfahrung wird im nächsten Artikel das Profil eines praxisnahen Berufsbilds der IT-Analyse beschrieben.

#### Quellen und weiterführende Informationen:

[1] IT-Analyse I - Wer braucht eigentlich IT-Analyse? <https://www.seqis.com/de/blog/wer-braucht-eigentlich-it-analyse>

[2] Wedekind, Hartmut, Systemanalyse – Die Entwicklung von Anwendungssystemen für Datenverarbeitungsanlagen, München, Wien, 1976

[3] Wedekind, Hartmut, a.a.O., S 265

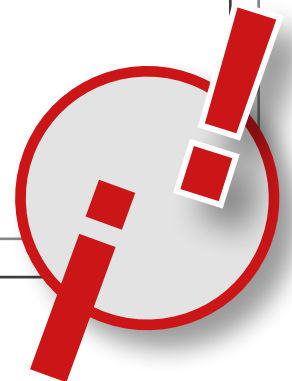
[4] Wedekind, Hartmut, a.a.O, S 17

[5] Hansen, H.R., Wirtschaftsinformatik I, Stuttgart, 1983, S 277 f.

[6] [https://www.ireb.org/content/downloads/3-cpre-foundation-level-handbook/cpre\\_foundationlevel\\_handbook\\_de\\_v1.1.1.pdf](https://www.ireb.org/content/downloads/3-cpre-foundation-level-handbook/cpre_foundationlevel_handbook_de_v1.1.1.pdf)

[7] International Institute of Business Analysis, BABOK – A Guide to the Business Analysis Body of Knowledge, Toronto 2015

[8] <https://www.digitaldesign.org/de/syllabus>



#### Weitere Artikel aus dieser Reihe:

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



**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 Projektstätigkeit befasst er sich mit der Entwicklung der Business Analyse.

# Der EU AI Act - die KI-Verordnung der Europäischen Union zur Regulierung von künstlicher Intelligenz

von Alexander Lewisch



Abbildung 1: (Quelle: Foto von Dušan Cveta-  
nović: <https://www.pexels.com/de-de/foto/blau-bewegung-flagge-europaische-union-12541596/>)

Die Digitalisierung und vor allem der Einsatz neuer Systeme mit Künstlicher Intelligenz (KI) schreiten schnell voran und stellt die gesamte Gesellschaft vor neue Herausforderungen. Neben den weitreichenden Möglichkeiten, die neue KI-Technologien bieten, birgt der damit verbundene technologische Wandel aber auch erhebliche Risiken und einige Ungewissheiten. Aus diesem Grund hat die Europäische Kommission am 21. April 2021 einen Vorschlag zur Festlegung harmonisierter Vorschriften für Künstliche Intelligenz in der Europäischen Union veröffentlicht.

## Der EU AI Act<sup>[1]</sup> - 3 Jahre zwischen Vorschlag und Einigung

Drei Jahre nach der Veröffentlichung des Kommissionsvorschlags konnten das Europäische Parlament und der Europäische Rat am 21. Mai 2024 eine Einigung über den EU AI Act (AI ist die Abkürzung für "Artificial Intelligence"), eine Verordnung zur Regulierung von künstlicher Intelligenz in der Europäischen Union, erzielen. Der EU AI Act ist somit das erste umfassende Regelwerk (113 Artikel) für künstliche Intelligenz weltweit. Die EU will damit eine rechtliche Grundlage für die Entwicklung und den Einsatz von

KI schaffen, um mögliche Risiken zu minimieren und daraus folgende Schäden abzuwenden. Der Rechtsrahmen hat aber nicht nur eine Vereinheitlichung der Rechtsvorschriften zum Ziel, sondern soll auch das Fundament für die Entfaltung der neuen Technologie sein, um Investitionen, Innovationen und kreative Lösungen im Bereich der KI zu fördern.

### Struktur des EU AI Acts<sup>[3]</sup>

Kapitel I: Allgemeine Bestimmungen

Kapitel II: Verbotene Praktiken der künstlichen Intelligenz

Kapitel III: Hochriskantes KI-System

Kapitel IV: Transparenzverpflichtungen für Anbieter und Betreiber von bestimmten KI-Systemen und GPAI-Modellen

Kapitel V: AI-Modelle für allgemeine Zwecke

Kapitel VI: Maßnahmen zur Unterstützung der Innovation

Kapitel VII: Governance

Kapitel VIII: EU-Datenbank für hochriskante KI-Systeme

Kapitel IX: Überwachung nach dem Inverkehrbringen, Informationsaustausch, Marktüberwachung

Kapitel X: Verhaltenskodizes und Leitlinien

Kapitel XI: Befugnisübertragungen und Ausschussverfahren

Kapitel XII: Vertraulichkeit und Sanktionen

Kapitel XIII: Schlussbestimmungen

Anhang I: Liste der Harmonisierungsrechtsvorschriften der Union

Anhang II: Liste der Straftatbestände

Anhang III: Hochriskante AI-Systeme

Anhang IV: Technische Dokumentation

Anhang V: EU-Konformitätserklärung

Anhang VI: Konformitätsbewertungsverfahren auf der Grundlage der internen Kontrolle

Anhang VII: Konformität aufgrund der Bewertung des Qualitätsmanagementsystems und der Bewertung der technischen Dokumentation

Anhang VIII: Informationen, die bei der Registrierung von AI-Systemen mit hohem Risiko vorzulegen sind (Artikel 49)

Anhang IX: Informationen, die bei der Registrierung von AI-Systemen mit hohem Risiko vorzulegen sind (Anhang III / Artikel 60)

Anhang X: Rechtsvorschriften der Union über IT-Großsysteme im Raum der Freiheit, der Sicherheit und des Rechts

Anhang XI: Technische Unterlagen gemäß Artikel 53 Absatz 1a



Anhang XII: Transparenzinformationen gemäß Artikel 53 Absatz 1b

Anhang XIII: Kriterien für die Bestimmung von AI-Modellen für allgemeine Zwecke mit systemischem Risiko

### Was ist eine Verordnung?

In der EU ist eine Verordnung ein Rechtsakt, der in den Mitgliedstaaten unmittelbare Geltung hat und nicht erst wie eine Richtlinie durch innerstaatliche Gesetze umgesetzt werden muss<sup>[2]</sup>

Im AI Act versucht die EU, sowohl die Interessen von Unternehmen und staatlichen Behörden, die KI-Systeme innerhalb der EU anbieten und einsetzen, als auch die Grundrechte und Interessen der EU-Bürger und EU-Bürgerinnen, die mit KI-basierten Anwendungen in Berührung kommen, zu berücksichtigen und in Einklang zu bringen.

Gemäß Artikel 2 des AI Acts gelten die Regelungen für alle innerhalb der EU, unabhängig vom Sitz des Betreibers oder dem Ort, wo die KI-basierten Systeme hergestellt oder betrieben werden. Außerdem gelten die Regeln nicht für Systeme, die in der EU entwickelt wurden, aber außerhalb der EU zum Einsatz kommen. Die Verordnung hat keine Gültigkeit für:

- KI-Systeme, die nicht in den Anwendungsbereich des Unionsrechts fallen, wie z.B. die Zuständigkeiten der einzelnen Mitgliedstaaten im Bereich der nationalen Sicherheit
- KI-Systeme, wenn diese ausschließlich für militärische, verteidigungspolitische oder nationale Sicherheitszwecke angewendet, betrieben bzw. verändert oder unverändert genutzt werden

- KI-Systeme, die von ausländischen Behörden oder internationalen Organisationen für die Strafverfolgung und die justizielle Zusammenarbeit eingesetzt werden, sofern sie nicht gegen die Rechte des Einzelnen verstoßen
- KI-Systeme, die für wissenschaftliche Forschung und Entwicklung eingesetzt werden
- KI-Systeme, die noch gar nicht auf dem Markt sind
- Einzelpersonen, die KI-Systeme für persönliche, nicht berufliche Aktivitäten nutzen
- sowie für KI-Systeme, die unter Open-Source-Lizenzen veröffentlicht werden, außer sie sind hochriskant oder fallen daher unter bestimmte Einschränkungen oder Schutzbestimmungen der Verordnung.<sup>[4]</sup>

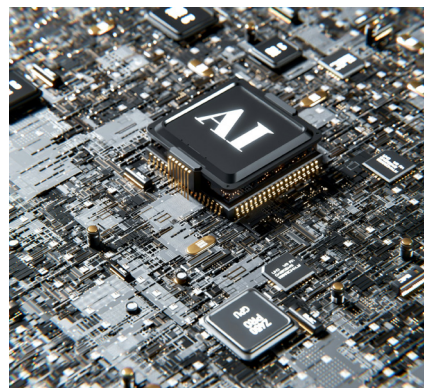


Abbildung 2: (Quelle: Foto von Igor Omilaeu auf Unsplash)

### Ein risikobasierter Ansatz zur Bewertung von KI-Systemen

Den Kern der KI-Verordnung bildet die Klassifizierung der KI-Systeme nach Risikoklassen. Dieser risikobasierte Ansatz bewertet die KI-Systeme nach ihrem Risiko für die Sicherheit, Gesundheit und Grundrechte von Menschen. Dabei werden im AI Act vier Risikostufen definiert und voneinander abgegrenzt.<sup>[5]</sup>

#### Stufe 1: KI-Verbote - unannehmbares Risiko

In Artikel 5 des zweiten Kapitels werden KI-Systeme aufgezählt, die gemäß der Verordnung ein unannehmbares Risiko darstellen und

daher verboten sind. Dazu zählen KI-Systeme, die

- unterschwellige, manipulative oder täuschende Techniken einsetzen, um Entscheidungen bzw. das Verhalten von Personen gezielt zu manipulieren, wodurch mit hoher Wahrscheinlichkeit ein erheblicher Schaden für die Betroffenen entstehen kann.
- Schwächen von Personen oder ganzen Personengruppen (z.B. Alter, Behinderungen, sozioökonomische Verhältnisse) nutzen, um ein bestimmtes Verhalten zu bewirken und die daraus entstandenen Schäden bewusst in Kauf nehmen.
- biometrische Kategorisierungssysteme beinhalten, die Rückschlüsse auf sensible Merkmale wie Rasse, politische Meinungen, Gewerkschaftszugehörigkeit, religiöse/philosophische Überzeugungen oder sexuelle Ausrichtung zulassen. Ausgenommen sind rechtmäßig erworbene biometrische Datensätze wie z.B. Bilder oder die Kategorisierung biometrischer Daten durch die Strafverfolgungsbehörden.
- zur nachteiligen Bewertung oder Klassifizierung von Personen oder Personengruppen nach Sozialverhalten oder persönlichen Eigenschaften eingesetzt werden.
- die zur Risikobewertung dienen, um Profile zu erstellen, ob und wie wahrscheinlich natürliche Personen Straftaten begehen werden.
- die Datenbanken zur Gesichtserkennung aus dem Internet oder aus Aufnahmen der Videoüberwachung erstellen oder erweitern.
- zum Zweck eingerichtet werden, um Emotionen am Arbeitsplatz oder in Bildungseinrichtungen ableiten zu können, ausgenommen sie werden bewusst aus medizinischen oder Sicherheitsgründen eingesetzt.
- zum Zweck der Strafverfolgung

eine biometrische Fernidentifizierung in Echtzeit in öffentlich zugänglichen Räumen verwenden. Dies gilt nicht für bestimmte Tatbestände, wie die Suche nach vermissten Personen, Entführungsopfern, Opfer von Menschenhandel oder sexueller Ausbeutung, die Verhinderung einer erheblichen und unmittelbaren Bedrohung des Lebens oder eines vorhersehbaren terroristischen Angriffs sowie die Identifizierung von Verdächtigen bei schweren Straftaten.

### Stufe 2: KI-Systeme mit hohem Risiko

Die Einstufung von KI-Systemen mit hohem Risiko nimmt einen beträchtlichen Teil der Verordnung ein. Abgesehen von den Artikel 6 bis 27 in den Abschnitten 1 bis 3 des dritten Kapitels, bezieht sich der Anhang I auf die Harmonisierungsrechtsvorschriften der Union, die in Artikel 6 festsetzt, unter welchen Bedingungen ein KI-System mit hohem Risiko eingestuft wird. In Anhang III werden AI-Systeme mit hohem Risiko im Sinne des Artikel 6 Absatz 2 aufgezählt. Demnach sind hochriskante AI-Systeme, wenn sie zu den Bereichen Biometrie, kritische Infrastrukturen, allgemeine und berufliche Bildung, Beschäftigung, Beschäftigung, Arbeitnehmermanagement, Zugang zur Selbstständigkeit, Dienstleistungen, Strafverfolgung, Migration und Justiz zählen. Auch werden Fälle der biometrischen Identitätsprüfung, zur Aufdeckung von Finanzbetrug oder zur Organisation politischer Kampagnen als Ausnahme, genannt.

Für als hoch riskant eingestufte KI-Systeme müssen die Anbieter und Betreiber, gemäß den Artikeln 8 bis 17, umfangreiche Verpflichtungen hinsichtlich Risikomanagement, Daten- und Datenverwaltung, technische Dokumentation, Transparenz, Protokollierung, menschliche Überwachung, Konformitätsbewertung und Cybersicherheit erfüllen. In Folge hat im Zusammenhang mit KI-Sys-

temen zusätzlich auch immer eine Bewertung der Auswirkungen auf die Grundrechte zu erfolgen.

### Stufe 3: KI-Systeme mit spezifischem Risiko und besonderen Transparenzpflichten

Eine weitere Risikoklassifizierung betrifft Anbieter bestimmter KI-Systeme und GPAI-Modelle (GPAI steht für General Purpose AI), die in Artikel 50 zur Transparenz verpflichtet werden und ihre Nutzer über Interaktionen mit KI-Systemen informieren müssen, wie z.B. bei Chatbots oder Deepfakes. Dazu zählen auch biometrische Systeme, sofern sie nicht unter die verbotenen Systeme fallen.

In Artikel 53 enthält der AI Act spezielle Regeln für die Entwicklung und den Einsatz von GPAI-Modellen. Die Anbieter und Entwickler dieser KI-Modelle sind verpflichtet, detaillierte Aufzeichnungen über die Entwicklung und die Tests ihrer KI zu führen. Diese Informationen müssen mit anderen Unternehmen geteilt werden, die diese nutzen wollen. Ausgenommen von dieser Regelung sind KI-Modelle, die quelloffen sind, außer sie stellen ein systemisches Risiko dar. Die Anbieter von GPAI-Modellen verpflichten sich außerdem zur Zusammenarbeit mit der Kommission und den zuständigen nationalen Behörden.

### General Purpose AI (GPAI)<sup>[6]</sup>

Ein **GPAI-Modell** ist ein KI-Modell, auch wenn es mit einer großen Datenmenge unter Verwendung von Selbstüberwachung in großem Maßstab trainiert wurde, das eine signifikante Allgemeinheit aufweist und in der Lage ist, ein breites Spektrum unterschiedlicher Aufgaben kompetent auszuführen, unabhängig davon, wie das Modell auf den Markt gebracht wird, und das in eine Vielzahl von nachgelagerten Systemen oder Anwendungen integriert werden kann.

Ein **GPAI-System** ist ein KI-System, das auf einem allgemeinen KI-Modell basiert, das für eine Vielzahl von Zwecken eingesetzt werden kann, sowohl zur direkten Verwendung als auch zur Integration in andere KI-Systeme.

### Stufe 4: KI-Systeme mit minimalem Risiko

Die große Mehrheit der KI-Systeme fällt in die vierte Stufe mit dem geringsten Risiko. Diese KI-Anwendungen sind bisher nahezu unreguliert. Hierzu zählen beispielsweise KI-gestützte Videospiele oder Spam-Filter. Anbieter dieser Systeme können sich freiwillig zu Verhaltenskodizes beken-

**Fristen** für das Inkrafttreten der AI Act-Bestimmungen gemäß Artikel 113.<sup>7</sup>

- **Juni-Juli 2024:** Der EU Ai Act wird im Amtsblatt der EU veröffentlicht
- **20 Tage später** nach Veröffentlichung tritt die Verordnung in Kraft
- **6 Monate später** finden Kapitel I und Kapitel II (Verbote von AI) Anwendung
- **12 Monate später** gelten die Kapitel III, Kapitel V, Kapitel VII, Kapitel VII, Kapitel XII
- **24 Monate später** treten die meisten übrigen Bestimmungen in Geltung
- **36 Monate später** gelten Artikel 6 Absatz 1 und die entsprechenden Verpflichtungen

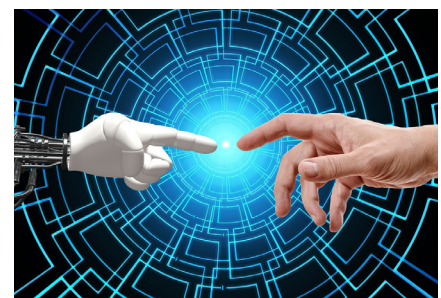


Abbildung 3: (Quelle: Bild von Gerd Altmann auf Pixabay)

Bei Verstößen drohen Unternehmen empfindliche **Geldbußen**<sup>[8]</sup>:

- Die Bußgelder würden zwischen 35 Mio. EUR oder 7 % des weltweiten Jahresumsatzes (je nachdem, welcher Betrag höher ist) für Verstöße gegen verbotene AI-Anwendungen, 15 Mio. EUR oder 3 % für Verstöße gegen andere Verpflichtungen, und 7,5 Mio. EUR oder 1,5 % für die Übermittlung falscher Informationen liegen.
- Für KMU und Start-ups sollen bei Verstößen gegen die Verordnung verhältnismäßige Obergrenzen für Bußgelder vorgesehen werden.

### Quo Vadis - Künstliche Intelligenz in der EU?

Ähnlich der DSGVO (Datenschutz-Grundverordnung) hat die Europäische Union mit dem EU AI Act wieder ein umfassendes Regelwerk geschaffen, um den Einsatz künstlicher Intelligenz zu regulieren. Die EU möchte damit der rasanten Entwicklung auf dem Gebiet der KI Rechnung tragen und den möglichen Nachteilen und Risiken zuvorkommen und Gefahren entschlossen entgegenwirken. Die getroffenen Beschlüsse sind aber nur der Beginn vieler weiterer Maßnahmen, die aufgrund der künftigen Entwicklungen und damit verbundenen Herausforderungen gesetzt werden müssen. Während die einen über die Initiative der EU zur Regulierung der KI bei gleichzeitigem Schutz der Grundrechte erleichtert und zufrieden sind, sehen andere in der Verordnung ein bereits überholtes Regelwerk, das sich zum Nachteil im globalen Wettbewerb, vor allem mit den USA und China, entwickeln könnte. Trotzdem gehen sowohl Datenschützer und Grundrechtsexperten als auch Vertretern der Wirtschaft die Bestimmungen aus ihrer Sicht nicht weit genug bzw. zu weit. Beide Seiten zweifeln an der Zukunftsfähigkeit der

bestehenden Verordnung. Besonders kritisch werden die Fristen gesehen, weil die meisten Regeln erst 2026 in Kraft treten werden. Dies wird dazu führen, dass aufgrund der schnellen Entwicklungen auf dem Gebiet der KI, eine Vielzahl der Regeln schon wieder überholt sein wird.

Die Praxis wird weisen, wo mögliche Schlupflöcher in den Bestimmungen liegen, ob die Risiken hinsichtlich der Wahrung von Grundrechten wirklich minimiert werden können, welche Auswirkungen die Verordnung für Anbieter und Betreiber von KI-Systemen haben werden und wie sich die Rolle Europas im globalen Wettbewerb entwickeln wird.

#### Quellen und weiterführende Informationen:

<sup>[1]</sup> Europäische Kommission: Vorschlag für eine Verordnung des Europäischen Parlaments und des Rates zur Festlegung harmonisierter Vorschriften für künstliche Intelligenz (Gesetz über künstliche Intelligenz) und zur Änderung bestimmter Rechtsakte der Union. COM(2021) 206 final 2021/0106 (COD), Brüssel, den 21.4.2021. In: [https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0019.02/DOC\\_1&format=PDF](https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0019.02/DOC_1&format=PDF)

<sup>[2]</sup> Bundeskanzleramt Österreich: <https://www.oesterreich.gv.at/lexicon/V/Seite.990040.html>

<sup>[3,4]</sup> EU Artificial Intelligence Act: <https://artificialintelligenceact.eu/de/AI-Act-explorer/>

<sup>[5,8]</sup> LexisNexis Verlag: Einigung beim EU AI ACT: So soll künstliche Intelligenz reguliert werden. In: <https://www.lexisnexis.at/blog-post/einigung-beim-eu-ai-act-so-soll-kuenstliche-intelligenz-reguliert-werden/>

<sup>[6]</sup> EU Artificial Intelligence Act: <https://artificialintelligenceact.eu/de/high-level-summary/>

<sup>[7]</sup> EU Artificial Intelligence Act: <https://artificialintelligenceact.eu/de/entwicklungen>



**Alexander Lewisch** ist Consultant bei SEQIS.

Als IT-Quereinsteiger und Wiedereinsteiger konnte er Erfahrungen im Software Testing sammeln und Einblicke in diverse Bereiche wie Testmanagement, Testdurchführung, Testautomation, Testkoordination, Reporting und Consulting sammeln.

Derzeit widmet er sich verschiedenen Testprojekten im Bereich der Remote Testing Services (RTS).

## 5 Fragen an Hansjörg Münster

von Hansjörg Münster



Abbildung 1: (Quelle: undraw.co)

### Mit welchen Branchen hast du in den letzten Jahren zusammengearbeitet und war/ist IT-Modernisierung dort ein Thema?

In den letzten Jahren konnte ich Erfahrungen in der Zusammenarbeit mit verschiedenen Branchen sammeln, in denen die IT-Modernisierung eine wichtige Rolle spielte und heute noch spielt. Neben einigen IT-Unternehmen, für die kontinuierliche Modernisierung der IT-Systeme selbstverständlich ist, habe ich auch zwei Webshops bei der Modernisierung ihrer ERP-Software aus Qualitätssicht begleitet.

Die Ausgangssituation bei den Webshops war jedoch deutlich anders als bei den IT-Unternehmen. In beiden Fällen wurde die ERP-Software über einen längeren Zeitraum nicht weiterentwickelt und neue Versionen des Herstellers wurden nicht eingespielt. Auch neue Entwicklungen in der IT-Infrastruktur, wie zum Beispiel die Cloud-Technologie, wurden nicht oder nur unzureichend umgesetzt.

Diese mangelnde Modernisierung führte zu hohen Kosten, einem langwierigen und komplexen Umstellungsprozess sowie einem erhöhten Risiko bei der Einführung eines neuen Systems.

Zusammenfassend lässt sich sagen, dass die IT-Modernisierung ein Thema ist, das in allen Branchen relevant ist. Der Grad der Dringlichkeit und die konkreten Herausforderungen können jedoch je nach Branche und Unternehmen stark variieren.

### Wie gehst du mit Widerständen oder Bedenken von Unternehmen um, die neuen Technologien während der Modernisierung gegenüber skeptisch sind?

Bei der Einführung neuer Technologien in Modernisierungsprojekten ist es üblich, dass Unternehmen Bedenken oder Skepsis äußern. Um diese Widerstände erfolgreich zu adressieren, setzt man am besten auf sachliche Argumente, belastbare Zahlen und konkrete Beispiele.

Zunächst ist es wichtig, die Bedenken der Unternehmen ernst zu nehmen und zu verstehen. Dazu führt man offene und ehrliche Gespräche mit allen Stakeholdern, um ihre individuellen Herausforderungen und Bedürfnisse zu erfassen.

Wichtig sind konkrete Argumente, die den Mehrwert der neuen Technologien belegen:

- Geringere Kosten: Moderne Systeme unterstützen den Fachbereich in deren Prozessen besser, benötigen weniger Wartung und können Lizenzkosten senken.
- Erhöhte Sicherheit: Neue Technologien bieten fortschrittliche Sicherheitsfunktionen, die Unternehmen besser vor Cyberangriffen schützen können.

- Verbesserte Effizienz und Produktivität: Neue Technologien können Prozesse automatisieren, manuelle Tätigkeiten reduzieren und die Zusammenarbeit verbessern, was zu einer Steigerung der Effizienz und Produktivität führt.
- Verbesserte Skalierbarkeit und Flexibilität: Moderne IT-Infrastrukturen sind skalierbar und anpassungsfähig, insbesondere Cloudbasierte System sind rasch anpassbar.
- Innovationspotenzial: Neue Technologien eröffnen neue Geschäftsmöglichkeiten und ermöglichen Unternehmen, wettbewerbsfähig zu bleiben

Um diese Argumente zu untermauern, nutzt man belastbare Zahlen, Erfolgsgeschichten und Best Practices:

- Fallstudien von Unternehmen, die ähnliche Modernisierungsprojekte erfolgreich durchgeführt haben.
- Kennzahlen, die die positiven Auswirkungen der neuen Technologien auf die Geschäftsleistung belegen.
- Expertenmeinungen und Branchenanalysen, die den Wert der neuen Technologien bestätigen.

Das wichtigste Argument sind in der Regel die Kosten. Um in dieses Thema Transparenz zu bringen, erstellt man am besten eine nachvollziehbare, detaillierte und belastbare Kosten-Nutzen-Analyse (ROI-Berechnung), die alle relevanten Kosten und potenziellen Einsparungen berücksichtigt.



Für Unternehmen, die sich Agilität auf die Fahnen geschrieben haben, ist eine kontinuierliche Verbesserung und Modernisierung der IT selbstverständlich. Moderne Technologien ermöglichen es diesen Unternehmen, schneller auf Marktveränderungen zu reagieren, neue Produkte und Dienstleistungen zu entwickeln und ihre Wettbewerbsfähigkeit zu stärken.

Zusammenfassend lässt sich sagen, dass ein offener und transparenter Kommunikationsstil, gepaart mit sachlichen Argumenten, belastbaren Zahlen und konkreten Beispielen, der Schlüssel zum Erfolg bei der Bewältigung von Widerständen gegen neue Technologien ist.

### Welche Rolle spielt deiner Meinung nach die persönliche Weiterentwicklung und das kontinuierliche Lernen in der IT-Branche?

In der dynamischen Welt der IT sind persönliche Weiterentwicklung und kontinuierliches Lernen nicht nur wichtig, sondern unerlässlich für den Erfolg. Die Gründe dafür sind vielfältig:

#### 1. Rasante technologische Entwicklung:

Die IT-Branche entwickelt sich in einem atemberaubenden Tempo. Neue Technologien, Programmiersprachen, Frameworks und Tools entstehen ständig. Um auf dem neuesten Stand und um relevant zu bleiben, müssen IT-Fachkräfte kontinuierlich lernen und ihre Fähigkeiten aktualisieren.

#### 2. Steigender Wettbewerb:

Der Wettbewerb um qualifizierte IT-Fachkräfte ist hart, auch wenn ein Fachkräftemangel herrscht. Unternehmen suchen talentierte Mitarbeiter mit aktuellem Wissen und Fähigkeiten, um in den prestigeträchtigsten Positionen und an den „coolsten“ Projekten zu arbeiten. Weiterentwicklung und kontinuierliches Lernen sind daher entscheidend, um

wettbewerbsfähig zu bleiben und die besten Karrierechancen zu nutzen.

#### 3. Erhöhte Anforderungen:

Die Anforderungen an die IT und die IT-Fachkräfte steigen stetig. Unternehmen benötigen Mitarbeiter, die über ein breites Spektrum an Hard- und Soft Skills verfügen, sich schnell an neue Gegebenheiten anpassen können und komplexe Probleme lösen können. Kontinuierliches Lernen ermöglicht es, diese Anforderungen zu erfüllen und sich für neue Aufgaben zu qualifizieren.

#### 4. Karrierechancen und Gehaltsentwicklung:

IT-Fachkräfte, die sich weiterentwickeln und neue Fähigkeiten erlernen, haben bessere Karrierechancen und können höhere Gehälter erzielen. Unternehmen sind bereit, mehr für Mitarbeiter zu zahlen, die über die neuesten Kenntnisse und Fähigkeiten verfügen.

#### 5. Persönliche Weiterentwicklung und Motivation:

Lernen hält nicht nur die Fähigkeiten auf dem neuesten Stand, sondern fördert auch die persönliche Weiterentwicklung und Motivation. Wer sich neuen Herausforderungen stellt und sein Wissen erweitert, ist zufrieden im Job und gestaltet die Zukunft des Unternehmens mit.

Zusammenfassend lässt sich sagen, dass die persönliche Weiterentwicklung und das kontinuierliche Lernen in der IT-Branche der Schlüssel zum Erfolg sind. Wer diese Aspekte vernachlässigt, riskiert, den Anschluss zu verlieren und seine Karrierechancen zu schmälern.

### Inwiefern hat die vermehrte Nutzung von Remote-Arbeit in den letzten Jahren die Anforderungen an die IT-Infrastruktur beeinflusst?

Die vermehrte Nutzung von Remote-Arbeit hat in den letzten Jah-

ren die Anforderungen an die IT-Infrastruktur in Unternehmen erheblich verändert. Durch die COVID-19-Pandemie und die damit verbundenen Lockdowns mussten Unternehmen innerhalb kürzester Zeit auf Remote-Arbeit umstellen. Dies stellte die IT-Abteilungen vor große Herausforderungen, da sie schnellstmöglich Online-Meetings, Collaboration-Tools und Remote-Zugriffe auf die Unternehmens-IT bereitstellen mussten. Die vermehrte Nutzung von Remote-Arbeit ist mittlerweile zum Standard geworden und hat die Anforderungen an die IT-Infrastruktur nachhaltig verändert:

- Collaboration-Tools sind unverzichtbare Tools für die Zusammenarbeit: Online Meeting, Chaträume, gemeinsame Dokumentation und gleichzeitiges Bearbeiten von Dokumenten. Allerdings muss die lokale IT diese Tools unterstützen, warten, ausrollen und nahtlos in die bestehende Toolchain integrieren.
- Cloud-Lösungen entlasten die internen lokalen IT-Abteilungen und bieten oft eine erhöhte Sicherheit. Allerdings erfordert die Einführung von Cloud-Lösungen oft einen hohen Aufwand für die IT: Installation am Client, Aufbau von Strukturen, Schulung der Anwender. In großen Unternehmen kann dies rasch zu einem Großprojekt führen.
- Remotes Arbeiten erhöht den Bedarf an Netzwerkkapazität. Remote-Mitarbeiter greifen von verschiedenen Standorten aus auf Unternehmensressourcen zu, was zu einem erhöhten Datenverkehr im Netzwerk führt. Die IT muss diese Kapazitäten bereitstellen, wofür sie dauerndes Monitoring, Planungen und Anpassungen vornehmen muss. Meist ist jedoch der Engpass in letzten Meile: der Internetanbindung in den Heimarbeitsplät-

zen. Eine Familie mit mehreren Personen im Home-Office und Homeschooling kann die Internetkapazität des Heiman schlusses schnell an die Grenzen bringen.

Remotes Arbeiten bietet jedoch auch viele Vorteile: Die Zufriedenheit der Mitarbeiter wächst, Fahrten von und zum Arbeitsplatz entfallen teilweise und es bietet eine Reihe von neuen Geschäftsmöglichkeiten. Zwar gab es die „digitalen Nomaden“ schon länger, aber heute ist dieses Konzept in vielen Branchen in der Breite angekommen. SEQIS „RTS - Remote Testing Services“: SEQIS hat die Potenziale von Remote-Arbeit erkannt und bietet mit „RTS - Remote Testing Services“ ein neues Service an. Das Service ermöglicht es Kunden, Testaktivitäten wie manuelles Testen, Testfallerstellung und Testautomatisierung rasch an SEQIS zu übergeben. Dieses Service ist ausschließlich remote. Dadurch können wir es günstiger anbieten und haben bereits internationale Kunden gewinnen können.

Zusammenfassend zeigt sich, dass die vermehrte Nutzung von Remote-Arbeit die IT-Infrastruktur vor neue Herausforderungen stellt, bietet aber auch neue Möglichkeiten. Die IT-Infrastruktur muss stetig den wechselnden und erhöhten Anforderungen angepasst werden. Allerdings bietet remotes Arbeiten auch vielfältige Vorteile und neue Geschäftsfelder.

### Gibt es bewährte Strategien, um den Übergang der Mitarbeiter zu neuen Technologien reibungslos zu gestalten?

Ja, es gibt solche Strategien. Es geht um Ängste und Bedenken von Mitarbeitern, durch die der Projekterfolg wegen mangelnder Akzeptanz verwehrt bleibt. Gegenstrategien, die helfen:

- Offene und transparente Kommunikation: Informieren Sie die

Mitarbeiter frühzeitig und umfassend über die bevorstehenden Veränderungen. Machen Sie die Ziele deutlich und erklären den Nutzen neuer Technologien und Systeme. Nehmen Sie die Ängste, Bedenken und Einwände ernst und lassen diese gelten. Jeder Mitarbeiter soll das Recht haben, jederzeit Fragen zu stellen dürfen und diese auch ehrlich beantworten zu bekommen.

- Einbeziehung der Mitarbeiter: Nehmen Sie gerade skeptische Mitarbeiter in das Projekt mit. Das beugt Gerüchten, Fehlinformationen und Verbreitung von schlechter Stimmung in der Belegschaft vor. Ermutigen Sie Mitarbeiter, ihre Ideen und Vorschläge einzubringen und geben Sie den Mitarbeitern die Möglichkeit sich frühzeitig mit den Technologien auseinander zu setzen und diese auszuprobieren.
- Schulung und Training: Natürlich gehört auch eine gute Schulung vor dem GO-Live und Support danach dazu. Ein mit seinen Problemen alleine gelassener Kollege hat es „ja eh schon vorher gewusst“ und lernt nur, dass Neues nicht mit „Gut“ und Fortschritt nicht mit Verbesserung

gleich zu setzen ist. Wie er es ja schon seit je her weiß.

- Change Management: All dies bewerkstelligen Sie am besten durch aktives Change Management. Verwenden Sie die Toolbox der Change manager und steuern, damit aktiv den Übergangsprozess. Überwinden Sie die Widerstände gegen das Projekt, gegen neue Prozesse und neue Technologien, indem Sie sich aktiv damit auseinandersetzen. Die Einbeziehung eines erfahrenen Change Managers kann dabei hilfreich sein, dann verfügen auch Sie im Projekt über das notwendige Wissen und die Erfahrung, um den Übergangsprozess effektiv zu planen, zu steuern und zu begleiten.

Zusammenfassend zeigt sich, dass die vermehrte Nutzung von Remote-Arbeit die IT-Infrastruktur vor neue Herausforderungen stellt, bietet aber auch neue Möglichkeiten. Die IT-Infrastruktur muss stetig den wechselnden und erhöhten Anforderungen angepasst werden. Allerdings bietet remotes Arbeiten auch vielfältige Vorteile und neue Geschäftsfelder.

Der Text stammt von mir, Google Gemini hat ihn grammatikalisch und sprachlich korrigiert.



**Hansjörg Münster** ist Principal Consultant und Teamlead bei SEQIS.

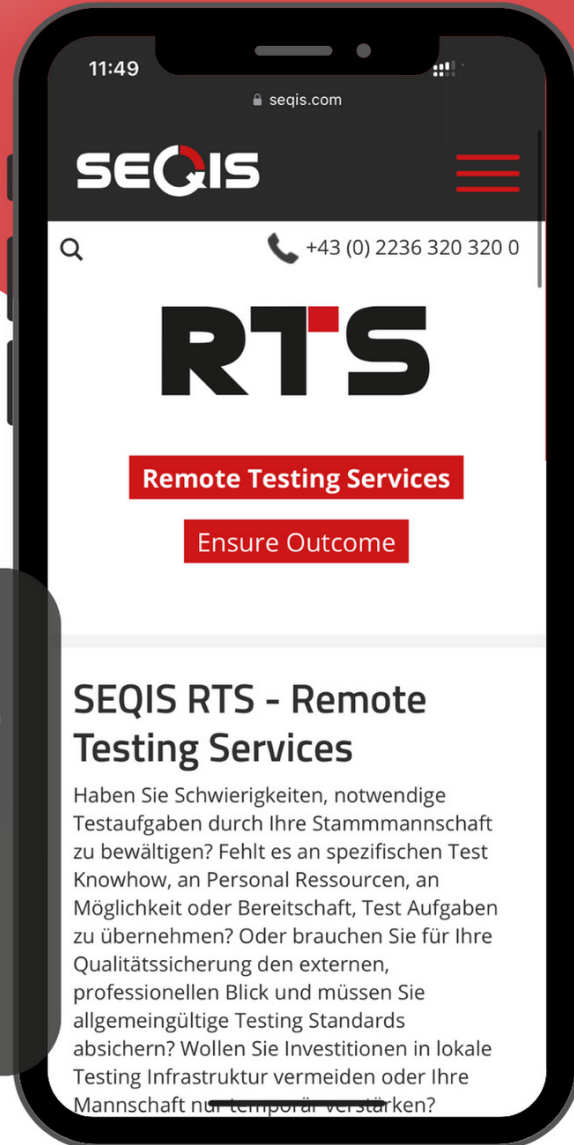
Als Allrounder deckt er ein breites Spektrum an Aufgaben ab. Die Schwerpunkte seiner Tätigkeit liegen in den Bereichen Test Management, Testautomation und Lasttest.

Ganz oben auf der Prioritätenliste des IT Profis steht, einen Nutzen und Mehrwert in der Qualitätssicherung seiner IT Projekte zu generieren.



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

Kontaktieren Sie mich unter  
[alexander.weichselberger@SEQIS.com](mailto: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.

# Test Automation: Paving the Way for IT Modernization

von Maged Elsherif

## What is test automation?

Before exploring the future of test automation, it's essential to first understand what test automation entails and the benefits it brings to the table.

According to the ISTQB Glossary "Test automation is the use of software to perform or support test activities".

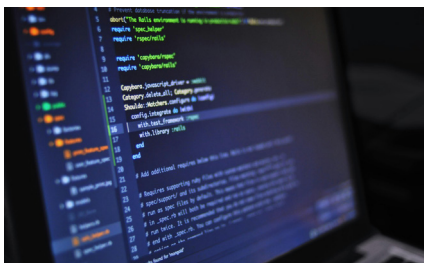


Abbildung 1: (Quelle: Foto von luis gomes auf Pexels)

## How can test automation be beneficial?

Test automation aims to simplify test execution and analysis, saving time and costs while ensuring consistent results. It expands code coverage, allowing for thorough examination of functionalities, especially in early development stages, leading to better software quality. Unlike manual testing, which consumes significant time and resources, test automation provides quick results and immediate feedback, enabling developers and testers to promptly act on the findings. By automating repetitive tasks, resources are freed up, allowing testers to focus on exploratory testing, enhancing their ability to find defects. Although initially requiring an investment in training and tool setup, test automation ultimately reduces costs, particularly for regression tests, as automated tests are much faster than manual ones. This results in improved software quality and lower maintenance costs.



Abbildung 2: (Quelle: Bild von Tumisu auf Pixabay)

## How to choose the right test automation tool?

Before diving into the opportunities and challenges that meet any organization shifting to test automation, let's discuss first how to choose the right test automation tool.

There are many open-source software test automation frameworks, such as Selenium, Appium, Watir, Mocha, Cucumber and Galen Framework.

New testing tools are being developed all the time, and it can get complicated to choose a suitable tool, especially with the new tools emerging every day. There is no one-size tool that fits all. There are important questions that need to be asked before using or purchasing a new tool.

- What will the tool be used for?
- Does it solve a current problem you have?
- Will it improve your work?
- Is your team already trained to use this tool or training will be needed?
- What is the ROI for using this tool?

Every team is different and you need to understand your team's needs and skills before utilizing any tool. Take your time and implement a pilot project before purchasing any tool and also don't forget that you can take advantage of open-source tools.

## Challenges and opportunities

### Challenges

The selection process for a test automation tool can be a difficult task for organizations. Selecting the suitable tool depends on project requirements, scalability needs, team's skills, and existing projects and integration of the test automation tool with other tools. Maintaining the test automation framework with diverse platforms and configurations is another challenge that requires a lot of attention to details. Furthermore, training and adapting teams to new automation technologies are necessary. There is also the burden of script maintenance, with the continuous updates and new features added to the software all the time. New challenges might be added constantly, especially if organizations are changing to Agile and DevOps practices.

### Opportunities

Even though there are many challenges, there are even more opportunities for organizations embracing test automation modernization. New automation frameworks and libraries provide better support to specific testing needs, offering solutions for CI/CD pipeline integration and parallel test executions. Today artificial intelligence and machine learning technologies, cloud-based testing platforms and shift-left testing practices are creating a new era for test automation, with AI-driven testing and test case generation, finding bugs earlier and helping with predicting them, improving communication and increasing test coverage. All those opportunities are leading to faster feedback and improved overall software quality. With good understanding of the current market and tools, organizations can overcome challenges and unlock the full potential of test



automation.

### Emerging trends and innovations

Some trends and innovations were already mentioned in the previous paragraph but I would like to dive deeper into these aspects and explain in more detail those new technologies and strategies.

There are many emerging trends that are aiming at changing and reshaping the testing practices for the better. Firstly, artificial intelligence is of course one of the leading trends at the moment along with machine learning technologies. Integrating those technologies into test automation frameworks offers better and faster test case generation which leads to faster testing process, prediction of issues which helps in saving costs, and more advanced and detailed reports. Secondly, shift-left testing practices are becoming more popular. Shift-left testing is an approach to software testing in which testing is performed earlier in the software development lifecycle, thereby offering faster feedback loops and improved software quality. Thirdly, microservices testing strategy is a method that involves breaking down an application into smaller, independent services and testing each one separately. This approach focuses on testing the APIs and communication between these microservices to ensure they integrate properly. Fourthly, DevOps practices promise continuous development, testing, integration, delivery, deployment and monitoring which leads to better and faster product delivery and also a more stable operating environment.

By following these trends, staying updated, and adapting to change, organizations can open new opportunities for effectiveness, agility and quality assurance in their test automation techniques.

### Test Automation: Past, Present and Future

Before taking a peek into the future of test automation, it is important to understand its past and how much test automation has already developed. Software testing began as a completely manual process, performed entirely by people without the use of external tools. This approach, known as manual testing, is still in use today. While manual testing was sufficient for simpler software, the increasing complexity of software required more advanced testing methods. This led to the introduction of using computers to automate tests. Software testers began writing test scripts, providing instructions for computers to execute. This automation proved to be effective in finding bugs and saving time. Since then test automation started to really take shape. Record and playback tools started to emerge, then as software became more complex, software testers started automating functional and regression tests to save time. The process needed to be faster so

separating test data from test scripts became necessary and data-driven and keyword-driven testing appeared, making the process of test creation much faster. As test automation needed to be more scalable and easier to manage, the need for test automation frameworks to help organize and structure tests in a better way came up. As the software development cycle has accelerated with the adoption of agile methodologies and DevOps, test automation has also had to evolve to keep pace with development. Adapting to the current landscape, which includes CI/CD, agile practices, and complex applications, necessitates both shift-left and shift-right testing. This approach integrates testing earlier in the development process and extends it into the post-deployment phase to ensure software behaves correctly in real-world scenarios. Given this rapid progress, artificial intelligence and machine learning are poised to play a leading role in the future, ensuring that software testing continues to match the speed and complexity of development.



IT Trends und Themen  
aus den Bereichen  
Softwaretest, Development, PM und  
Business Analyse auf  
unserem **Videoblog!**



## Conclusion

In the ever-evolving landscape of software testing, the journey towards test automation modernization is not a destination but a continuous evolution and determination of organizations striving to stay ahead. As challenges, opportunities and innovations of test automation were discussed in the article, we can clearly see that the future of test automation is packed with potential. It is a future where artificial intelligence improves the software test process, cloud-based testing platforms enhance communication and shift-left testing becomes the norm and not the exception.

As we close this chapter, remember that test automation is here to stay and that the future holds more surprises and that this is just the beginning of a long, ever-evolving journey of modernization.

Weitere spannende  
Blogartikel finden Sie  
hier:

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



## Quellen und weiterführende Informationen:

<https://www.getxray.app/blog/the-complete-guide-to-test-automation-best-strategies-frameworks-and-tools>

<https://www.perfecto.io/blog/what-is-test-automation#:~:text=Test%20automation%20is%20the%20process,testing%20or%20automated%20QA%20testing.>

<https://www.wetest.net/blog/benefits-and-challenges-of-test-automation-775.html>

<https://www.atlassian.com/devops>

<https://www.browserstack.com/guide/what-is-cloud-testing#:~:text=Cloud%20Testing%20refers%20to%20the,accessible%20online%20at%20all%20times.>

<https://smartbear.com/learn/automated-testing/shifting-left-in-testing/>

<https://www.keysight.com/blogs/en/tech/software-testing/2022/4/6/5-truths-about-ai-driven-software-testing>

<https://www.redhat.com/en/topics/devops/what-cicd-pipeline>

<https://www.xenonstack.com/blog/microservices-testing#:~:text=Microservices%20Testing%20Strategy%20is%20a,to%20ensure%20they%20integrate%20properly.>

<https://www.guvi.in/blog/the-future-scope-of-automation-testing/>



**Maged Elsherif** ist Consultant bei SEQIS.

Mit einem Hintergrund im Maschinenbau wechselte er in das Gebiet der Softwaretests und begann als Tester für ein Video-spieleunternehmen. 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.

# 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

- 26. September 2024: Nachhaltigkeit in der Softwareentwicklung: Wurst oder Wahnsinn?
- 14. November 2024: Projekt & Change Management – mit Support durch AI
- 3. April. 2025: Thema TBD
- 22. Mai 2025: Thema TBD
- 25. September 2025: Thema TBD
- 13. November 2025: Thema TBD

Informiert bleiben



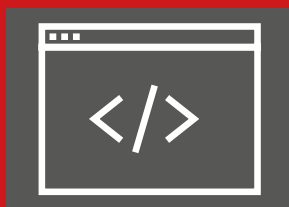
[www.seqis.com/events-index](http://www.seqis.com/events-index)

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



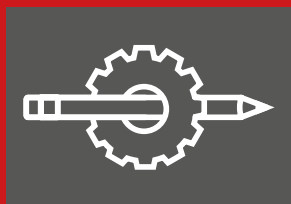
### TESTING

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