

GWGD NACHRICHTEN 05|26

Reproducible Research

.....
Data Protection Officer
and AI

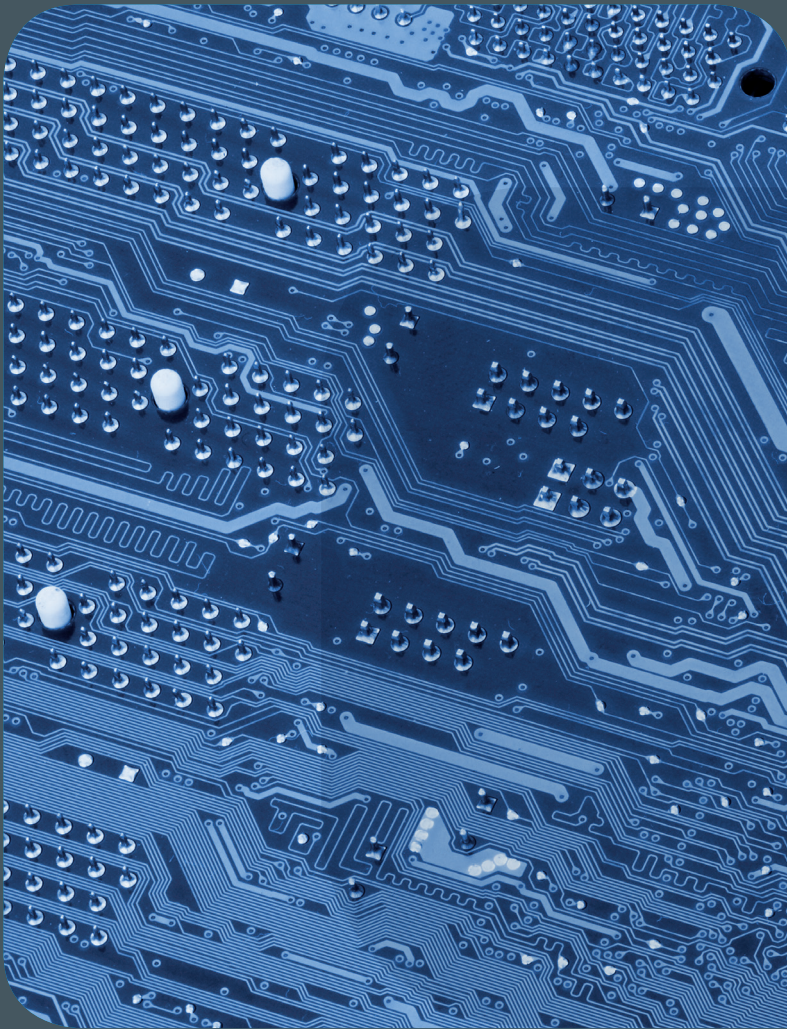
.....
Anwendungsentwicklung
mit KI-Agenten

.....
Open-Source Operating
Systems

.....
AVefi

ZEITSCHRIFT FÜR DIE KUND*INNEN DER GWGD





GWDG **NACHRICHTEN**

05|26 Inhalt

-
- 4 The 80/20 of Reproducible Research – Ten Years after the Crisis** **9 The Enhanced Role of the Data Protection Officer in the Age of AI** **14 Spezifikationsgesteuerte Entwicklung von Anwendungen mit KI-Agenten** **24 Open-Source Operating Systems: More than just Linux?** **28 Die GWDG bei Science goes City** **30 Gemeinsamer Zukunftstag bei der GWDG und beim IT-Service-Center der UMG** **32 Filmrechercheplattform geht online – Projekt „AVefi“ vernetzt Filmbestände für Archive, Museen und Forschung** **34 Personalia** **35 Academy** **39 Kurz & knapp**

Impressum

.....

Zeitschrift für die Kund*innen der GWDG

ISSN 0940-4686
49. Jahrgang
Ausgabe 5/2026

Erscheinungsweise:
10 Ausgaben pro Jahr

www.gwdg.de/gwdg-nr

Auflage:
500

Fotos:
© tadamichi - stock.adobe.com (1)
© WrightStudio - stock.adobe.com (8)
© AndSus - stock.adobe.com (23)
© Nataliya Kalabina - Fotolia.com (27)
© Shuo - stock.adobe.com (31)
© Robert Kneschke - Fotolia.com (35)
© GWDG (2, 3, 31,34)

Herausgeber:
Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen
Burckhardtweg 4
37077 Göttingen
Tel.: 0551 39-30001
Fax: 0551 39-130-30001

Redaktion und Herstellung:
Dr. Thomas Otto
E-Mail: thomas.otto@gwdg.de

Druck:
Kreationszeit GmbH, Rosdorf



Prof. Dr. Ramin Yahyapour
ramin.yahyapour@gwdg.de
0551 39-30130

*Liebe Kund*innen und Freund*innen der GWDG,*

*dass die Anwendung von Künstlicher Intelligenz eine disruptive Entwicklung ist, deren Auswirkungen wir aktuell nur erahnen, geschweige denn vorhersagen können, lesen wir allenthalben. Die Entwicklung ist hochgradig dynamisch. Was vor kurzem noch unlösbar war, ist nun schon praktisch einsetzbar. Wer einmal moderne KI-Agenten für Softwareentwicklung eingesetzt hat, stellt schnell fest, dass sich das Berufsbild von Programmierer*innen stark wandelt. Forschende aller Disziplinen profitieren von ad-hoc erstellten Codes für bspw. die Datenauswertung ihrer Experimente. Die Lehrinhalte an den Hochschulen müssen an diese Veränderungen angepasst werden.*

Als KI-Zentrum stellt dies auch die GWDG vor Herausforderungen. Wir stellen vielfältige KI-Dienste zur Verfügung, darunter auch externe Sprachmodelle wie ChatGPT. Auf vielfachen Wunsch haben wir im April auch Claude ins Programm genommen. Offenbar haben wir damit einen Nerv getroffen, da uns durch die Nutzung in kürzester Zeit die Kostenexplosion im Vergleich zu den Vormonaten dazu gezwungen hat, das Gratisangebot von externen Modellen einzuschränken und nur noch bei Kostenübernahme bereitstellen zu können. Auf der einen Seite bedauerlich, aber auf der anderen Seite ein Zeichen, wie dynamisch die Entwicklung ist. Die GWDG arbeitet aktuell gemeinsam mit ihren Gesellschaftern und externen Einrichtungen Konzepte für die nachhaltige Bereitstellung und Nutzung von KI-Diensten aus. Im April angebotene Workshops für den Übergang zu kostenpflichtigen Modellen des Chat AI-Ökosystems wurden von 400 Teilnehmenden besucht.

Ramin Yahyapour

GWDG – IT in der Wissenschaft

The 80/20 of Reproducible Research – Ten Years after the Crisis

Text and Contact:

Georgios Kaklamanos
georgios.kaklamanos@gwdg.de

In the 2010s, a series of studies across psychology, cancer biology, computer science, and other fields brought a long-standing problem into sharp focus: a significant fraction of published research findings could not be reproduced, even by their original authors. A decade on, the tools and infrastructure for reproducible research have improved considerably – but the problem persists. The barrier is not technical; it is a gap between available practices and their adoption. This article identifies six practices that represent the highest-return investment a computational researcher can make: scripting every analytical step, using version control, organising projects consistently, recording the computational environment, documenting code and data, and archiving outputs with an appropriate licence. Along with the practices, we present the relevant services offered by the GWDG that can assist in implementing them. Taken together, they address the most common causes of irreproducibility while remaining accessible to any researcher regardless of their computational background.

INTRODUCTION

In 2016, a landmark article was published in Nature. It was a survey of more than 1,500 scientists across disciplines. The results showed that over 70% had tried and failed to reproduce another researcher's work, and more than half had failed to reproduce their own (Baker, 2016). A large-scale effort to replicate 100 published psychology studies found that only thirty-six percent of replications had statistically significant results (Open Science Collaboration, 2015). In computer science, attempts to obtain and run code from published papers succeeded less than half the time (Collberg & Proebsting, 2016).

These findings did not emerge from nowhere. As early as 2005, John Ioannidis had argued on theoretical grounds that most published research findings are likely false, given the statistical realities of underpowered studies, researchers' degrees of freedom, and publication bias (Ioannidis, 2005). What the 2010s added was the empirical evidence – and the scale of the problem proved difficult to dismiss.

THE IMPORTANCE OF REPRODUCIBLE RESEARCH

The consequences of irreproducible research are not merely academic. In 2010, the economists Carmen Reinhart and Kenneth Rogoff published a study arguing that countries with public debt exceeding 90% of GDP experienced negative economic growth (Reinhart & Rogoff 2010). Their finding was widely cited in policy

Die 80/20-Regel der reproduzierbaren Forschung – zehn Jahre nach der Krise

In den 2010er Jahren rückte eine Reihe von Studien aus den Bereichen Psychologie, Krebsbiologie, Informatik und anderen Fachgebieten ein seit langem bestehendes Problem in den Fokus: Ein erheblicher Teil der veröffentlichten Forschungsergebnisse ließ sich nicht reproduzieren, nicht einmal von den ursprünglichen Autor*innen. Ein Jahrzehnt später haben sich die Werkzeuge und die Infrastruktur für reproduzierbare Forschung erheblich verbessert – doch das Problem besteht weiterhin. Die Barriere ist nicht technischer Natur; es handelt sich um eine Kluft zwischen verfügbaren Praktiken und deren Umsetzung. Dieser Artikel identifiziert sechs Praktiken, die für eine*n Computerwissenschaftler*in die Investition mit dem höchsten Ertrag darstellen: das Skripten jedes analytischen Schritts, die Verwendung von Versionskontrolle, die konsistente Organisation von Projekten, die Aufzeichnung der Rechenumgebung, die Dokumentation von Code und Daten sowie die Archivierung von Ergebnissen unter einer geeigneten Lizenz. Neben diesen Vorgehensweisen stellen wir die entsprechenden Dienste der GWDG vor, die bei deren Umsetzung helfen können. Zusammengekommen gehen sie die häufigsten Ursachen für Nichtreproduzierbarkeit an und bleiben dabei für jede*n Forscher*in zugänglich, unabhängig von seinem/ihrer computergestützten Hintergrund.

debates across Europe and the United States to justify austerity measures. Three years later, a graduate student attempting to replicate the analysis discovered that five countries had been accidentally excluded from a key calculation due to an error in a Microsoft Excel cell selection range. Correcting the error substantially changed the paper’s conclusions (Herndon, Ash, & Pollin, 2014).

The data had been available, and the methods had been described – but the actual computational steps had never been independently verified, and an error with real policy consequences went undetected for years. This example illustrates the most direct cost of irreproducibility: findings that are wrong, or less robust than reported, propagate into the literature, into policy, and into the work of other researchers who build upon them. On the other hand, the benefits also have effects at multiple levels.

For **individual researchers**, scripted and version-controlled analyses mean that returning to a project after months away – as invariably happens between submission and revision – requires reading code rather than reconstructing lost mental context. When a reviewer requests a change to an analysis, a reproducible pipeline can regenerate all affected results with a single command rather than a week of manual rework. Studies with openly available data and code also receive significantly more citations than those without (Piwowar & Vision, 2013), and funding agencies increasingly require data management plans that include provisions for computational reproducibility.

For **research groups and institutions**, the benefits compound. Laboratories experience constant turnover: students graduate, postdoctoral researchers move on, and each departure risks taking with it the tacit knowledge required to maintain or extend existing analyses. Reproducible practices serve as institutional memory, protecting the group’s intellectual output against personnel changes. At a broader level, institutions that invest in reproducibility infrastructure – training programmes, research software engineering support, data management services – are better placed to meet the growing requirements of funders such as Horizon Europe, which requires open access to data and encourages open-source software.

This article identifies a small set of practices – the 20% of effort – that address the largest share of reproducibility failures in computational research. It is written for researchers who want to make their work more robust and transparent without completely changing their workflow.

WHAT DO WE MEAN BY REPRODUCIBILITY

The National Academies of Sciences, Engineering, and Medicine (2019) differentiates between reproduction and replication. Reproduction refers to the process of obtaining the same results, using the same data, and the same processes as the original research. On the other hand, replication is the process of collecting new data and using (possibly new) procedures to test the same research question (see Figures 1 and 2). Reproduction evaluates the validity of the scientific analysis, while replication evaluates the validity of the scientific claim.

This article focuses on computational reproducibility: the ability for someone (including your future self) to use your data and code to reproduce your published results. This is, in principle, the easiest form of scientific verification. If a result cannot even be reproduced, it certainly cannot be evaluated, extended, or corrected.

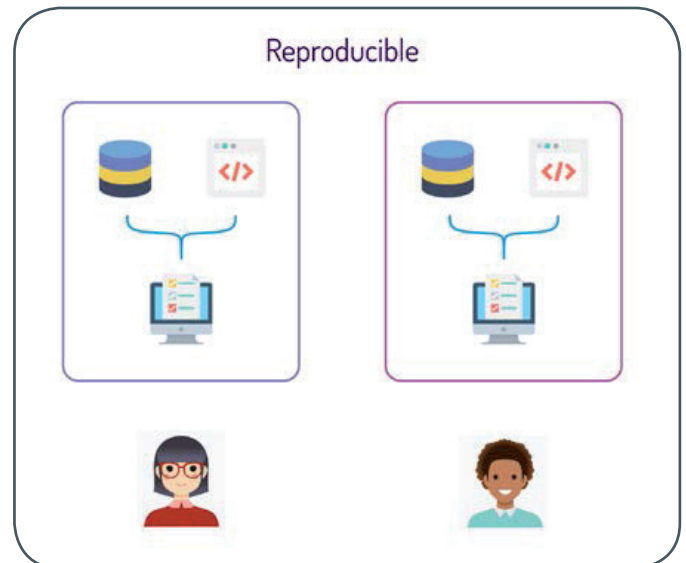


Figure 1: Reproducible Analysis: same data and processes, same results, Allard (2018)

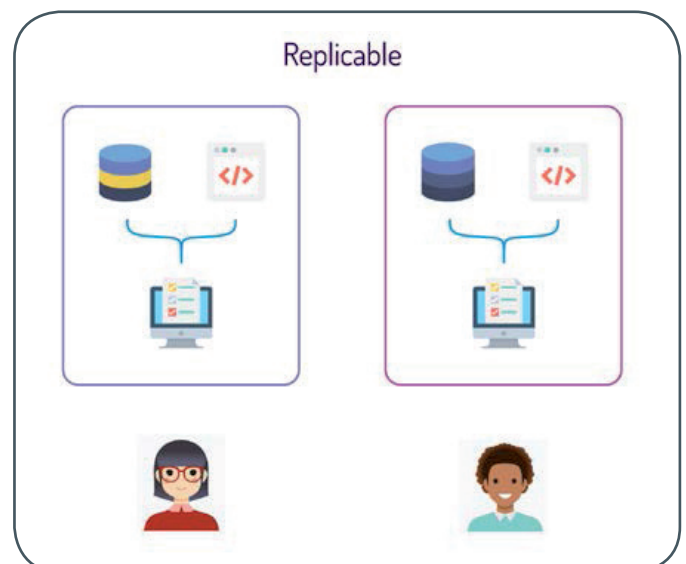


Figure 2: Replicable Analysis: different data and processes, same results, Allard (2018)

PRACTICE 1: DON'T MAKE CHANGES BY HAND, SCRIPT EVERYTHING

The single highest-impact change a researcher can make is to eliminate manual steps from their analytical pipeline. Every operation between raw data and final result – data cleaning, transformation, statistical analysis, figure generation – should be expressed as executable code rather than performed interactively. Simply put, clicking with the mouse shouldn’t be part of the analysis/cleaning/etc process.

This does not require elegant code. A messy script that runs correctly is infinitely more reproducible than a polished procedure that was performed by hand in a spreadsheet and is only stored in the researcher’s brain.

The practical test is simple: starting from your raw data, can you regenerate every number, figure, and table in your paper by executing a sequence of scripts, without any manual intervention? If the answer is no, there is a manual step that is both undocumented and unreproducible. Sandve et al. (2013) elevate this to their

first rule of reproducible computational research: for every result, keep track of how it was produced – and the most reliable way to keep track is to express the production as code.

PRACTICE 2: USE VERSION CONTROL

Version control – specifically Git – provides three capabilities that are essential for reproducible research: an audit trail, safe experimentation, and collaboration infrastructure:

- **Audit Trail**, also known as commit history: Every change to every file is recorded with a timestamp, an author, and (if you write good commit messages) a reason. When something breaks, you can identify exactly when and why. When a reviewer asks what changed between draft versions, the answer is in the commit history.
- **Branching**: Enables multiple versions of your codebase that can exist simultaneously. This allows experimentation and an alternative analysis on a branch without risking the main pipeline. Afterwards, you can discard or merge the result as appropriate. This encourages exploration without the anxiety of breaking something that currently works.
- **Remote hosting**: Services such as GitHub and GitLab provide backup, collaboration, and a point of access for others. It also creates a public record of the project’s development, which can serve as evidence of research integrity and timeline. The GWDG is hosting a GitLab instance available to researchers [1].

For researchers new to Git, the learning curve is real but modest. Graphical clients and IDE integrations (RStudio, VS Code) considerably reduce the friction.

It is important to note here that Git is designed for plain text files (e.g., code, small CSV files, etc). Large data sizes and binary

files should not be committed directly to a Git repository. For these use cases, specialized tools such as DataLad [2] or DVC (Data Version Control) [3] are recommended. They integrate well with Git and are designed specifically to handle data.

PRACTICE 3: ORGANIZE CONSISTENTLY

A reproducible project needs a clear, consistent directory structure that separates raw data, processed data, code, results, and documentation. A widely recommended layout suggested in Wilson et al., (2017) is shown in Figure 3.

Two principles matter most:

1. **Raw data is read-only**: the *data/raw/* directory contains the original data exactly as received, and nothing in the pipeline modifies it. All transformations produce new files elsewhere. This ensures that the starting point of any analysis is always known, unambiguous, and recoverable.
2. **Generated outputs are disposable**: everything in *results/* can be deleted and regenerated from the code and data. If it cannot, something is missing – there is a hidden manual step or an unrecorded dependency.

File naming conventions deserve a brief mention. Jenny Bryan’s (Bryan, 2015) widely cited guideline is that file names should be:

1. Machine-readable
2. Human-readable
3. Play well with default ordering

These can be achieved by some simple heuristics such as: using dates in ISO format ('YYYY-MM-DD'), avoiding spaces and special characters, making names descriptive enough that the file’s purpose is clear without opening it. Some example file names could be:

```

.
├── AUTHORS.md      <- File containing the contact information of authors
├── bin             <- Compiled models or programs used for the analysis
├── config          <- Configuration files for models, programs, etc
├── data            <- Directory storing your data
│   ├── external   <- Data from 3rd party sources
│   ├── interim    <- Intermediate transformed data
│   ├── processed  <- The final data for modeling and plots
│   └── raw         <- The original data dump
├── docs            <- Documentation, papers, etc.
├── LICENSE         <- The licence of your project
├── notebooks      <- Directory with Jupyter or R Notebooks
├── README.md      <- File providing general information for the project
├── reports        <- Manuscript files (e.g. LaTeX, Markdown, etc) for reports
│   └── figures    <- Figures and plots for paper and reports
└── src            <- Directory containing code
    ├── data       <- Scripts that process data
    ├── external   <- External code and tools
    ├── models     <- Code for your own models
    ├── tools      <- Helper scripts
    └── visualization <- Scripts for making figures
  
```

Figure 3: A recommended project directory structure adapted from Cookiecutter Data Science [4]

1. *2026-01-20_Generation-of-requests-per-region-histogram.py*
2. *Requests-per-day-by-IP_PostgreSQL_part01.csv*
3. *Requests-per-day-by-IP_NGINX_part01.csv*

Here we use dashes ("-") to separate the different parts inside a section (i.e. Year-Month-Day), underscores ("_") to separate the different sections of the file names (i.e. date, purpose of script, etc), and by using zero padding on the parts section, we ensure that there won't be any ordering errors (e.g. instances where part10 appears before part2).

PRACTICE 4: RECORD YOUR COMPUTATIONAL ENVIRONMENT

Code that runs on your machine today may fail on your collaborator's machine tomorrow, or on your own machine next year, because a dependency has been updated. The computational environment – language version, library versions, operating system – is as much a part of the analysis as the code itself, and it must be recorded.

At a minimum, provide a specification file that lists all dependencies with version numbers: *requirements.txt* or *environment.yml* for Python, *renv.lock* for R. This allows others (and your future self) to recreate the environment. The effort is negligible – most tools can generate these files automatically – but the payoff is significant. Trisovic et al. (2022) found that a large fraction of archived R code failed to run due to dependency issues, a problem that environment specification files directly prevent.

For stronger guarantees, containerization tools such as Docker and Apptainer can package the entire computational environment – operating system, libraries, tools, and code – into a portable image. They run identically on any compatible system and are particularly valuable for complex pipelines with many dependencies. Users of the GWDC HPC systems can use Apptainer containers for their analysis [5].

PRACTICE 5: WRITE A README AND DOCUMENT YOUR CODE AND DATA

Every project needs a README file that answers the questions a newcomer would ask: What does this project do? What data and software does it require? How do I run the analysis? What does each directory contain? Who should I contact with questions? A few hundred words suffice. The README is the front door of the project; without it, even well-organized code is inaccessible.

A common failure mode is to write the README at the start of a project and never update it, or to write it hastily at the end when the details have faded. The best approach is to treat the README as a living document, updated as the project evolves – much like a lab notebook.

Data documentation is equally critical. Every dataset should be accompanied by a description of its variables – names, types, units, codes for missing values – and provenance information: where the data came from, when it was accessed, and what transformations were applied. This can be a simple text or CSV file alongside the data. Without it, the data is an opaque artifact that invites misinterpretation. For projects using community data, adherence to discipline-specific metadata standards (Dublin Core, Data-Cite, BIDS, EML, and others) ensures that documentation is both

complete and interoperable with existing tools and repositories.

PRACTICE 6: ARCHIVE AND LICENSE YOUR WORK

When a paper is published, the supporting code and data should be deposited in a repository that provides a persistent identifier (PID) and long-term preservation. GitLab is excellent for development, but it is not an archive: repositories can be deleted, organizations can change, and the platform offers no formal preservation guarantees. Archiving is the responsibility of data repositories. Zenodo [6] (backed by CERN) integrates directly with GitHub and can automatically archive each tagged release with a Digital Object Identifier (DOI). Similarly, the GWDC and the Göttingen eResearch Alliance provide GRO.data [7], a general-purpose research data repository based on the Dataverse platform. The GWDC also provides a PID service [8].

Crucially, all published and archived materials need a license. Without one, copyright law in most jurisdictions defaults to "all rights reserved", meaning no one can legally reuse the work regardless of the researcher's intent. For code, a permissive open-source license (MIT, Apache 2.0) is simplest. For data, Creative Commons licenses (CC BY or CC0) are standard. The choice matters less than the act of choosing: any open license is better than none. Morin et al (2012) provide a quick guide on licensing for researchers, and tools like "Choose a licence" [9] can simplify the selection process.

These practices align with the FAIR principles – Findable, Accessible, Interoperable, Reusable (Wilkinson et al., 2016) – which have been adopted by major funders as the framework for research data management. Compliance with FAIR is increasingly not optional; it is a requirement for funded research.

FINAL THOUGHTS

It is now 2026, a decade after the Nature article was published and the reproducibility crisis has entered public consciousness. There has been progress. The tools available for reproducible research are better and easier to use than ever before. Journals have tightened their policies regarding data and code availability. Yet surveys continue to find that a large fraction of computational research cannot be re-executed even when code is nominally available (Trisovic et al., 2022). Similarly, Hornung et al (2026) note that "the proportion of research articles aiming for reproducibility remains low and uneven across disciplines". The gap is not technological. It is a gap in practice.

The six practices described here are not novel, not technically demanding, and not time-consuming once they become a habit. Yet they can have a significant impact on closing that gap. For further information in this topic, the references of this article can be a good starting point, specifically Hornung et al (2026), Wilson et al (2017), and Sandve et al (2013).

The Royal Society's founding motto, Nullius in verba – take nobody's word for it – was a call for evidence over authority. In computational research, honoring that call means ensuring that behind every publication, there is a script that can be run, an environment that can be recreated, data that can be found, and documentation that makes the whole thing intelligible. That is not a high bar. It should be the bare minimum.

LINKS

- [1] <https://gwdg.de/en/services/e-mail-collaboration/gitlab>
- [2] <https://www.datalad.org>
- [3] <https://dvc.org>
- [4] <https://cookiecutter-data-science.drivendata.org/#directory-structure>
- [5] https://docs.hpc.gwdg.de/software_stacks/list_of_modules/apptainer/index.html
- [6] <https://zenodo.org>
- [7] <https://www.eresearch.uni-goettingen.de/services-and-software/goettingen-research-online>
- [8] <https://gwdg.de/en/services/research-data-management/persistent-identifier-pid>
- [9] <https://choosealicense.com>

REFERENCES

- Allard, Tania (2018). Down the rabbit hole. A 101 on reproducible workflows with Python. figshare. Presentation. <https://doi.org/10.6084/m9.figshare.6222554.v2>
- Baker, M. (2016) 1,500 scientists lift the lid on reproducibility. *Nature* 533, 452–454 (2016). <https://doi.org/10.1038/533452a>
- Bryan, J. (2015). Naming things. Slides from a talk at the Reproducible Science Workshop, Duke University.
- Collberg, Christian, and Todd A. Proebsting. "Repeatability in computer systems research". *Communications of the ACM* 59.3 (2016): 62-69.
- Herndon, Thomas, Michael Ash, and Robert Pollin. "Does high public debt consistently stifle economic growth? A critique of Reinhart and Rogoff". *Cambridge journal of economics* 38.2 (2014): 257-279.
- Hornung, Roman, et al. "Overcoming Barriers to Computational Reproducibility". arXiv preprint arXiv:2602.03863 (2026).
- Ioannidis, John PA. "Why most published research findings are false". *PLoS medicine* 2.8 (2005): e124.
- Morin, Andrew, Jennifer Urban, and Piotr Sliz. "A quick guide to software licensing for the scientist-programmer". (2012): e1002598.
- National Academies of Sciences, et al. *Reproducibility and replicability in science*. National Academies Press, 2019.
- Open Science Collaboration. "Estimating the reproducibility of psychological science." *Science* 349.6251 (2015): aac4716.
- Piwowar, Heather A., and Todd J. Vision. "Data reuse and the open data citation advantage". *PeerJ* 1 (2013): e175.
- Reinhart, Carmen M., and Kenneth S. Rogoff. "Growth in a Time of Debt". *American Economic Review* 100.2 (2010): 573-578.
- Sandve, Geir Kjetil, et al. "Ten simple rules for reproducible computational research". *PLoS computational biology* 9.10 (2013): e1003285.
- Trisovic, Ana, et al. "A large-scale study on research code quality and execution". *Scientific Data* 9.1 (2022): 60.
- Wilkinson, Mark D., et al. "The FAIR Guiding Principles for scientific data management and stewardship". *Scientific data* 3.1 (2016): 1-9.
- Wilson, Greg, et al. "Good enough practices in scientific computing". *PLoS computational biology* 13.6 (2017): e1005510. ●



The Enhanced Role of the Data Protection Officer in the Age of AI

Text and Contact:

Meisa Kamyab
meisa.kamyab@gwdg.de

Data breaches rarely happen in the spotlight. They are often quiet, barely noticeable, and their effects only become apparent after some time. But when they do come to light, the consequences are all the more serious: legally, financially and in terms of reputation. In an increasingly data-driven economy, where artificial intelligence is playing an ever more central role, this tension is further exacerbated. Companies face the challenge of reconciling innovation with regulatory requirements. This article highlights the importance of compliance with data protection regulations in the context of modern AI systems and examines, in particular, the role of the Data Protection Officer (DPO) in relation to the European AI Act. It demonstrates why effective compliance structures are more crucial today than ever before.

INTRODUCTION

There is a famous parable in Masnavi-e-Ma'navi by the Persian mystic, Rumi, about a burglar who is digging into the wall of a house to break into it. A man witnessing this trespass asks him what he is doing and faces the answer: "I am playing a drum". The man, astonished by the response from the alleged drummer, inquires where the sound of the drum is. The thief answers: "You will hear it tomorrow", needless to say, in the form of the household's outcry [1]. A parable that is certainly relevant to the issues of data protection and AI in today's world.

Many of the data breaches that occurred after the enactment of GDPR (General Data Protection Regulation) resembled the sounds of a drum, unnoticed during the night. While infringing upon data subjects' fundamental rights and freedoms in the absence of efficient monitoring, they happened in silence but still had far-reaching consequences later on. Examples of data breaches are not rare, some of which have been advertised as scandals. In 2018, it was revealed that Cambridge Analytica, a British political consulting firm, had harvested the personal data of over 87 million Facebook users without their consent (violating Facebook's policies and users' privacy). This data was allegedly used to influence voter behaviour during major political events, including the 2016 U.S. Presidential Election and the Brexit referendum.

However, there is no guarantee that the sounds of the drum will be gracefully embraced by the enforcement authorities. Facebook faced intense public backlash, congressional hearings, and a \$ 5 billion fine from the U.S. Federal Trade Commission (FTC). When France's data protection authority noticed that Google made cookies refusal harder than acceptance, the result was a € 150 million fine. TikTok was fined € 345 million for not providing adequate

protection for children's privacy and the breach of transparency obligations, and suffered another € 530 million in May 2025 from Irish regulators for unauthorized data transfer to China.

One might argue that in a data-driven economy, taking the risk of potential penalties is a business strategy for IT companies. Big Tech companies are frequently criticized for their digital sovereignty, and performing the functions that was exclusive to states by determining the terms and conditions in cyberspace [2]. Nevertheless, the price is the reputation of companies and their trust. This article tries to address the enhanced role of the data protection officers and their responsibilities which are intensified with the European AI Act coming into force. In doing so, it positions the AI Act among other regulations. But before that, it clarifies the importance of complying with GDPR.

Die wachsende Rolle des Datenschutzbeauftragten im KI-Zeitalter

Datenpannen bleiben zunächst oft unbemerkt, doch wenn sie ans Licht kommen, sind die Folgen schwerwiegend – rechtlich, finanziell und rufschädigend. In einer zunehmend datengesteuerten Wirtschaft, die von Künstlicher Intelligenz geprägt ist, verschärft sich dieser Konflikt, da Unternehmen bestrebt sind, Innovation und Einhaltung gesetzlicher Vorschriften in Einklang zu bringen. Dieser Artikel untersucht die Bedeutung der Einhaltung von Datenschutzvorschriften in modernen KI-Systemen, wobei der Schwerpunkt insbesondere auf der Rolle des Datenschutzbeauftragten im Kontext des Europäischen KI-Gesetzes liegt, und zeigt auf, warum robuste Compliance-Strukturen wichtiger denn je sind.

THE AI ACT AND OTHER REGULATIONS

With the exponential outburst of AI technologies and the insatiable desire of industry to incorporate AI models into systems, the AI Act has found an important place in the compliance debate. However, as AI models are trained on a large amount of data, compliance with data protection rules remains vital. The AI Act applies without prejudice to other already in force regulations [3].

In a landmark decision against the German credit scoring agency Schufa AG Holding (SCHUFA) [4], the European Court of Justice (ECJ) held that creating a credit score (the automated establishment of a probability value based on personal data and concerning his or her ability to meet payment commitments in the future) constitutes an automated decision about individuals according to Article 22 of GDPR [5]. In these cases, the data controller shall implement suitable measures to safeguard the data subject's rights and freedoms and legitimate interests, at least the right to obtain human intervention on the part of the controller, to express his or her point of view, and to contest the decision (Article 22 (3) GDPR). Many of the AI systems that use automated decision-making need to take GDPR rules into account. This is the reason the idea of assigning the duties of AI office to the DPO originated.

The AI Act does not define the role of an AI Officer, nor does it specifically mandate the appointment of an AI Officer. In contrast, the designation and responsibilities of the DPO are defined in articles 37-39 of the GDPR. If the DPO assumes the role of the AI officer, it is possible that there will be a conflict of interest and a possibility to undermine the independence of a DPO. Article 38 of GDPR prevents a conflict of interest by ensuring that the person overseeing compliance with data protection rules is not the same person executing the data processing operations. While overseeing AI regulatory compliance may not be the DPO's primary responsibility, their knowledge is essential in areas where AI and GDPR intersect. Among them are data protection, identifying bias in high-risk AI systems, and evaluating when human oversight is legally required to safeguard individual rights [6].

This is why the data protection officer is best placed to cooperate with the AI office.

THE STEPS IN LEGAL COMPLIANCE

Finding the Regulations, Compliant with what?

Many regulations are of interest in a data center. Network and Information Security Directive (NIS2), Data Act, EU Cybersecurity Act, ISO/IEC 27001, European Standard for Data Center Facilities, AI Act, Digital Service Act, or Data Act are some of them. When it comes to rights and freedoms of individuals and a responsible Data Center, regulations like GDPR and AI Act come to the fore.

Knowing regulations alone does not guarantee compliance. Some of the rules are specified in court rulings. They provide a source of interpretation when the legal rules lack clarity. For instance, according to the Court of Justice of the European Union, consent to cookies is not validly constituted if it is given through "a pre-checked checkbox which the user must deselect to refuse consent" [7]. Establishing "the right to be forgotten" was the result of refusing a businessman's request by Google to remove links to old newspaper notices about a property auction related to his debt. His claim was accepted, broadening the scope of the right to erasure (Article 17 GDPR). Search engines are obligated

to remove links to inadequate, irrelevant, or excessive personal data upon request, even if the original publication is lawful and remains online [8]. In balancing freedom of the press and respect for privacy, a higher balance will be given to freedom of expression if the personal data relates to a public figure [9]. For understanding these interpretation, constant tracking of the court decisions is necessary.

Understanding the Risks

Legal norms differ from those of ethics in benefiting from a compulsory authority for enforcement. Lawyers are famous for using loopholes in the law to minimize sanctions or evade them by offering a different interpretation. It is neither unethical nor unprofessional, but the art of a legal expert is to advise their clients toward what is best for them. In this sense, law is not far from compliance.

Legal risks do not always carry the likelihood of severe repercussions. In the context of general-purpose AI models, if the model was placed on the market before 2 August 2025, the developers would have more time to adapt to the requirements of the AI Act [10]. This difference is due to challenges they might face in compliance and avoiding retrospective application of law. "In particular, providers of general-purpose AI models placed on the market before 2 August 2025 are not required to conduct retraining or unlearning of models, where it is not possible to do this for actions performed in the past, where some of the information about the training data is not available, or where its retrieval would cause the provider disproportionate burden. Such instances must be clearly disclosed and justified in the copyright policy and in the summary of the content used for training" [11]. Delay in placing on the market can be interpreted as a risk, as it will need more organisational resources to fully comply with the AI-related obligations. Therefore, strategies gain significant importance.

Deciding on the Strategies

Intents are important and decisive in law, as no responsibility can be assigned to a human without an intent. Autonomy is what makes humans an "end" in themselves in the language of Immanuel Kant. In the context of criminal law, playing with mens rea (the guilty intention) can turn a murder into manslaughter with a less severe sentence. Having the importance of intent in mind, the fines for infringing data protection rules can vary depending on the level of cooperation with data protection authorities. "Any action taken by the controller or processor to mitigate the damage suffered by the data subjects", how "the infringement became known to the supervisory authority", "any relevant previous infringement", "any action taken to mitigate the damage suffered by the data subjects", "the nature, gravity and duration of the infringement" and its "intentional or negligent character" determine the amount of fines by authorities (Article 83 (2) GDPR).

Having an active data protection officer, even if not required by GDPR, indicates that an organization has no malicious intention in breaching the privacy rules. Therefore, taking the data protection rules and principles is the intelligent strategy.

THE ROLE OF THE DATA PROTECTION OFFICER

Defining the duties of the data protection office in organizations in terms of AI services is no different from those in terms of

processing data.

The accountability principle, enshrined in Article 5 (2) GDPR, considers the controller the first and final responsible person for implementing appropriate and effective measures and respecting the rights and requirements of a natural person (Recital 72). The controller shall be responsible for, and be able to demonstrate compliance with lawfulness, fairness and transparency of processing (Article 5 (1) a), purpose limitation (5 (1) b), data minimisation and storage limitation, accuracy, integrity and confidentiality of the processing.

Appointing a (Natural) Data Protection Officer (DPO)

Data centres, whether in a controller or processor role, need to appoint a DPO according to Article 37 (1) GDPR, because

1. the core activities of them consist of processing operations which, by virtue of their nature, their scope and/or their purposes, require regular and systematic monitoring of data subjects on a large scale; or
2. the core activities of the controller or the processor consist of processing on a large scale of special categories of data pursuant to Article 9 or personal data relating to criminal convictions and offences referred to in Article 10.

If a controller or processor's use of processing personal data is the key operation to achieve its objectives, it can be considered a core activity. An example is processing a patient's health record (health data) in the context of a hospital. On the other hand, paying employees is a supporting activity to the work of an organization and is considered an ancillary function.

The number of data subjects concerned, the volume of data and/or the range of different data items being processed, the duration, or permanence, of the data processing activity, and the geographical extent of the processing activity are elements that can determine whether processing is happening on a large scale. Processing of customer data for behavioural advertising by a search engine or processing of student data during semester by a university can be considered "large scale". On the other hand, processing of a patient's data by an individual physician does not constitute large-scale processing.

The data protection officer needs to be a natural person. Even if the data protection tasks are outsourced to a legal person, it must designate a specific natural person to act as the actual DPO contact. The reason is that many of the responsibilities assigned to the DPO can only be done by a natural person. They should be accessible to supervisory authorities and carry out their tasks, as established in Article 39 of the GDPR, personally. Imaginable scenarios where a future AI system can perform the tasks of a DPO are not permissible, as a data protection officer shall not receive any instruction regarding the exercise of their assigned tasks (Article 38 (3) GDPR).

Consulting on Data Protection Impact Assessments (DPIA)

Where a type of processing is likely to result in a high risk to the rights and freedoms of natural persons, the controller shall, prior to the processing, carry out an assessment of the impact of the envisaged processing operations on the protection of personal data (Article 35). Data protection impact assessments are

necessary at least in the following cases:

- a systematic and extensive evaluation of the personal aspects of an individual, including profiling;
- processing of sensitive data on a large scale;
- systematic monitoring of public areas on a large scale (Article 35 (3) GDPR).

The controller or the processor should seek the advice of the DPO on many issues, including: whether or not to perform a DPIA and, in case of necessity, the method that should be followed, consulting on technical and organizational measures that can be applied as safeguards to mitigate the risks to the rights and interests of the data subjects. The DPO is also in charge of consulting the controller/processor on the accuracy of the assessment and checking its conclusions with the data protection requirements [12].

The AI Act has mandated risk management systems in relation to high-risk AI systems (Article 9). These risks include, but are not limited to, the risks that might interfere with data subjects' privacy. Many of the automated decision-making systems may fall within the definition of high-risk systems. In these cases, the data protection officer must ensure that a data protection impact assessment is carried out, and appropriate safeguards are introduced when such systems are intended to be deployed. The DPIA documentation should explicitly state in writing why the advice has not been taken into consideration if the controller disagrees with the DPO's opinion [13].

Cooperating with the Supervisory Authority

The DPO plays the role of a 'facilitator' when it acts as a contact point to facilitate access by the supervisory authority to the documents and information for the performance of the tasks like enforcing the regulation, promote public awareness, conducting investigations, and handling complaints lodged by data subjects, as mentioned in Article 57 as well as for the exercise of its investigative, corrective, authorisation, and advisory powers mentioned in Article 58.

Data subjects may contact the data protection officer with regard to all issues related to the processing of their personal data and to the exercise of their rights under GDPR (Article 38 (4)). Because of this role, it is required to ensure that DPO's independence is guaranteed.

Advisory, Informative, and Monitoring Role within the Organization

The DPO is in charge of informing and advising the controller or the processor and the employees who carry out processing of their obligations set down by GDPR. An independent DPO is best situated to monitor compliance with the regulation, national and other Union data protection provisions, and the policies of the controller or processor on the protection of personal data (Article 39 (1) (a) and (b)). The assignment of responsibilities, awareness-raising, and training of staff involved in processing operations and carrying out the audits can form a part of the compliance strategy.

Record-Keeping

Both the controller and the processor must 'maintain a record of processing operations under its responsibility' or 'maintain a record of all categories of processing activities carried out on behalf of a controller.'" (Article 30 (1) and (2))

In practice, in many current national laws and EU institutions, the DPOs often hold a register of processing operations based on the information they receive from the responsible sectors within their organizations. Although this does not exempt the controller or the processor from the responsibility, assigning the DPO the task of maintaining the records of the processing operations is a tool that helps them in better playing their monitoring, informing, and advisory roles [14].

The records of processing required to be kept under Article 30 GDPR allow the supervisory authority to have an overview of all the data processing operations and therefore function as an effective accountability measure. Therefore, it is a compliance strategy.

CONCLUSION

Let us reconsider the drummer parable one more time. A man who analogizes his act of theft to playing a drum, whose sound will echo the next day. Violating privacy and infringing data protection rules is like trying to break into a house during the night. It may remain unnoticed for a while, but will have significant consequences when the contested act (stealing or data breach) is exposed during the day. The AI Act has no explicit rule on establishing an AI Office, while it came with different monitoring, record-keeping, transparency, and risk assessment burdens. Focusing on AI governance, however, will not lead to data protection rules being left out of the scope of surveillance. On the contrary, with the advent increase in AI systems and automated decision-making, the role of data protection officers in organizations has been intensified.

DPOs shall be appointed in organizations which is engaged in data processing as their core activities, when data subjects are systematically monitored, or sensitive data are processed on a large scale. The contact details of DPOs should be provided to the data subjects, and they should be vested with adequate powers to act independently. They contribute to the organization's mission by record-keeping, cooperating with supervisory authorities, acting as a contact point for safeguarding individuals' rights, advising the data controller or processor on several issues, including data protection risk assessment, and, in one word, keeping the organization compliant. When personal data is at stake, they help with guaranteeing that data protection rules are implemented properly. Even if negligently, the invisible drum is being played, they prevent it from being strident or scandalous.

BIBLIOGRAPHY

- [1] Available in Persian at: <https://ganjoor.net/moulavi/masnavi/daftar3/sh128>
- [2] Susana Coroado. Leviathan vs Goliath or States vs Big Tech and what the digital services act can do about it, Working Papers, Forum Transregionale Studien 25/2023, DOI: <https://doi.org/10.25360/01-2023-00038>
- [3] Recital 9 AI Act states: The harmonised rules laid down in this regulation should apply across sectors and, in line with the New Legislative Framework, should be without prejudice to existing Union law, in particular on data protection, consumer protection, fundamental rights, employment, and protection of workers, and product safety, to which this regulation is complementary.
- [4] Case C-634/21, 7 Dec. 2023, Available at: <https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:62021CJ0634>
- [5] Ibid, at para. 85: „ Article 22 (1) of Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) must be interpreted as meaning that the automated establishment, by a credit information agency, of a probability value based on personal data relating to a person and concerning his or her ability to meet payment commitments in the future constitutes ‘automated individual decision-making’ within the meaning of that provision, where a third party, to which that probability value is transmitted, draws strongly on that probability value to establish, implement or terminate a contractual relationship with that person.“
- [6] Is the DPO the right person to be the AI Officer?, CEDPO AI and Data Working Group, Micro-Insights Series July 2024, Available at: <https://cedpo.eu/wp-content/uploads/The-DPO-and-the-AI-Officer.pdf>
- [7] Case C 673/17, Bundesverband der Verbraucherzentralen und Verbraucherverbände – Verbraucherzentrale Bundesverband eV v Planet49 GmbH, ECLI:EU:C:2019:801, Judgment of the Court (Grand Chamber) of 1 October 2019, paragraph {63}, available at: <https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:62017CJ0673>
- [8] Case C 131/12, Google Spain SL and Google Inc. v Agencia Española de Protección de Datos (AEPD) and Mario Costeja González, ECLI:EU:C:2014:317, Judgment of the Court (Grand Chamber) of 13 May 2014.
- [9] Case C 73/07, Satakunnan Markkinapörssi Oy and Satamedia Oy v Finland, ECLI:EU:C:2008:727, Judgment of the Court (Grand Chamber), 16 December 2008, available at: <https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:62007CJ0073>
- [10] The obligations for providers of general-purpose AI models laid down in Chapter V of the AI Act apply from 2 August 2025 (Article 113, point (b), AI Act). However, under Article 111 (3) AI Act, ‘[p]roviders of general-purpose AI models that have been placed on the market before 2 August 2025 shall take the necessary steps in order to comply with the obligations laid down in this Regulation by 2 August 2027.’
- [11] Guidelines on the scope of the obligations for general-purpose AI models established by Regulation (EU) 2024/1689 (AI Act), Brussels, 18.07.2025, C(2025) 5045, paragraph 111.
- [12] Guidelines on Data Protection Officers (‘DPOs’) (wp243rev.01), Guidelines on Data Protection Officers (‘DPOs’), dopted on 13 December 2016, last Revised and Adopted on 5 April 2017, p. 15
- [13] The Guidelines on Data Protection Officers (‘DPOs’) states that: Article 24(1) provides that ‘*taking into account the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for the rights and freedoms of natural persons, the controller shall implement appropriate technical and organisational measures to ensure and to be able to demonstrate that processing is performed in accordanc with this Regulation. Those measures shall be reviewed and updated where necessary.*’
- [14] Ibid, p. 19. ■

Chat

Chat AI

KI SICHER UND ANONYM NUTZEN!

Ihre Anforderung

Sie suchen zur Unterstützung Ihrer täglichen Arbeit nach einem einfachen Chatbot auf dem aktuellen Stand der KI-Technologie, der keine Kompromisse beim Datenschutz macht.

Unser Angebot

Wir bieten mit Chat AI einen einfachen und sicheren Zugang zu leistungsstarker KI. Über eine intuitive Oberfläche können Sie direkt mit verschiedenen KI-Modellen chatten, Ihre Fragen stellen, Antworten und Unterstützung für Ihr Studium, Ihre Lehre oder Ihre Forschung erhalten. Die KI-Modelle können für die Verarbeitung von Nachrichten, Audioaufnahmen oder Dateien genutzt werden. Wenn Sie unsere internen KI-Modelle verwenden, werden Ihre Daten und Inhalte nur in Ihrem Browser gespeichert und bei uns direkt nach der Verarbeitung wieder gelöscht.

Ihre Vorteile

- > Genaue und relevante Informationen
- > Einfache Benutzeroberfläche

- > Laufende Verbesserung und Weiterentwicklung
- > Nutzung verschiedener KI-Modelle (u. a. Llama, Qwen, DeepSeek und Mistral)
- > Integration von OpenAI-Modellen möglich
- > Unterstützung bei technischen Problemen
- > Hohe Sicherheit und Datenschutz
- > Einfache Nutzung des RAG-Systems „Arcana“ zur Einbeziehung einer Vektordatenbank
- > Konfigurierbare, speicherbare und teilbare System-Prompts
- > Auf Anfrage API-Zugriff zur Einbindung eigener Werkzeuge

Interessiert?

Jede*r Nutzer*in mit einem Konto der Max-Planck-Gesellschaft oder der Universität Göttingen und Nutzer*innen einer teilnehmenden Hochschule der Academic Cloud können den Dienst „Chat AI“ nutzen. Sie benötigen lediglich einen aktuellen Webbrowser.

Spezifikationsgesteuerte Entwicklung von Anwendungen mit KI-Agenten

Text und Kontakt:

Thorsten Hindermann
thorsten.hindermann@gwdg.de

Agentische KI markiert einen Wendepunkt in der Softwareentwicklung – weg von klassischer Handcodierung, hin zu spezifikationsgesteuerter Entwicklung (engl. Specification Driven Development, kurz SDD). Der Artikel schildert praxisnahe Erfahrungen aus den Jahren 2025 und 2026 und zeigt, wie KI-Agenten Softwareentwicklung nahezu vollständig übernehmen. Entwickler*innen werden zu Supervisor*innen, Spezifikationen zum zentralen Artefakt und SDD zum neuen Standard für Qualität, Effizienz und Dokumentation. Ein Erfahrungsbericht über einen fundamentalen Rollen- und Paradigmenwechsel.

ERFAHRUNGEN MIT AGENTISCHER KI

Der Autor dieses Artikels beschreibt aus persönlicher Sicht, wie rasant sich die Anwendungsentwicklung nach über 40 Jahren (1985 bis heute) innerhalb weniger Monate verändert hat. Die hier geschilderten Erfahrungen basieren derzeit nahezu vollständig auf privaten „Herzens-Programmierprojekten“. In rund zwei Monaten (Januar bis Februar 2026) habe ich mich autodidaktisch von „händischer“ Programmierung hin zu spezifikationsgesteuerter Entwicklung mit dem Werkzeug KI-Agent umgestellt. Das war eine sehr aufregende Erfahrung – auch für mich als sehr erfahrener Entwickler, sowohl mit Blick auf Lebensalter als auch Berufserfahrung.

Die rasante Entwicklung künstlicher Intelligenz hat die Softwareentwicklung in den letzten Jahren tiefgreifend verändert. Besonders der Übergang von klassischer, gelegentlicher KI-Unterstützung in unserem Hauptwerkzeug, der integrierten Entwicklungsumgebung (engl. Integrated Development Environment, kurz IDE), hin zu agentischen Systemen und Specification Driven Development (SDD) markiert für mich einen Wendepunkt. In diesem Artikel beschreibe ich meine persönlichen Erfahrungen mit agentischer KI und zeige auf, wie SDD unseren Arbeitsalltag als Entwickler*innen nachhaltig verändert.

Ausgangssituation vor agentischer KI (bis 2025)

Bis ins Jahr 2025 hinein war KI für mich vor allem ein unterstützendes Werkzeug innerhalb klassischer Entwicklungsumgebungen. Ab Mitte 2025 (konkret ab Juni) testete ich verstärkt KI-Assistenten in JetBrains IDEs. Der Assistent zeigte zwar hilfreiche Ansätze, blieb insgesamt aber hinter meinen Erwartungen zurück.

Besonders positiv fielen automatisierte Git-Commit-Meldungen auf, die zuverlässig und konsistent erzeugt wurden. Auch die

Erstellung von XML-Kommentaren für C#-Code, meine favorisierte, hauptsächliche Programmiersprache, funktionierte sehr gut und brachte spürbare Erleichterungen und Zeitersparnis im Bereich der Dokumentation von Quellcode. Die automatische Generierung von Tests für einzelne Methoden war grundsätzlich brauchbar, erreichte jedoch nicht die Qualität, die für einen produktiven, durchgängigen Einsatz erforderlich gewesen wäre. Insgesamt waren die Ergebnisse bis Ende 2025 solide, aber noch nicht zufriedenstellend genug, um von einem echten Paradigmenwechsel zu sprechen.

Der Wendepunkt ab 2026: „Programmierung nahezu gelöst“

Ein grundlegender Wandel setzte ab Januar 2026 ein. In dieser Phase begann ich erstmals systematisch mit dem Einsatz und Vergleich verschiedener KI-Agenten, darunter Claude Code, OpenAI Codex, GitHub Copilot, Google Gemini und dem Open-Source-KI-Agenten OpenCode. Parallel dazu startete ich mehrere „Herzens-Projekte“, die sich hervorragend als Experimentierfelder eignen. Dazu gehören unter anderem folgende drei:

Specification Driven Development of Applications with AI Agents

Agentic AI represents a turning point in software development: moving from manual coding to Specification Driven Development (SDD). This article shares hands on experiences from 2025 and 2026 and illustrates how AI agents can implement complete features based on precise specifications. Developers evolve into supervisors, specifications become the single source of truth, and SDD emerges as a new standard for efficiency, quality, and compliance. A first hand account of a profound shift in how software is built.

1. TinyPI0

TinyPI0 [1] ist ein kompaktes Lehrprojekt zur Portierung des historischen PL/0 Beispielcompilers (inkl. Compiler/Parser und stackbasierter VM/Interpreter) von Pascal nach C# auf .NET 10. Es dient als praxisnahes Beispiel, um Compilerbau, Codegenerierung und die Ausführung in einer virtuellen Maschine nachvollziehbar zu verstehen.

2. TinyCalc

TinyCalc (MicroCalc) [2] ist ein Lehrprojekt zur Portierung der Borland-Beispielanwendung „MicroCalc“ (Turbo Pascal 3.02A) nach C# auf .NET 10 als textbasierte Tabellenkalkulation (TUI) auf Basis von Terminal.Gui. Es umfasst Kernlogik (Spreadsheet/Formula Parser, Persistenz/Export) sowie Unit-Tests und wurde als Beispiel für agentische KI + Spec-Kit/SDD erstellt.

3. TuiVision

TuiVision [3] ist ein kompaktes Beispielprojekt für textbasierte Benutzeroberflächen (TUI) in .NET: ein moderner, C#/.NET-basierter Ansatz im Stil klassischer Turbo Vision-Anwendungen, umgesetzt mit Terminal.Gui. Es dient als Experimentierfeld für typische TUI-Bausteine (Fenster/Dialogs, Menüs, Eingabefelder) und deren Architektur in einer „Console UI“.

Mein Vorgehen veränderte sich dabei grundlegend: Statt direkt mit der Implementierung zu beginnen, habe ich zunächst Lastenhefte in strukturiertem Markdown-Format erstellt. Diese dienten als Grundlage, auf deren Basis ich mit Prompts die KI-Agenten-Pflichtenhefte, ebenfalls im Markdown-Format, habe erstellen und verfeinern lassen. Die anschließende Umsetzung in C#-Quellzeilen erfolgte schrittweise in klar abgegrenzten Abschnitten aus den Pflichtenheften.

Bereits ab Februar 2026 kam dabei konsequent Specification Driven Development zum Einsatz. Zentral ist dabei aktuell die Nutzung des GitHub Spec-Kits [4], das einen klar strukturierten Ablauf vorgibt. Die Ergebnisse waren insbesondere mit kommerziellen Large Language Models, kurz LLMs, wie z. B. Claude Sonnet 4.6, Opus 4.6, GPT-5.3-Codex und GPT-5.4, beeindruckend präzise. Parallel dazu testete und teste ich OpenCode gezielt für den Einsatz in der Ausbildung der Fachinformatiker*innen für Anwendungsentwicklung und für Systemintegration bei der GWDG, um auch dort den Auszubildenden erste Erfahrungen und praktisches Arbeiten mit agentischer KI und SDD zu vermitteln. Das wird aus ihrem zukünftigen Arbeitsalltag nicht mehr wegzudenken sein.

Mit diesen beiden Werkzeugen, KI-Agenten plus SDD, ist die Programmierung von schwierigen Aufgaben nahezu gelöst, wie auch Boris Cerny, Erfinder und Hauptentwickler von Claude Code bei Anthropic, in diesem Interview ausgeführt hat, abrufbar unter [5].

Agentische KI und SDD in der Ausbildung

Ein für mich persönlich besonders wichtiger Teilbereich meiner Arbeit bei der GWDG ist die Unterstützung der Ausbildung von Fachinformatiker*innen der beiden Fachrichtungen Anwendungsentwicklung und Systemintegration. In Kürze bzw. mit dem Erscheinen dieses Artikels in der Mai-Ausgabe 2026 der GWDG-Nachrichten werde ich das Thema SDD gemeinsam mit den Auszubildenden schrittweise erarbeiten und in den Arbeitsalltag integrieren – bewusst mit OpenCode [6] als Werkzeug. Dieser KI-Agent ist vom KI-Beauftragten der GWDG auf meine Anfrage

hin für den Einsatz in der Ausbildung freigegeben worden. Als nächsten Schritt werden sich die Auszubildenden über das SAIA Dashboard [7] einen Chat AI API Key erstellen, um diesen in OpenCode und somit auf den Chat AI-LLMs, die bei der GWDG zur Verfügung stehen, nutzen zu können.

Das didaktische Vorgehen ist bewusst iterativ angelegt: Zunächst werden XML-Kommentare für Methoden und Klassen erstellt. Danach folgen erste kleine Implementierungen von Funktionalitäten sowie dazugehörige Methodentests (Unit-Tests). Darauf aufbauend setzen wir schließlich vollständige Features mithilfe von Specification Driven Development um. Mein Ziel ist es, den Auszubildenden frühzeitig ein strukturiertes, modernes Entwicklungsverständnis für diese neuen – und künftig alltagsbestimmenden – Technologien und Techniken zu vermitteln.

Wie agentische KI und SDD den Entwickleralltag verändern

Mit dem Einsatz agentischer KI verändert sich die Rolle von Entwickler*innen grundlegend. Unsere Hauptaufgabe wird zunehmend die eines Supervisors: Wir sind nicht mehr primär für jede einzelne Codezeile verantwortlich, sondern übernehmen die Rolle der „Human Intelligence“ (H) – also der Intelligenz, die den Gesamtprozess steuert, überprüft und bei Bedarf korrigiert.

Die klassische Handcodierung tritt in den Hintergrund. Stattdessen übernehmen KI-Agenten zunehmend die vollständige Implementierung von Features nach unseren Vorgaben in der darauf resultierenden Spezifikation. Die KI wird damit zu einer Art „digitalem Junior oder Peer-Entwickler“ – zuverlässig in der Ausführung, aber angewiesen auf klare Anweisungen und Kontrolle durch uns menschliche Entwickler*innen. Parallel zeichnet sich ein Wechsel von klassischen IDEs hin zu spezialisierten agentischen Entwicklungsumgebungen ab, die über CLI (Command Line Interface, deutsch: Kommandozeile), TUI (Text User Interface, deutsch: textbasierte Benutzeroberfläche) oder GUI (grafische Benutzeroberfläche) bedient werden. Für diese neue Werkzeugklasse etabliert sich zunehmend die Abkürzung ADE – Agent Development Environment (Agenten-Entwicklungsumgebung).

DER SDD-WORKFLOW MIT DEM GITHUB SPEC-KIT

Das GitHub Spec-Kit bildet das Rückgrat des hier verwendeten SDD-Ansatzes, auch wenn es nur eines von mehreren möglichen SDD-Workflow-Projekten darstellt, die im Internet und auf GitHub zur Verfügung stehen. Gerade mit Blick auf die Ausbildung ist dieser Ansatz aus meiner Sicht besonders übersichtlich, strukturiert und gut nachvollziehbar. Der Workflow folgt einer klaren Abfolge: *Constitution*, *Specification*, *Plan*, *Tasks* und *Implementation*. Optional können Schritte wie *Clarify*, *Analyze* oder *Checklist* ergänzt werden, um Unklarheiten zu beseitigen oder Qualität abzusichern.

Ein konkretes Beispiel hierfür ist z. B. das Projekt *TinyCalc*, bei dem das GitHub Spec-Kit zur Erweiterung (Implementierung einer erweiterten Formel-Bibliothek) zum Einsatz kommt und das unter [8] einzusehen ist. Die Vorteile dieses Ansatzes sind erheblich: Das YAGNI-Prinzip („You Aren’t Gonna Need It“) wird konsequent eingehalten, da ausschließlich nur das vom KI-Agenten implementiert wird, was in der Spezifikation definiert ist. Gleichzeitig entstehen eine exakte, nachvollziehbare Umsetzung der Anforderungen sowie eine sehr hochwertige Dokumentation, die in weiten Teilen

den Dokumentationsanforderungen und notwendigen Nachweisen der ISO 27001 für den Bereich der sicheren Softwareentwicklung gerecht wird.

BEISPIEL DES SCHRITTWEISEN ABLAUFES EINER FEATURE-IMPLEMENTIERUNG

Die nächsten klar definierten Schritte zeigen den Ablauf des SDD-Prozesses zur Implementierung eines Features in einem Beispielprojekt, das unter [9] zu finden ist. Dieses kleine Projekt wurde mithilfe der OpenCode CLI mit ein paar „Single-Prompts“ erstellt – also mit dem, was am Anfang der agentischen KI-Programmierung weitläufig als „Vibe-Coding“ bezeichnet wurde. Nun soll mithilfe von SDD ein weiteres Feature in dem Projekt implementiert werden, nämlich Lokalisierung (engl. Localization, kurz L10N, weil sich zwischen dem L und dem N in „Localization“ 10 Buchstaben befinden). Hierbei soll die Anzeige von Ausgaben neben Deutsch wahlweise auf die englische Sprache und im weiteren Verlauf auf weitere Sprachen umgestellt werden können. Somit sind das Feature und der Umfang definiert. Wie aber sieht die Umsetzung nun mithilfe von SDD aus? Dies soll im Folgenden beschrieben werden.

Verwendet werden OpenCode CLI, der erstellte API_KEY für Chat AI und das auf Chat AI gehostete LLM (engl. Large Language Model, deutsch großes Sprachmodell, kurz LLM) Qwen 3 Coder. Zusammen mit den Auszubildenden werden wir experimentieren, welche anderen bei der GWDG gehosteten LLMs sich für SDD noch eignen werden.

OpenCode kann unter [6] heruntergeladen werden. Hier kann zwischen verschiedenen Installationsmöglichkeiten ausgewählt werden. Meine bevorzugte Installationsmethode ist mit dem Paketverwaltungsprogramm *brew*, geeignet für macOS und Linux. Somit lautet also dann der Befehl *brew install anomalyco/tap/opencode*.

Der folgende Inhalt von *auth.json* muss noch um den Chat AI API_KEY geändert und nach *~/.local/share/opencode* kopiert werden:

```
{
  "chat-ai": {
    "type": "api",
    "key": "Chat-AI API_KEY"
  }
}
```

Beim folgenden Inhalt von *config.json* muss *baseURL* auf den Chat AI-URL geändert werden und das Qwen 3 Code LLM unter *models* eingetragen und nach *~/config/opencode* kopiert werden:

```
{
  "$schema": "https://opencode.ai/config.json",
  "provider": {
    "chat-ai": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "chat-ai",
      "options": {
        "baseURL": "https://chat-ai.academiccloud.de/v1"
      },
      "models": {
        "qwen3-coder-30b-a3b-instruct": { "name": "Qwen 3 Coder 30B A3B Instruct" }
      }
    }
  }
}
```

Als nächstes muss das GitHub Spec-Kit installiert werden. Die Anleitung dazu ist unter [10] erreichbar.

Nach diesem Schritt muss das Projektverzeichnis mit dem geklonten Beispielprojekt (*git clone https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool.git*) noch für SDD vorbereitet werden. Dazu in den Ordner des gerade geklonten Projekts wechseln und den folgenden Befehl ausführen: *specify init --here --ai opencode*

```
Warning: Current directory is not empty (46 items)
Template files will be merged with existing content and may overwrite existing files
Do you want to continue? [y/N]: y
```

Specify Project Setup

```
Project      sysinfotool
Working Path /Users/thorstenhindermann/RiderProjects/sysinfotool
```

Choose script type (or press Enter)

```
▶ sh (POSIX Shell (bash/zsh))
  ps (PowerShell)
```

Use ↑/↓ to navigate, Enter to select, Esc to cancel

Next Steps

1. You're already in the project directory!
2. Start using slash commands with your AI agent:
 - 2.1 `/speckit.constitution` – Establish project principles
 - 2.2 `/speckit.specify` – Create baseline specification
 - 2.3 `/speckit.plan` – Create implementation plan
 - 2.4 `/speckit.tasks` – Generate actionable tasks
 - 2.5 `/speckit.implement` – Execute implementation

Enhancement Commands

Optional commands that you can use for your specs (improve quality & confidence)

- o `/speckit.clarify` (optional) – Ask structured questions to de-risk ambiguous areas before planning (run before `/speckit.plan` if used)
- o `/speckit.analyze` (optional) – Cross-artifact consistency & alignment report (after `/speckit.tasks`, before `/speckit.implement`)
- o `/speckit.checklist` (optional) – Generate quality checklists to validate requirements completeness, clarity, and consistency (after `/speckit.plan`)

2_Zusammenfassung der Spec-Kit-Arbeitsschritte

Unter macOS/Linux wird `sh` ausgewählt, unter Windows `ps` (siehe Abbildung 1). Nach erfolgreicher Installation von Spec-Kit für OpenCode erscheint die folgende Übersicht mit den nächsten Schritten (siehe Abbildung 2).

Nachdem das Projektverzeichnis und der KI-Agent OpenCode vorbereitet sind, erfolgt die Arbeit über Spec-Kit Slash Commands, namentlich `/speckit.constitution`, `/speckit.specify`, `/speckit.plan`, `/speckit.tasks` und `/speckit.implement`. Ergänzend können bei Bedarf die optionalen Kommandos `/speckit.clarify`, `/speckit.checklist` und `/speckit.analyze` genutzt werden. In der Praxis hat sich gezeigt: Wenn man diese Schritte und auch die optionalen Schritte als Pflichtschritte, konsequent durchläuft, erhält der KI-Agent so präzise Vorgaben für die Implementierung, dass Halluzinationen der KI nahezu eliminiert werden.

Starten wir unsere agentische Entwicklungsumgebung mit dem Befehl `opencode` und beginnen anschließend mit den Slash-Commands, die in den folgenden Unterabschnitten beschrieben werden (siehe Abbildung 3).

Noch ein Hinweis vorab: KI-Agenten (oft auch „Coding Agents“ genannt), wie z. B. OpenCode, können besonders gut mit strukturierten Textinformationen arbeiten – insbesondere Dateien im Markdown-Format. Wie man diese Dateien dem KI Agenten zur Verarbeitung übergibt, sehen wir gleich.

Auch die Spec-Kit-Vorlagendateien der fünf hauptsächlichen und drei optionalen Schritte (siehe Abbildung 2) beinhalten strukturierten Markdown-Text [11].

Verfassung

Als erstes benötigt jedes Projekt eine Verfassung, in der nicht verhandelbare Grundsätze festgelegt werden – z. B., dass die Testabdeckung mindestens (besser: deutlich) über 70 % liegen soll oder dass im Projekt .NET 10 zum Einsatz kommt. Die Verfassung sollte dabei möglichst wenige technische Detailvorgaben enthalten; technisch wird es spätestens im Planungsschritt. Sie kann

```

/speckit.plan      Execute the implementation planning workflow us
/speckit.tasks    Generate an actionable, dependency-ordered task
/speckit.clarify  Identify underspecified areas in the current fe
/speckit.specify  Create or update the feature specification from
/speckit.analyze  Perform a non-destructive cross-artifact consis
/speckit.implement Execute the implementation plan by processing a
/speckit.checklist Generate a custom checklist for the current fea
/speckit.constitution Create or update the project constitution from
/speckit.taskstoissues Convert existing tasks into actionable, depende

/speckit.
Build Qwen 3 Coder 30B A3B Instruct chat-ai
  
```

3_Auflistung der Spec-Kit-Slash-Kommandos in einem KI-Agenten, hier OpenCode

```

/speckit.constitution Erstelle eine neue oder ändere die vorhandene
Verfassung des Projekts mit den Inhalten der Datei
@Verfassungsanforderungen.md
Build Qwen 3 Coder 30B A3B Instruct chat-ai
  
```

4_Aufruf des Spec-Kit-Slash-Kommandos, um die Spec-Kit-Verfassung für das vorliegende Projekt zu erstellen

außerdem jederzeit an neue Gegebenheiten angepasst werden und ist somit nicht „für immer“ in Stein gemeißelt.

In Abbildung 4 ist der Prompt zu sehen (und weiter unten im Absatz nachzulesen), der eine neue oder vorhandene Verfassung mit den Inhalten aus der referenzierten Datei erstellen bzw. ändern soll. Die Datei wird über ein vorangestelltes `@`-Zeichen eingebunden, in diesem Fall `@Verfassungsanforderungen.md` [12]. Aus Gründen der Barrierefreiheit hier noch einmal wörtlich der Satz aus der Abbildung:

`/speckit.constitution Erstelle eine neue oder ändere die vorhandene Verfassung des Projekts mit den Inhalten der Datei @Verfassungsanforderungen.md`

In Abbildung 5 ist ein Schnappschuss der Arbeit von OpenCode zu sehen. Insgesamt hat OpenCode zusammen mit dem LLM `Qwen 3 Coder 30B A3B Instruct` von Chat AI etwas über

Projektverfassung basierend auf Verfassungsanforderungen.md aktualisieren

23,969 (\$0.00)

- Nicht-funktionale Anforderungen
- Technische Rahmenbedingungen

2. Version wurde auf 1.2.0 angehoben (minor update) gemäß Semantic Versioning, da neue Prinzipien hinzugefügt wurden.

3. Die Governance-Regeln und alle anderen bestehenden Prinzipien wurden beibehalten.

Die aktualisierte Verfassung enthält nun alle Anforderungen aus der Verfassungsanforderungen.md und ist konsistent mit den bestehenden Strukturen und Richtlinien des Projekts.

Build · qwen3-coder-30b-a3b-instruct · 1m 5s

5_Ausschnitt des Endes aus der Zusammenfassung der Projektverfassung, nachdem das Slash-Kommando aus Abbildung 4 erfolgreich ausgeführt worden ist.

eine Minute an der Erstellung/Änderung der Verfassung gearbeitet.

Nun beginnt für uns als Entwickler*innen ein neuer Arbeitsschritt im Umgang mit KI-Agenten: die erzeugte Datei lesen, verstehen und Entscheidungen treffen. Bei der Verfassung ist das noch überschaubar; in den nächsten Schritten werden diese Tätigkeiten deutlich wichtiger. Die neue Verfassung ist unter [13] einsehbar.

Spezifikation

Im weiteren Verlauf werde ich die Eingabeprompts, die ich in OpenCode eingegeben habe, hier im Text einfügen, auch aus Gründen der Barrierefreiheit. Denn: Programmierung `#include<everyone>`

Als nächstes soll für das vom KI-Agenten zu implementierende Feature eine Spezifikation erstellt werden – wenn man so will ein Lastenheft („Was will der Auftraggeber?“) als Grundlage für das Pflichtenheft („Wie setzt der Auftragnehmer das um?“). Wichtig ist eine Formulierung, die sowohl für den KI-Agenten als auch für das mit ihm zusammenarbeitende LLM verständlich ist.

Dazu wurde der folgende Eingabe-Prompt eingegeben:

/speckit.specify Erstelle die Spezifikation für die Sprach-Lokalisierung mit den Informationen in der Datei @Lastenheft_L10N.md

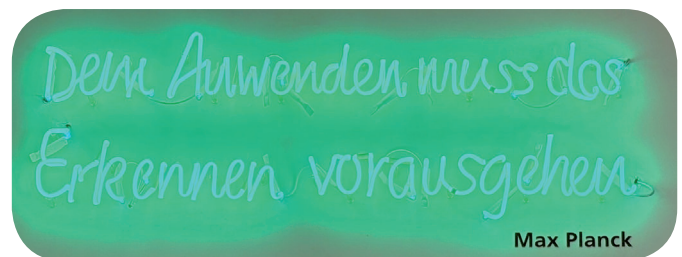
Die Eingabedatei ist unter [14] zu finden.

Nach gut einer Minute hat OpenCode aufgrund seiner Skripte einen neuen Git Feature Branch `003-110n-support` angelegt und im Verzeichnis `specs` einen Ordner mit gleichem Namen erstellt. Darin liegt die Datei `spec.md` [15].

Wo der KI-Agent nur eine Minute Arbeit hatte, haben wir als Supervisor nun viele Minuten Aufwand: Lesen, Verstehen und Entscheiden. Das ist ein neuer, zentraler Aspekt der zukünftigen Arbeitsweise von Anwendungsentwickler*innen. Deshalb sollten Auszubildende frühzeitig und kontinuierlich geschult werden, damit sie diese Handlungskompetenz aufbauen und sicher anwenden können.

Denn schon Max Planck – erster Präsident der Max-Planck-Gesellschaft – sagte einmal: „Dem Anwenden muss das Erkennen vorausgehen“.

Was enthält diese Datei im Wesentlichen? Der KI-Agent hat anhand der Spezifikationsvorlage u. a. drei „User Stories“, mehrere „Functional Requirements“ sowie „Success Criteria“ erstellt. Diese bilden die Grundlage für die weiteren SDD-Schritte und spätere



6_Leitspruch von Max Planck (im Harnack-Haus der MPG in Berlin aufgenommenes Foto)

Umsetzung.

An dieser Stelle sei daran erinnert, dass das Ergebnis ebenfalls eine Datei mit strukturiertem Markdown-Inhalt ist, die auch wir gut lesen und verändern können, sollte dies notwendig sein. In automatisiert erstellten Texten sollte man jedoch möglichst wenig manuell eingreifen. Auf dem Weg zur Feature-Implementierung folgen weitere Schritte, die der KI-Agent mit uns durchläuft und die uns zusätzliche Arbeit abnehmen. Unser „digitaler Junior oder Peer-Entwickler“ wird dabei nicht müde, sondern arbeitet zuverlässig weiter.

Klarheit

Das nächste Slash-Command ist in der Abbildung 2 zwar als optional markiert, sollte aus meiner Erfahrung aber verpflichtend ausgeführt werden. In mehreren Runden wird die Spezifikation damit möglichst „wasserdicht“ geschärft und abgerundet.

Entsprechend sollte `/speckit.clarify` mehrfach ausgeführt werden, bis keine Rückfragen mehr auftauchen und das Kommando den nächsten Arbeitsschritt von sich aus empfiehlt.

Die Klärung hat im ersten Lauf gleich folgendes Ergebnis erbracht:

No critical ambiguities detected worth formal clarification. The specification for the L10N feature is complete and clear. All functional requirements, user stories, and success criteria are well-defined and testable. The spec covers all necessary aspects of language localization including user scenarios, supported languages, automatic detection, fallback mechanisms, and technical requirements.

Auch hier gilt: Je genauer und präziser die Lastenheft-Datei gemeinsam mit einem Auftraggeber formuliert ist, desto besser kann der KI-Agent (mitsamt Spec-Kit-Vorlagen) damit arbeiten und exakte Ergebnisse liefern. Kurz: „Schlechte Eingabe, schlechte Ausgabe“. Sorgfalt lohnt sich – genauso wie „früher“ bei der „händischen“ Programmierung.

Plan

Im nächsten Schritt wird der KI-Agent den Plan für die Implementierung des L10N-Features ausarbeiten und erstellen.

Die Anweisung dafür liest sich wie folgt:

/speckit.plan Bitte erstelle den Plan unter Berücksichtigung der Dateien @Plan_Details.md und @DeveloperGuidelines.md

Die beiden Dateien sind unter [16] und [17] einsehbar.

Ein erstes Durchlesen der Dokumente aus diesem Schritt ergab, dass wir uns als Entwickler*innen für eine andere JSON-Bibliothek entscheiden als vom KI Agenten vorgeschlagen. Deshalb wurde der Plan mit folgender Anweisung korrigiert:

/speckit.plan Eine Änderung im Plan, Newtonsoft.Json soll ersetzt werden durch Microsoft.Text.Json

Die *plan.md*-Datei mit allen abhängigen Dokumenten ist vom KI-Agenten erstellt und verbessert worden und kann unter [18] eingesehen werden.

Da hier nun einige Dokumente neben *plan.md* erstellt worden sind, geht es nun erst einmal an das Sichten und Durchlesen der neuen Dokumente. Das kann je nach Umfang der Dokumente (abhängig vom Umfang der Spezifikation und vom verwendeten LLM) schon einige Zeit in Anspruch nehmen. Nachdem die Dokumente sorgfältig gelesen worden sind und vielleicht noch kleinere Verbesserungen manuell nachgepflegt wurden, geht es an die Prüfung des Plans.

Prüfung

Auch dieser als optional in Abbildung 2 aufgeführte Schritt sollte verpflichtend ausgeführt werden. Je präziser die Markdown-Dokumente aus jedem Hauptschritt werden, umso besser ist der daraus resultierende Code, da durch die präzisen Dokumente im Markdown-Format kein Spielraum mehr für die Halluzination des LLMs bleibt.

Mit dem Slash-Command */speckit.checklist* wird die Prüfung des Plans mit allen seinen Dateien ausgeführt. Aus diesem Aufruf resultiert eine umfangreiche Checkliste, die unter [19] einzusehen ist.

Nach der Erstellung der Checkliste wurde folgende klärende Frage gestellt:

prüfe @specs/003-l10n-support/checklists/localization.md gegen @specs/003-l10n-support/spec.md und @specs/003-l10n-support/plan.md. Decken diese Dokumente alle Items der @specs/003-l10n-support/checklists/localization.md ab?

Dabei kamen noch Unstimmigkeiten zu Tage. Mit folgendem Befehl/Prompt ist der KI-Agent aufgefordert worden, diese zu beseitigen:

Bitte ergänze für die vollständige Implementierung die fehlenden Details in den Spezifikationen.

Daraufhin hat der KI-Agent zurückgemeldet:

Diese Ergänzungen schließen die bisherigen Lücken in den Spezifikationen und stellen sicher, dass alle erforderlichen Informationen für eine vollständige Implementierung vorhanden sind. Die Checkliste kann nun vollständig abgedeckt werden, da alle benötigten Details dokumentiert sind.

Somit können wir nun zum nächsten Hauptschritt kommen, der Erstellung der Aufgaben für die spätere Implementierung des L10N-Features.

Aufgaben

Dieser Schritt ist mit dem folgenden Slash-Command

gestartet worden:

/speckit.tasks Erstelle die Aufgaben unter Berücksichtigung der Datei @DeveloperGuidelines.md

Zwischenzeitlich meldet der KI-Agent, was dieser tut:

Ich erstelle nun die Aufgabenliste unter Berücksichtigung der Developer Guidelines. Basierend auf den vorhandenen Dokumenten und den Richtlinien werden strukturierte Aufgaben für die Implementierung der Lokalisierung erstellt.

-> *Read specs/003-l10n-support/plan.md*

-> *Read specs/003-l10n-support/spec.md*

-> *Read specs/003-l10n-support/data-model.md*

-> *Read specs/003-l10n-support/contracts/command-contract.md*
~ *Preparing write...*

Nach mehreren Minuten Tätigkeit wurde folgende Vollzugsmeldung vom KI-Agenten angezeigt:

Alle Aufgaben sind nach den Developer Guidelines strukturiert und berücksichtigen die Anforderungen aus den Developer Guidelines sowie die Verfassungsanforderungen. Die Aufgaben sind so gegliedert, dass sie unabhängig implementiert und getestet werden können, was die Einhaltung der Testabdeckungsanforderungen (mindestens 80%) und die Einhaltung der Branching-Richtlinien gewährleistet.

Auch nach Beendigung dieses Schrittes geht es nun erst einmal an das Lesen der erstellten Datei *tasks.md* [20].

Analyse

Wie bei den anderen optionalen Schritten sollte auch dieser letzte Schritt vor der Implementierungsphase genutzt werden, um die zuvor erstellte Aufgabendatei und alle weiteren Dokumente noch einmal final zu prüfen. Ziel ist es, vor der Implementierung verbleibende Fragen, Unklarheiten oder Widersprüche zu identifizieren und zu beseitigen.

Auch dieser Schritt wird mit einem Slash-Command gestartet, konkret mit */speckit.analyze*

Das Ergebnis in diesem Fall ist, nochmals mit dem Update/Bereinigung der Spezifikation zu starten: *Run /speckit.specify with refinement to improve requirement clarity*

Das zieht bei Bedarf erneut mehrere Prozessschritte nach sich. Es zeigt aber auch, dass sich die Rolle als Entwickler*in zunehmend in Richtung Supervisor*in und Entscheider*in verschiebt, statt selbst jede Quellcodezeile zu schreiben. Gleichzeitig bleiben dadurch mehr Kapazitäten für die wesentlichen Tätigkeiten im Prozess: Lesen, Verstehen und Entscheiden.

Wichtiger Hinweis: Vor der Implementierung sollten die Prozessschritte (vollständig oder in Teilen) so lange wiederholt werden, bis der KI-Agent vor dem eigentlichen Implementierungsschritt keine offenen Fragen oder Unstimmigkeiten mehr meldet.

Ein abschließender Aufruf von */speckit.analyze* hat folgendes ergeben:

The implementation can proceed confidently with clear requirements and comprehensive testing coverage. All constitutional principles are maintained, and the feature is ready for implementation.

Nun ist alles vorbereitet, um den letzten Prozessschritt auszuführen: die Implementierung des L10N-Features durch den KI-Agenten.

Implementierung

Wichtiger Hinweis: Vor dem Ausführen des Slash-Commands für den Implementierungsschritt sollte der KI-Agent

verlassen und in einer frischen Sitzung neu gestartet werden. So hat der KI-Agent die Inhalte aus den vorherigen Schritten nicht mehr in seinem Kontext und wird bei der Implementierung des L10N-Features nicht durch bereits vorhandene Zwischenstände in seinem Kontext bei der Implementierung irritiert.

Nachdem der KI-Agent mit einer frischen Sitzung gestartet wurde, kann nun das einzig verbleibende Slash-Command eingegeben werden: `/speckit.implementation`

Mit diesem Slash-Command wird das „heavy code lifting“ angestoßen. Nach gut einer Stunde war die Arbeit erledigt und das L10N-Feature implementiert. Mit Nacharbeiten zur Fehlerkorrektur, die bei Testläufen auffielen, lag der Gesamtaufwand bei etwa zwei bis drei Stunden. Die Korrekturen wurden jeweils mit einzelnen ein- oder mehrzeiligen Prompts mithilfe des KI-Agenten durchgeführt.

Wichtiger Hinweis: KI-Agenten arbeiten im Normalfall so, dass sie vor potenziell kritischen Aktionen nachfragen, z. B. bevor sie Betriebssystem-Kommandos ausführen oder Dateien außerhalb des Projektordners bzw. des vorgesehenen Arbeitsbereichs verändern. Somit hat der/die Entwickler*in immer die Kontrolle über das Geschehen, so dass der KI-Agent nicht „eigenständig“ irgendetwas macht, über die wir keine Kontrolle haben.

Fazit

Zum Abschluss der Feature-Implementierung folgen einige statistische Zahlen und Berechnungen, um zu veranschaulichen, welche Auswirkungen der Einsatz von agentischer KI plus SDD auf unseren Alltag als Entwickler*innen hat und voraussichtlich haben wird.

Ich möchte an dieser Stelle betonen, dass niemand vor dieser Entwicklung Angst oder Sorge haben muss. Agentische KI wird uns nicht ersetzen, weil KI auch künftig primär auf statistischen Verfahren und Vorhersagen basiert – wenn auch mit stetig steigender Leistungsfähigkeit. Entscheidend bleibt: Wir Entwickler*innen sind die menschliche Intelligenz, die Ziele formuliert, Entscheidungen trifft und den Prozess steuert. Wie dieser Artikel zeigt, bleiben wir damit in der Steuerungs- und Entscheidungsschleife. KI unterstützt uns als Werkzeug, indem sie unsere Anweisungen ausführt. Ideen, fachliche Verantwortung und das Bild der gewünschten Umsetzung liegen weiterhin bei uns.

Unser Alltag wird sich aber grundlegend verändern – weg vom Schreiben einzelner Quellzeilen per Hand. Auch unsere Werkzeuge werden sich deutlich wandeln; gleichzeitig entstehen neue Freiräume. Während der agentische Helfer Arbeiten übernimmt, die häufig unliebsam oder monoton sind, können wir uns früher und stärker auf die nächsten Schritte konzentrieren. Im Ergebnis erhalten wir besser dokumentierten Quellcode, Projektdokumentation sowie Tests inklusive hoher Testabdeckung für Funktionen. Insgesamt sind hier deutliche Verbesserungen in mehreren Dimensionen zu erkennen und zu erwarten.

Nun aber zu einigen Zahlen und Berechnungen zur Einordnung (Hinweis: Dieser Abschnitt ist mit Unterstützung von KI erstellt worden):

Zeit- und Produktivitätsbetrachtung

Methodische Einordnung

Die nachfolgende Zeit- und Produktivitätsbetrachtung dient nicht dazu, individuelle Leistungen von Entwickler*innen zu bewerten oder Produktivität ausschließlich auf Codezeilen zu reduzieren.

Sie verfolgt vielmehr das Ziel, eine nachvollziehbare Vergleichsbasis zwischen traditioneller, handcodierter Entwicklung und spezifikationsgesteuerter Entwicklung mit agentischer KI zu schaffen – mit Fokus auf zeitlicher Beschleunigung und veränderten Rollen im Entwicklungsprozess.

Als Vergleichsmaßstab wird bewusst eine konservative, in der Praxis verbreitete Faustregel herangezogen: Ein*e erfahrene*r Entwickler*in schafft unter realistischen Bedingungen im Mittel etwa 80 manuell erstellte Zeilen pro Arbeitstag. Dieser Wert beinhaltet nicht nur Produktivcode, sondern den gemeinsam sichtbaren Lieferumfang aus Produktivcode, Tests und Dokumentation. Der Wert von 80 Zeilen pro Tag berücksichtigt bereits, dass ein erheblicher Teil der Arbeitszeit für Analyse, Fehlersuche, Tests, Dokumentation, Abstimmung und Recherche aufgewendet wird. Der Vergleichswert stammt aus einem Blogbeitrag, der unter [21] zu finden ist.

Für *sysinfotool* wird zusätzlich eine zweite, Repository-spezifische Referenz geführt: *Thorsten-Solo* mit 100 Zeilen pro Arbeitstag. Diese Zweitrechnung berücksichtigt jahrzehntelange .NET-/C#-Praxis (2,5 Dekaden) und dient als erfahrungsadjustierte Vergleichsbasis.

Ausgangsdaten und Annahmen

Für das Feature *003-110n-support* ergeben sich nach der neuen Methodik folgende objektiv messbaren Werte (siehe Tabelle 1).

| KENNZAHL | WERT |
|--|-------------------------|
| Gesamt-Netto nach neuer Methodik (Produktivcode + Testcode + Doku) | 1.620 Zeilen |
| Technische Teilmenge (C# + RESX netto) | 1.060 Zeilen |
| C# netto (Produktivcode + Tests) | 460 Zeilen |
| Produktivcode netto | 338 Zeilen |
| Produktivität konservative Referenz | 80 Zeilen / Arbeitstag |
| Produktivität Thorsten-Solo-Referenz | 100 Zeilen / Arbeitstag |
| Arbeitstage pro Monat | 21,5 |
| Arbeitszeit pro Tag | 7,8 Stunden |

Tabelle 1: Kennzahlen und Werte für die Feature-Implementierung *003-110n-support*

Die Einbeziehung von Tests und Dokumentation ist bewusst gewählt. Gerade bei spezifikationsgesteuerter Entwicklung gehören Spezifikationen, Aufgabenlisten, Dokumentationsnachweise und Testartefakte zum echten Lieferumfang und müssen bei klassischer Entwicklung ebenfalls manuell erstellt, gepflegt und geprüft werden.

Szenario 1: Voller Feature-Umfang nach neuer Methodik

Berechnung

- $1.620 / 80 = 20,25$ Arbeitstage
- $20,25 / 21,5 = 0,94$ Monate
- $20,25 * 7,8 = 157,95$ Arbeitsstunden

Gerundet ergibt das:

- 20,3 Arbeitstage
- 0,9 Monate
- 158,0 Arbeitsstunden

Dieses Szenario bildet den voll sichtbaren Lieferumfang des Feature-Branches ab: Implementierung, Tests und Dokumentation.

Szenario 2: Technische Teilmenge (C# + RESX netto)

Berechnung:

- $1.060 / 80 = 13,25$ Arbeitstage
- $13,25 * 7,8 = 103,35$ Arbeitsstunden

Gerundet ergibt das:

- 13,3 Arbeitstage
- 0,6 Monate
- 103,4 Arbeitsstunden

Dieses Szenario entspricht in etwa der früheren Betrachtung des technischen Implementierungsumfangs und schließt die Lokalisierungsressourcen bewusst mit ein.

Szenario 3: Reiner C#-Umfang (Produktivcode + Tests netto)

Berechnung:

- $460 / 80 = 5,75$ Arbeitstage
- $5,75 / 21,5 = 0,27$ Monate
- $5,75 * 7,8 = 44,85$ Arbeitsstunden

Gerundet ergibt das:

- 5,8 Arbeitstage
- 0,3 Monate
- 44,9 Arbeitsstunden

Dieses Szenario betrachtet ausschließlich die C#-Änderungen und blendet den signifikanten Spezifikations- und Dokumentationsanteil aus.

Ergänzende Zweitrechnung: Thorsten-Solo-Referenz

Für denselben vollen Feature-Umfang von 1.620 Zeilen ergibt die repo-spezifische Zweitrechnung mit 100 Zeilen pro Arbeitstag:

- $1.620 / 100 = 16,2$ Arbeitstage
- $16,2 / 21,5 = 0,75$ Monate
- $16,2 * 7,8 = 126,36$ Arbeitsstunden

Auch unter dieser erfahrungsadjustierten Annahme bleibt der Unterschied zur agentisch unterstützten Umsetzung erheblich.

Realistische Gesamteinordnung

Mit dem o. g. Wert von etwa 80 manuell erstellten Zeilen pro Arbeitstag ist für das Feature *003-110n-support* keine zusätzliche, pauschale Aufrundung mehr nötig, um Dokumentation und Tests mitzudenken. Sie sind bereits in der 1.620-Zeilen-Basis enthalten. Die frühere qualitative Schätzung von rund 20 Arbeitstagen wird damit jetzt durch den bereinigten Repo-Befund direkt bestätigt. Der klassische Aufwand für *003-110n-support* liegt damit auf konservativer Basis bei rund

- 20,3 Arbeitstagen
- 158,0 Arbeitsstunden

Vergleich mit agentischer KI und SDD

Die KI-gestützte Umsetzung desselben Features erfolgte mit spezifikationsgesteuerter Entwicklung weitgehend durch den agentischen Ablauf. Die aktive menschliche Arbeitszeit beschränkte sich im Wesentlichen auf:

- Erstellen und Nachschärfen der Spezifikationen
- Lesen, Prüfen und Entscheiden an den Prozessgrenzen
- Punktuelle Korrekturen nach Testläufen

Der dafür benötigte menschliche Zeitaufwand lag bei etwa 3-4 Stunden.

Wichtiger als eine tokenbasierte Abrechnung ist in diesem Kontext der Effekt auf den Entwicklungsprozess: Durch den agentischen Ablauf wird ein Feature trotz hoher Qualitätsanforderungen

(Tests, Dokumentation, nachvollziehbare Spezifikation) deutlich schneller realisiert, als wenn Entwickler*innen sämtliche Schritte selbst manuell ausführen müssten. Gleichzeitig verschiebt sich die aktive menschliche Arbeit weg vom „Tippen von Code“ hin zu höherwertigen Tätigkeiten: Anforderungen schärfen, Ergebnisse lesen und beurteilen, nächste Prozessschritte planen und die dafür nötigen Entscheidungen treffen.

Beschleunigungsfaktor

| KENNZAHL | WERT |
|--|-------------------|
| Klassisch, volle Feature-Basis nach neuer Methodik | ca. 158,0 Stunden |
| KI + SDD, menschliche Steuerungszeit | ca. 3-4 Stunden |

Tabelle 2: Vergleich der Arbeitszeiten je nach Programmiermethode

Daraus ergibt sich als Beschleunigungsfaktor

- bei 4 Stunden menschlicher Steuerungszeit $158/4 = 39,5$ bzw.
- bei 3 Stunden menschlicher Steuerungszeit $158/3 = 52,7$.

Das entspricht einer Beschleunigung durch agentische KI plus SDD um den (aufgerundeten) Faktor 40 oder 53. Dieser Faktor bedeutet nicht, dass KI einfach nur schneller Code schreibt. Er bedeutet vielmehr, dass durch SDD eine neue Abstraktionsebene entsteht: Der Mensch arbeitet primär konzeptionell und steuernd, während die KI einen großen Teil der operativen Umsetzung übernimmt und das deutlich schneller als bei herkömmlicher Entwicklung.

Schlussbemerkung zur Interpretation

Diese Zahlen stellen keinen allgemeinen, garantierten Produktivitätswert dar. Sie zeigen jedoch belastbar, dass sich durch agentische KI in Kombination mit konsequenter spezifikationsgesteuerter Entwicklung

- der menschliche Arbeitsfokus von Implementierung zu Steuerung verschiebt,
- sich Features in deutlich kürzerer Zeit umsetzen lassen
- und zugleich Dokumentations- und Testqualität sichtbar an Gewicht gewinnen.

Damit ist der Einsatz von agentischer KI plus SDD **kein Ersatz für Kompetenz von Entwickler*innen**, sondern eine Unterstützung und Beschleunigung ihrer Arbeitsweise bei der Entwicklung neuer Features für eine Software.

Ergänzung: Beschleunigung für das Gesamtprojekt

Wenn nicht nur das einzelne Feature *003-110n-support*, sondern das gesamte Projekt *sysinfotool* betrachtet wird, ergibt sich aus der laufenden Projektstatistik ein belastbarer Repo-weiter Beschleunigungswert.

Der aktuelle Gesamtstand des Repositories ist in Tabelle 3 dargestellt.

Das bedeutet:

- Gegen eine*n erfahrene*n Entwickler*in mit der konservativen 80-Zeilen-Referenz war die sichtbare Projektlieferung rund 13,2-mal schneller.
- Gegen die erfahrungsadjustierte Thorsten-Solo-Referenz mit 100 Zeilen pro Arbeitstag war sie rund 10,6-mal schneller.

| KENNZAHL | WERT |
|---|-------------------|
| Gesamtbasis für Handschätzung | 15.893 Zeilen |
| Git-Aktivtage | 15 |
| Konservative Referenz | 198,7 Arbeitstage |
| Thorsten-Solo-Referenz | 158,9 Arbeitstage |
| Repo-weiter Speedup vs. konservative Referenz | Faktor 13,2 |
| Repo-weiter Speedup vs. Thorsten-Referenz | Faktor 10,6 |

Tabelle 3: Beschleunigungsfaktor durch Mitwirkung von KI plus SDD im gesamten Projekt

Diese Gesamtprojektzahl ist bewusst anders zu lesen als die Feature-Einzelbetrachtung. Beim gesamten Repository handelt es sich um einen blended repository speedup, deutsch „gemischter (kombinierter) Beschleunigungsfaktor auf Repository-Ebene, also einen Vergleich von sichtbarem Lieferumfang zu sichtbaren Git-Aktivtagen. Das ist keine minutengenaue Zeiterfassung, sondern ein belastbarer Output-Indikator auf Repository-Ebene.

Dadurch erklärt sich auch der Unterschied: Das Einzel-Feature *003-110n-support* kommt auf einen deutlich höheren Faktor von rund 40 bis 53, weil dort die menschliche Steuerungszeit direkt mit 3-4 Stunden abgeschätzt wurde. Für das Gesamtprojekt liegt die sauber ableitbare Aussage deshalb beim Faktor 13,2 bzw. 10,6.

WEITERFÜHRENDE INFORMATIONEN

Wer nun Lust bekommen hat, agentische KI zusammen mit spezifikationsgesteuerter Entwicklung zu nutzen, dem sei dieses YouTube SDD Tutorial empfohlen. Es kann unter [22] abgerufen werden. Diese kurzweilige Serie gibt jedem Interessierten einen schnellen Über- und Einblick in die Nutzung von GitHub Spec-Kit.

Eine weitere Tutorial- bzw. Kurs-Quelle ist ein LinkedIn-Learning-Kurs [23]. Ein einem akzeptablen Zeitraum werden einem SDD und dessen Konzepte und Anwendung nahegebracht. Ergänzt sehr gut die zuvor genannte YouTube-Videoserie.

Zudem gibt es ein interessantes Video mit Anders Hejlsberg, der im Laufe seines Lebens vier wichtige Produkte federführend mitentwickelt hat: Turbo Pascal/Turbo Vision-Framework für MS-DOS, DELPHI/VCL-Framework für Windows, C#.Net-Framework für Windows und TypeScript, die auf JavaScript aufbaut und es um statische Typisierung erweitert. In dem Interview, das unter [24] abrufbar ist, macht er ab Minute 20 interessante und beachtenswerte Aussagen zu KI.

In einem weiteren YouTube-Video wird dargestellt, dass der Mensch als echte „Human Intelligence“ in der Entscheidungsschleife sein und bleiben sollte. Es werden auch andere Szenarien vorgestellt und verglichen. Aber HITL, engl. Human In The Loop, deutsch „Mensch in der Entscheidungsschleife“, bleibt das Modell der Wahl. Das Video kann unter [25] abgerufen werden.

Ein anderes YouTube-Video befasst sich mit der „besten Sprache“ für KI: der menschlichen Sprache – also unserer jeweiligen Muttersprache. In natürlicher Sprache formulieren wir Fragen und Anforderungen an den KI-Agenten. Dieser übersetzt sie dann in Anweisungen und Programme, die mit dem Computersystem interagieren. Der Vorteil: Diese „Programmiersprache“ müssen wir nicht erst mühsam erlernen – wir nutzen und verfeinern sie seit dem Spracherwerb in unserer Kindheit. Das Video kann unter [26] abgerufen werden.

Ein weiteres Video erläutert, dass sich die Arbeitsweise ändern wird und dies zu neuen Berufs-Spezialisierungen führen wird und dass von diesen Spezialist*innen eher mehr als weniger benötigt werden. Das Video kann unter [27] abgerufen werden.

NEUE WERKZEUGE

Ein Werkzeug ist nicht neu, erlebt aber eine Renaissance als „neues“ Werkzeug für die effektive Steuerung von agentischer KI: die Kommandozeile! Neben den KI-Agenten „sprießen“ immer mehr CLI-Werkzeuge (Command Line Interface, kurz CLI, deutsch Kommandozeile) aus dem Boden, da diese Programme von KI-Agenten aus sehr gut angesprochen und aufgerufen werden können, um die Aufgaben zu erledigen, die die agentische KI auf Grund der Frage- und Aufgabenstellung durch Entwickler*innen lösen soll.

Alle KI-Agenten, sowohl die CLI-, TUI- (Text-User-Interface) als auch die Graphical-User-Interface-Varianten, werden immer mehr zu Agent Development Environments, deutsch Agenten-Entwicklungsumgebungen, kurz ADE. Ein interessantes Konzept einer neuartigen ADE zeigt JetBrains mit der ADE *air*, die unter [28] abrufbar ist. Unter einer einheitlichen Oberfläche können mehrere KI-Agenten und deren LLMs angesprochen und gesteuert werden.

Weiterhin können mit agentischer KI und Unterstützung von SDD maßgeschneiderte und passgenaue Lösungen für bestimmte Anwendungszwecke schnell und präzise entwickelt werden. Ein Beispiel ist mein Public-Template-Projekt auf GitHub, mit dessen Hilfe sich Entwickler*innen einen agentischen Arbeitsbereich mithilfe eines Skripts anlegen können und eine Basisumgebung für agentische Entwicklung eingerichtet wird mit ersten Vorbereitungen für SDD. Das Projekt ist einseh- und abrufbar unter [29]. Erklärung und Anleitung, wie dieses Projekt genutzt werden kann, findet sich im README-Bereich, der unter [30] erreichbar ist.

Hier eine kurze Überblickszusammenfassung dieses kleinen Projekts, das ebenfalls darlegt, was mit agentischer KI und SDD in kürzester Zeit möglich ist:

Das Projekt *home-baseline* ist eine schlanke Vorlage („Baseline“) für einen reproduzierbaren Entwicklungs-Workspace auf dem eigenen Rechner: Es liefert eine vordefinierte Verzeichnis-/Konfigurationsstruktur, sodass sich ein neuer Arbeitsbereich schnell initialisieren lässt und man unmittelbar mit agentischer Entwicklung beginnen kann. Ziel ist, die wiederkehrenden Setup-Schritte (Grundstruktur, Basiskonfigurationen, Startpunkte für SDD-Workflows) zu standardisieren, damit man neue Projekte oder Experimente zügig und konsistent aufsetzen kann.

AUSBLICK

Zusammenfassend lässt sich sagen, dass agentische KI und spezifikationsgesteuerte Entwicklung Werkzeuge für einen grundlegenden Wandel in der Softwareentwicklung sind. SDD verändert Prozesse, Rollenbilder und letztlich auch unser Selbstverständnis als Entwickler*innen. Aus meiner Sicht besteht jedoch kein Anlass, diese Entwicklung zu fürchten – im Gegenteil: Es lohnt sich, die neuen Werkzeuge bewusst und verantwortungsvoll einzusetzen. Denn wir Menschen sind und bleiben die eigentliche Intelligenz (HI, Human Intelligence). KI-Agenten und die an sie angeschlossenen LLMs sind leistungsfähige Werkzeuge, die uns unterstützen. Und ja: Die „Geister, die wir riefen“, werden wir nicht einfach wieder los (in Anlehnung an Goethes „Der Zauberlehrling“: „Die ich

rief, die Geister, werd' ich nun nicht los.“). Wenn wir diese Werkzeuge jedoch mit klaren Regeln, Verantwortung und Kontrolle einsetzen, können und werden sie uns sehr, sehr gute Dienste leisten.

Die GW DG bietet auch verschiedene Dienste für die Unterstützung beim Agentic Coding an [31] und erweitert kontinuierlich ihr KI-Angebot.

LINKS

- [1] <https://github.com/hindermath/TinyPI0>
- [2] <https://github.com/hindermath/TinyCalc>
- [3] <https://github.com/hindermath/TuiVision>
- [4] <https://github.com/github/spec-kit>
- [5] https://youtu.be/We7BZVKbCVw?si=5JIE8MALIZJAhZ_x
- [6] <https://opencode.ai>
- [7] <https://saia.gwdg.de/dashboard>
- [8] <https://github.com/hindermath/TinyCalc/tree/main/specs/001-project-context>
- [9] <https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool>
- [10] <https://github.com/github/spec-kit?tab=readme-ov-file#-get-started>
- [11] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/master/opencode/command/speckit.specify.md?ref_type=heads
- [12] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/master/Verfassungsanforderungen.md?ref_type=heads
- [13] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/master/specify/memory/constitution.md?ref_type=heads
- [14] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/master/Lastenheft_L10N.md?ref_type=heads
- [15] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/003-l10n-support/specs/003-l10n-support/spec.md?ref_type=heads
- [16] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/master/Plan_Details.md?ref_type=heads
- [17] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/master/DeveloperGuidelines.md?ref_type=heads
- [18] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/003-l10n-support/specs/003-l10n-support/plan.md?ref_type=heads
- [19] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/003-l10n-support/specs/003-l10n-support/checklists/localization.md?ref_type=heads
- [20] https://gitlab-ce.gwdg.de/ausbildung/thorsten/sysinfotool/-/blob/master/specs/003-l10n-support/tasks.md?ref_type=heads
- [21] <https://www.holgerscode.com/blog/2026/02/23/adapt-or-disappear-how-ai-turned-a-2-year-project-into-a-1-week-sprint/#what-this-would-normally-cost>
- [22] <https://youtube.com/playlist?list=PL4cUxeGkcC9h9RbDpG8ZModUzwy45tLjb&si=BZTj9QLAT8v-a4Yj>
- [23] <https://www.linkedin.com/learning/spec-driven-development-with-github-spec-kit>
- [24] <https://youtu.be/uMqx8NNT4xY?si=IA1acwaknpvROJiv>
- [25] https://youtu.be/9iS-YYLIXiw?si=ZC1_yhRUJvtLQTRY
- [26] <https://youtu.be/g9UlamT4Pgc?is=uTwsGU3SYpc29nAy>
- [27] <https://youtu.be/XVNH8MPRgVY>
- [28] <https://air.dev>
- [29] <https://github.com/hindermath/home-baseline>
- [30] <https://github.com/hindermath/home-baseline#home-baseline>
- [31] <https://docs.hpc.gwdg.de/services/ai-services/commercial-models/agentic-coding/index.html>



Open-Source Operating Systems: More than just Linux?

Text and Contact:
Dr. Konrad Heuer
konrad.heuer@gwdg.de

Most people using notebooks or desktop computers for work or private purposes rely on Microsoft Windows or macOS as their preferred operating systems. Only a small fraction uses Linux, Chrome OS or other systems. Nevertheless, the name Linux is meanwhile well-known within a larger community of computer users. Linux has become a well-established term and synonymous for open-source software, and Linux has undoubtedly become a serious option for notebooks, gaming PCs and office workstations. But there is more than Linux, and this article takes a look without claiming to be exhaustive also at other open-source operating systems.

UNIX

Nearly all open-source operating systems have a common origin. The UNIX operating system was born within the AT&T Bell Laboratories in 1969 and was made easily available for universities till the beginning of the 1980s. Since UNIX was nearly completely written in the higher programming language C, it was an attractive project for academic research. Especially the Computer Science Research Group (CSRG) of the University of Berkeley in California contributed significantly to the development of UNIX. In particular, the introduction of a virtual address space and the TCP/IP protocol suite should be mentioned, and the UNIX variants developed in Berkeley and named BSD became very popular.

At the end of the 1980s and the beginning of the 1990s, AT&T began to market their then current version UNIX System V Release 4 (SVR4) aggressively. The CSRG published their last version 4.4BSD in 1993 as open-source, but AT&T started legal actions against the University of California which where (without going into details here) resolved a year later when SVR4 had been sold to Novell.

It seems to be totally absurdly (not to say idiotically) from today's point of view, that AT&T's restrictive approach initiated amazing developments that ultimately killed the original UNIX operating system including its last and high-developed version SVR4.

MINIX

As it became more and more obvious that AT&T changed its policy of making the UNIX source code available for teaching at universities, the American computer scientist Andrew S. Tanenbaum developed a clone of UNIX Version 7 and called it MINIX. UNIX Version 7 particularly did not include TCP/IP networking and virtual memory, and MINIX is thus a kind of "mini UNIX". It is still available as open-source software, but has not a great relevance anymore.

GNU/LINUX

Linus Torvalds, a Finnish computer science student, was not satisfied with the limited capabilities of MINIX. In the meantime, in 1991, PCs build on the Intel 80386 processor had become available. This processor enhanced the so-called "protected mode", offered preemptive multitasking and enabled demand-paged virtual memory. He started a hobby project of coding an operating system kernel based on MINIX and on UNIX concepts providing these capabilities. Soon the kernel was named Linux by his supporters. In 1992 the Linux kernel was licensed as open-source using the GNU General Public License (GPL).

GNU is a recursive acronym meaning "GNU not UNIX". It is an open-source project which was started by the US programmer Richard Stallman in 1983 with a similar motivation like Andrew S. Tanenbaum had. UNIX had become rather popular in the 1980s, and Richard Stallman wanted to create a true open-source alternative to UNIX. In the early 1990s, the GNU project had been

Open-Source-Betriebssysteme: Mehr als nur Linux?

Die meisten Nutzer*innen von Notebooks oder Desktop-Computern – ob beruflich oder privat – setzen auf Microsoft Windows oder macOS als bevorzugtes Betriebssystem. Nur ein kleiner Teil verwendet Linux, Chrome OS oder andere Systeme. Dennoch ist der Name Linux in der breiten Gemeinschaft von Computernutzenden mittlerweile sehr bekannt. Linux hat sich als etablierter Begriff und Synonym für Open-Source-Software etabliert und ist zweifellos zu einer ernstzunehmenden Option für Notebooks, Gaming-PCs und Büroarbeitsplätze herangewachsen. Doch es gibt mehr als nur Linux, und dieser Artikel wirft ohne Anspruch auf Vollständigkeit einen Blick auch auf andere Open-Source-Betriebssysteme.

successful in coding a UNIX-like “userland”, but still missed a kernel. Thus Linux and the GNU utilities made a great combination, and in 1993 the first so-called Linux distributions like Slackware and Debian were made available for general use.

The GPL is a very liberal open-source license allowing everyone to use and modify the code, but with the obligation to make any changes also available in source code under the GPL.

Today, more than 30 years have passed by, and hundreds of different Linux distributions have appeared since 1993, and dozens of them have already been discontinued. Famous and well-known distributions are Debian, Fedora, Manjaro, Mint, Red Hat, Rocky, SUSE or Ubuntu Linux. These distributions all use the Linux kernel and the GNU utilities, but usually with some extensions and modifications. Slackware Linux is also still available and the oldest active distribution.

Over the years, Linux experienced massive commercial interest and support. It evolved to a mature and fully functional operating system for server as well as desktop systems and notebooks. Undoubtedly, it can be said that Linux is the main reason why UNIX SVR4 vanished and Microsoft Windows didn't get the dominant server operating system. Furthermore, Android and ChromeOS are Linux descendants. On the other hand, if Linux hadn't made it, the open-source BSD family could have.

386BSD

As already mentioned above, the CSRG at Berkeley tried to get its version of UNIX open-source before the group was dissolved in 1995. In the 1980s, 4.2BSD and 4.3BSD were very successful operating systems which were used by commercial companies like Sun Microsystems and Digital Equipment as bases for their own UNIX systems called SunOS and ULTRIX. However, 4.2BSD and 4.3BSD contained AT&T code, and using them required to buy an AT&T UNIX license.

On the way to open-source 4.4BSD, the CSRG released the intermediate NET/2 release, which was in 1992 used by William and Lynne Jolitz for an implementation of BSD UNIX on PCs with 80386 processor. This project was called 386BSD, and unfortunately, the project was slow. Linus Torvalds once mentioned, that if 386BSD had been more widely available or easier to use at the time, Linux might not have been created. Today, 386BSD is only of historical interest.

The open-source BSD license is even more liberal than the GPL. As with GPL, the code can freely be used and modified, but modifications don't need to be published in source code. It has only to be mentioned that BSD code has been used. Rumors suggest that for example Microsoft has also use of the BSD TCP/IP stack, too.

FREEBSD

FreeBSD emerged as a fork of the 386BSD project, and in late 1993, Version 1.0 was released. Caused by the legal actions of AT&T against the University of California in Berkeley mentioned above, some dark clouds overshadowed the project. As a solution FreeBSD 2.0, which appeared 1994, was based on 4.4BSD-Lite, which was classified as unproblematic under licensing law. Because 4.4BSD-Lite missed some code and was not a complete operating system, some additional coding work had to be done by the project team before FreeBSD 2.0 could be released.

The FreeBSD project focused on the Intel 80386 platform and its Intel and AMD successors, although in the meantime some more hardware architectures like ARM are also supported. Major goal of the project is to provide an open-source UNIX-like system with high performance and maximal stability.

Like Linux, FreeBSD today has for more than 30 years continually been improved, and 4.4BSD was already in the 1990s a mature operating system. Commercial interest in FreeBSD is not as large as in Linux, but should not be underestimated; macOS by Apple and the Playstation Orbis OS by Sony are for example based on FreeBSD. The streaming servers of Netflix are run by FreeBSD, and Netflix sponsors the project significantly, too.

Linux and FreeBSD both are very respectable and powerful operating systems. In popularity, of course, Linux is the top dog, and FreeBSD remains the eternal runner-up by a wide margin. Basically, FreeBSD and Linux are at the same level as server operating systems, but Linux is easier to use and with significant advantage in hardware and software support on notebooks and desktop systems.

NETBSD

NetBSD started as a 386BSD fork at the same time as FreeBSD did. But the project goals are different. The NetBSD team aims for maximal portability and tries to support a couple of different hardware architectures. Thus NetBSD is used for example in embedded systems like Ricoh printers or Apple airport stations.

OPENBSD

In turn, OpenBSD is a NetBSD fork which evolved in 1995. The project team led by Theo de Raadt concentrates on security and code correctness. Some important and well-known software components like OpenSSH, LibreSSL and the PF firewall have been created in OpenBSD and migrated to other operating systems.

DRAGONFLY BSD

Around the turn of the millennium and the following years, FreeBSD versions 4.x were famous for stability and robustness. But similar to Linux at that time, there was a serious weak point. Multi-processor and multi-core support were rudimentary compared to Microsoft Windows or commercial UNIX systems. Only one process at any time could use kernel services, e.g. for any kind of data input or output. The kernel used a “big giant lock” to ensure this. The FreeBSD developers worked hard to achieve a much better solution in FreeBSD 5.x, but making the kernel “re-entrant” was anything else than easy and lasted several years.

Matthew Dillon, a FreeBSD developer, was dissatisfied with the path taken with FreeBSD 5.x, and created a FreeBSD fork which he named DragonFly BSD. DragonFly BSD uses different and simpler algorithms for multi-processor and multi-core support than other operating systems, and it uses its own file system with advanced capabilities, called HAMMER2.

ILLUMOS, OPENINDIANA AND OMNIO

Sun Microsystems sold UNIX-based workstations and servers from the 1980s till about 2010. In 1992, Sun switched from the BSD-based older SunOS to a SVR4 operating system called Solaris.

Sun was in 2010 acquired by Oracle. Some years before, Solaris was largely made open-source by Sun Microsystems. Oracle returned to a closed source policy, and OpenSolaris was discontinued.

To salvage the efforts made so far, the community started the Illumos project as a descendant. OpenIndiana, OmniOS and a few other projects are based on Illumos. These systems preserve the heritage of SVR4, although their practical relevance is limited.

It is very important to mention that a crucial component of the former OpenSolaris and its descendants made its way into FreeBSD and, with some additions, into Linux, too. The so-called Zettabyte File System (ZFS) is a very advanced and powerful file system, also suitable for large amounts of data. The current open-source implementation is called OpenZFS. Due to license differences compared to the GPL, OpenZFS cannot be an integral part of the Linux kernel and requires some additions. This is no problem for FreeBSD due to the more liberal BSD license, and therefore ZFS has nearly become the standard for FreeBSD servers.

FREEDOS AND REACTOS

These two operating systems do not belong to the UNIX family. FreeDOS is an MS-DOS compatible system with functional extensions like networking and still actively developed. It is of course useful, e.g. because some manufacturers ship their notebooks with FreeDOS to avoid buying a MS Windows license, or because FreeDOS is also needed to run legacy business or retro gaming software or to operate on specialized hardware.

Creating an open-source operating system compatible to MS Windows is much more challenging. Despite of already 30 years of continuous work, ReactOS is still in alpha stage and not ready for production use.

CURRENT SITUATION

Having read all this, where are we today? The spread and popularity of Linux is undoubtedly so overwhelming that the alternatives seem at first glance to have only a niche significance. But let's pause for a moment and look at commercial sponsoring. As already explained before the FreeBSD project is significantly supported directly or indirectly by Apple, Sony, NetApp, Netflix and others and is thus the second most important open-source operating system after Linux.

OpenBSD is sponsored by companies interested in IT security solutions. Even Microsoft had contributed financial resources to the project in the past. NetBSD, DragonFly BSD and OpenIndiana are mostly sponsored by the community and not by larger vendors.

Some people may say there is no need for any other open-source operating system than Linux. From a certain point of view, this is understandable. Linux can rely on large IT companies supporting the system with financial and human resources. Nearly all modern technological trends are available, and many software companies also sell their products for Linux. Thousands of individuals contribute to the kernel, and Linus Torvalds and a group of subsystem maintainers review changes and are responsible for quality. Having mentioned all that, there is no reason to list further arguments for choosing Linux.

But one should not underestimate that FreeBSD is a powerful operating system optimized for high performance servers and network nodes. It has a mature network stack and, for example, contains advanced firewall solutions and intrinsically, the famous

ZFS file system. Whereas Linux is, to put it bluntly, just a kernel with varying userland depending on the distribution, FreeBSD is a complete and homogeneous system composed of kernel and userland. Despite all necessary modernization, FreeBSD development is conservative with respect to administration without major breaks. And last not least, the permissive BSD license is commercially attractive because the system can be used in any way without obligations other than mentioning that FreeBSD code is included. Many hundred people are involved in the project, and some companies employ FreeBSD developers. In summary, there is enough evidence for FreeBSD, too.

Why are the other mentioned systems still alive? First of all, because enough people are interested in those projects. No other operating system supports more hardware environments like NetBSD, and no other operating system focuses more on security like OpenBSD. And because OpenIndiana and related are the last representatives of UNIX SVR4, they have their justification for existence.

CONCLUSION

The initial question of whether there is more to say about open-source operating systems than Linux can undoubtedly be answered with a yes. Anyone interested in open-source operating systems should, if possible, take their time to look at several alternatives and not commit prematurely. Of course, some constraints, such as the required and available commercial software, cannot be circumvented. It makes no sense to run HPC systems by any other operating system than Linux due to often required special device drivers or optimizing compilers, and modern tools like Kubernetes and Docker are strongly associated with Linux. On the other hand, using FreeBSD for networking appliances or as a ZFS file server may be a good idea. The range of open-source application software is usually by no means limited to Linux.

The GWDC relies heavily on open-source operating systems for providing services, and that will probably surprise nobody. That's not because for example Microsoft Windows server operating systems are worse in functionality and reliability, it is just because it is commonplace (and cheaper) to use open-source operating systems for many purposes today.

And what about the desktop? The author started using Linux and FreeBSD on the desktop in 1994 with Slackware Linux and FreeBSD 2.0. Newer releases followed, and both systems have nearly always worked for me. Since video conferencing has become standard at least since the COVID pandemic, Linux has an advantage on the desktop or notebook for some years now due to its better support for audio and video or other devices as well as video conferencing software.

For someone interested in taking their first steps with open-source operating systems, easy-to-install Linux distributions are often a good choice, and with solutions such as VirtualBox as an example, this can also be done risk-free in a virtual machine on the desktop/notebook.

References to literature and websites on the internet have been deliberately omitted in this article. Printed books dealing with the topics discussed here are often quickly outdated, and, to be honest, information in the world-wide-web about open-source operating systems can very easily be found via search engines, Wikipedia or AI. ■



GWDDG Pad

KOLLABORATION LEICHT GEMACHT!

Ihre Anforderung

Sie möchten allein oder gemeinsam mit Ihrem Team unkompliziert an Textdokumenten arbeiten oder Präsentationen erstellen und dabei auf eine Vielzahl nützlicher Funktionen zurückgreifen. Ihre Änderungen sollen sowohl für Sie als auch Ihre Teamkolleg*innen direkt und in Echtzeit einsehbar sein. Sie möchten die Lese- oder Schreibberechtigung für Ihre Dokumente einschränken können, sodass Sie Ihre Daten vor unbefugtem Zugriff schützen können. Zudem wollen Sie in der Auswahl Ihrer Endgeräte flexibel sein, sowohl mobiler Zugriff als auch Desktop-Varianten sollen unterstützt werden.

Unser Angebot

Auf Basis der freien Software „HedgeDoc“ bieten wir Ihnen einen Dienst, mit dem Sie schnell und unkompliziert Dokumente erstellen, mit anderen Personen teilen und gemeinsam bearbeiten können.

Ihre Vorteile

- > Kollaborativer Echtzeit-Editor
- > Übersicht über alle Ihre Dokumente nach Login

- > Unterstützung von UML-Diagrammen, mathematischen Formeln, Syntax-Highlighting, Musiknoten und vielem mehr
- > Modi zum Erstellen und Vorführen von Präsentationen
- > Einbinden externer Ressourcen wie Videos, PDF-Dateien oder SlideShare
- > Autovervollständigungs-Funktion für Markdown-Ausdrücke
- > Zugriffsbeschränkungen für jedes Dokument einstellbar
- > Veröffentlichung von Dokumenten möglich
- > Webbrowser ausreichend zur Benutzung, keine weitere Installation von Software nötig

Interessiert?

Wenn Sie unseren Dienst „GWDDG Pad“ unter <https://pad.gwdg.de> nutzen möchten, benötigen Sie lediglich einen aktuellen Webbrowser. Um eigene Dokumente erstellen zu können, ist zusätzlich die Verwendung eines gültigen GWDDG-Accounts oder die einmalige Registrierung unter <https://www.gwdg.de/registration> erforderlich.

Die GWDG bei Science goes City

Text und Kontakt:

Tilman Happek
tilman.happek@gwdg.de

Sophia Wolf
sophia.wolf@gwdg.de

Beim Aktionstag „Science Goes City“ am 19. April 2026 kam die Wissenschaft zu den Menschen in die Göttinger Innenstadt und verwandelte sie erneut in eine lebendige Wissenschaftsmeile. Auch die GWDG nahm wieder daran teil, diesmal mit einer interessanten Präsentation, die die unsichtbaren Wege von Internetdaten sichtbar machte, und freute sich über zahlreiche Besucher*innen und spannende Gespräche an ihrem Stand.

WISSENSCHAFT ZUM ANFASSEN BEI SCIENCE GOES CITY

Am 19. April 2026 fand in der Göttinger Innenstadt der Aktionstag „Science goes City“ (<https://science-goes-city.de>) statt. Groß und Klein waren eingeladen, Wissenschaft zum Anfassen direkt vor Ort in der Innenstadt und den Göttinger Geschäften zu erleben. Die zahlreichen Besucher*innen konnten an über 60 Stationen faszinierende Experimente, Shows, interaktive Präsentationen und aktuelle Forschungsarbeiten zu spannenden Themen hautnah erleben. Durch die Kombination mit einem verkaufsoffenen Sonntag hatte der Göttinger Einzelhandel zudem seine Türen geöffnet, sodass die Gäste das vielfältige Programm optimal mit einem gemütlichen Einkaufsbummel verbinden konnten.

PRÄSENTATION DER GWDG ZU DATENFLÜSSEN IM INTERNET

Wie schon beim letzten Mal, war die GWDG auch in diesem Jahr wieder dabei, diesmal mit einer interessanten Präsentation zu Datenflüssen im Internet. Der Standort dafür im neu eröffneten Modehaus „Vockeroth“ gegenüber dem Gänseliesel erwies sich als sehr günstig und zog zahlreiche interessierte große wie kleine Besucher*innen an.

Das World Wide Web ist unersetzlicher Begleiter unseres Alltags geworden. In allen Bereichen verwenden wir das Internet, um Informationen zu finden, auszutauschen oder miteinander zu kommunizieren. Durch diese intensive Nutzung stellt sich die Frage, wie viele Daten werden tagtäglich versendet? Wie lange dauert das und durch welche Länder fließen diese Daten? In einer kleinen Demonstration wurden diese Datenflüsse anschaulich dargestellt.

Um Datenflüsse sichtbar und auf einen Blick verständlich zu machen, hatten wir ein Programm geschrieben, das mit einer Grafik ausgestattet ist und die Datenströme des Internets visualisiert. Dabei mussten natürlich auch die Herausforderungen des Datenschutzes beachtet werden, da hierbei auch zu sehen ist, über wie viele Server z. B. eine Nachricht läuft, bevor die finale Website erreicht wird.

Am Ende haben wir auf einer Weltkarte die Unterseekabel visualisiert und einige der weltweit größten Internet Exchange

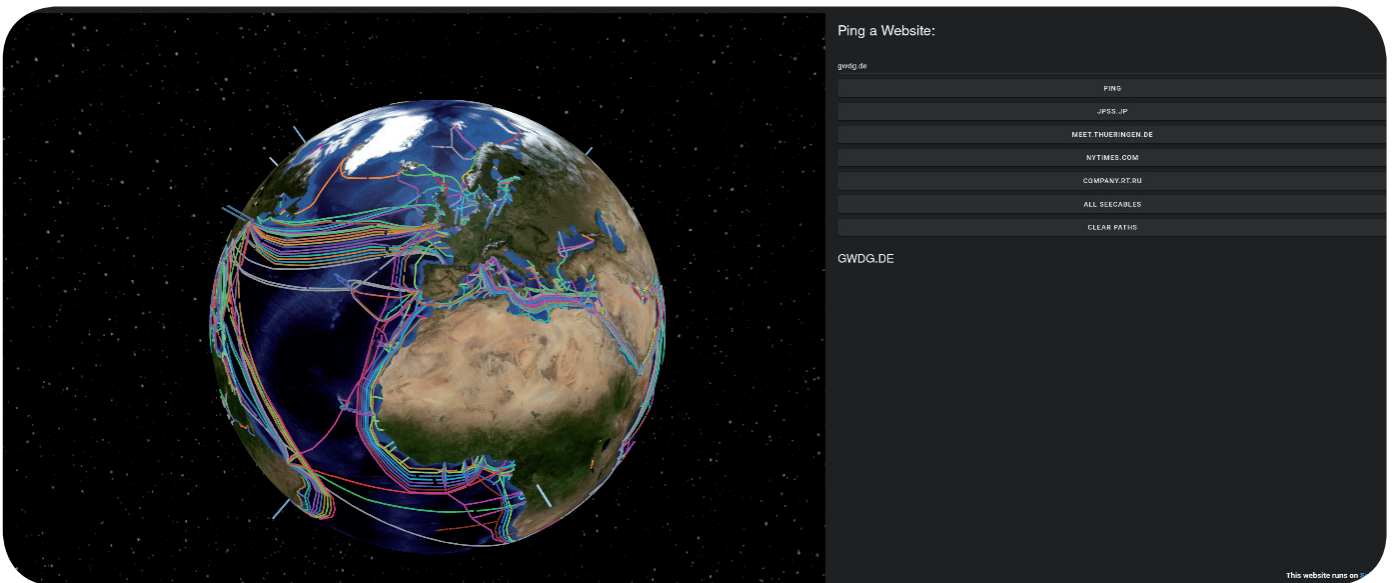


Points hinzugefügt, wie man in Abbildung 1 sehen kann. So konnten wir den Besucher*innen zeigen, dass, egal welche Website sie angepingt haben, der erste Weg in das Göttinger Rechenzentrum ging, oft gefolgt von Frankfurt und dann meist zu einem Unterseekabel.

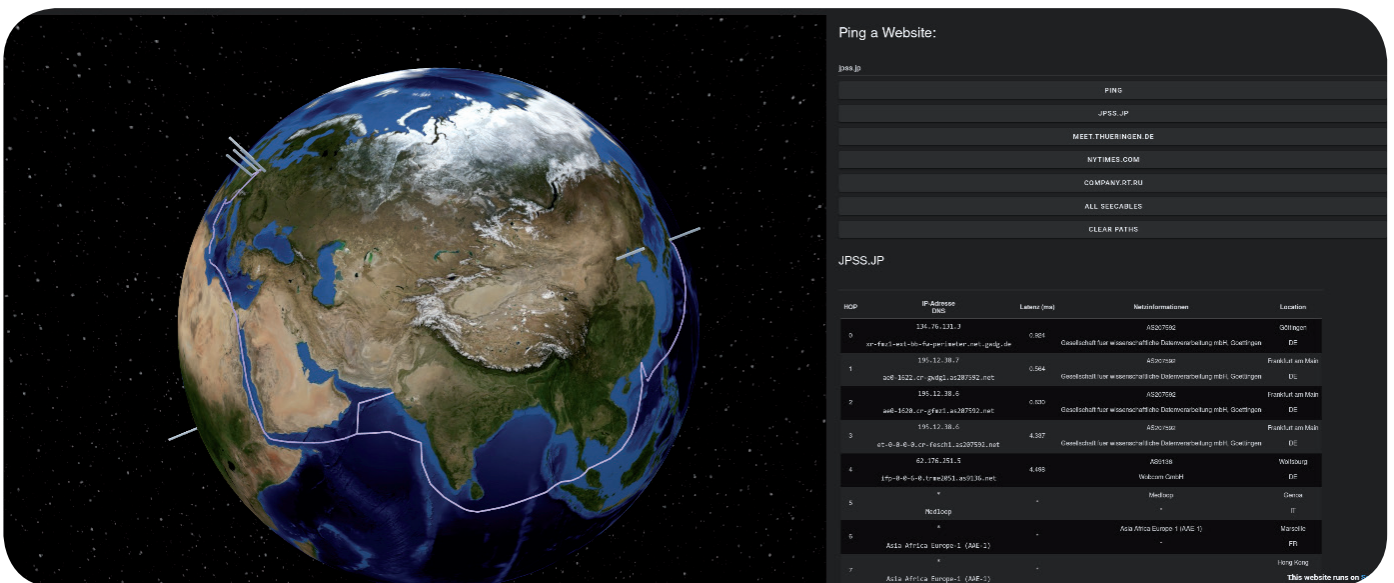
Bereits vorab wurden einige Websites als Voreinstellung vorbereitet. Dazu gehörte die Website der New York Times, deren

The GWDG at Science goes City

During the “Science goes City” event on April 19, 2026, science came to the people in downtown Göttingen, once again transforming the area into a lively science promenade. The GWDG also took part again, this time with an interesting presentation that shed light on the invisible paths of internet data, and was delighted to welcome numerous visitors and engage in fascinating conversations at its booth.



1_Weltkugel, auf der die aktuellen, öffentlich bekannten Unterseekabel eingezeichnet sind. Ein anschauliches Bild um den Besucher*innen zu zeigen, wie vernetzt der Planet bereits ist.



2_Weltkugel mit der Datenroute von Göttingen über die Unterseekabel SeaMeWe-6 und Southeast Asia–Japan Cable (SJC) nach Tokio zur dort gehosteten Website JAPAN STUDY SUPPORT (<https://jps.jp>), einem Informationsportal für internationale Studierende in Japan.

Aufruf auf einem Server in San Francisco landete, sowie eine Website in Japan, bei deren Aufruf man schließlich auf einem Großrechner in Tokio landete, wie in Abbildung 2 zu sehen ist. So konnten wir den Besucher*innen die komplexe internationale Dimension der Datenübertragung anschaulich zeigen.

Die meisten Besucher*innen waren überrascht, dass eine Anfrage an eine Website nicht direkt auf dem dafür vorgesehenen Server landet, sondern erst einmal eine kleine Weltreise über viele

Server antritt, bevor sie ihr Ziel erreicht. Besonders gefreut haben sich die Besucher*innen, wenn eine Website, die sie ausgewählt hatten, erfolgreich angepingt wurde. Auch die Visualisierung der Unterseekabel löste oftmals Staunen und Diskussion aus. So konnten wir anregende Gespräche über Infrastruktur und natürlich auch über Datenschutz führen und den Besucher*innen detailliert Informationen dazu geben. ●

Gemeinsamer Zukunftstag bei der GWDG und beim IT-Service-Center der UMG

Text und Kontakt:

Sascha Safenreider
sascha.safenreider@gwdg.de

Carolina Ranfla Jimenez
carolina.ranflajimenez@gwdg.de

Am 23. April 2026 nahmen rund 25 Schüler*innen am Zukunftstag bei der GWDG und beim IT-Service-Center der der Universitätsmedizin Göttingen (UMG) teil. In einem abwechslungsreichen Programm erhielten sie Einblicke in medizinische und informationstechnologische Berufsfelder, von der Besichtigung einer Biobank bis hin zu interaktiven Anwendungen im Bereich der Künstlichen Intelligenz. Ein Tag voller neuer Eindrücke, Neugier und möglicher Zukunftsperspektiven.

Der Zukunftstag ist ein besonderer Tag in Deutschland, an dem Schüler*innen die Möglichkeit bekommen, für einen Tag in verschiedene Berufe hineinzuschauen. Statt in die Schule zu gehen, besuchen sie Unternehmen, Universitäten oder andere Einrichtungen und erleben dort, wie der Arbeitsalltag wirklich aussieht. Ziel ist es, Horizonte zu erweitern, Interessen zu entdecken und erste Berührungspunkte mit der Arbeitswelt zu schaffen.

ABWECHSLUNGSREICHES PROGRAMM

Auch in diesem Jahr beteiligten sich die GWDG und das IT-Service-Center der Universitätsmedizin Göttingen (UMG) wieder mit einem gemeinsamen Programm an diesen besonderen Tag. Rund 25 Kinder und Jugendliche nahmen teil und erlebten einen abwechslungsreichen, informativen und inspirierenden Tag.

Der Tag begann um 9:00 Uhr am Haupteingang der UMG. Nach einer kurzen Begrüßung erhielten die Teilnehmer*innen erste Einblicke in die medizinische Welt. Ein besonderes Highlight war die Besichtigung der Biobank, bei der die Schüler*innen erfuhren, wie Proben für die Forschung gesammelt, gelagert und genutzt werden. Im Anschluss folgte ein Rundgang über das Gelände der UMG. Dabei konnten die Teilnehmenden die Größe und Vielfalt der Einrichtung kennenlernen und ein Gefühl dafür entwickeln, wie viele unterschiedliche Berufsgruppen im medizinischen Umfeld zusammenarbeiten. Später wurden verschiedene Ausbildungs- und Studienmöglichkeiten vorgestellt. Die Schüler*innen erhielten einen Überblick über berufliche Wege im Gesundheitswesen und konnten Fragen zu Voraussetzungen, Inhalten und Perspektiven stellen. Beim gemeinsamen Mittagessen wurde der Austausch in entspannter Atmosphäre fortgesetzt, bevor es gemeinsam mit dem Stadtbus weiter zur GWDG ging.

Am frühen Nachmittag wurden die Schüler*innen bei der GWDG im Göttinger Rechenzentrum begrüßt. Nach einer kurzen Einführung wurden sie in zwei Gruppen aufgeteilt, um unterschiedliche Programmpunkte parallel zu erleben. Eine Gruppe begann mit einer Führung durch das Rechenzentrum. Dabei konnten die

Teilnehmer*innen nicht nur theoretisch, sondern auch ganz praktisch verstehen, wie moderne IT-Infrastruktur funktioniert. Besonders beeindruckend war es, die Server aus nächster Nähe zu sehen und zu erfahren, wie diese gekühlt werden und warum eine stabile Stromversorgung so essenziell ist. Außerdem wurden Themen wie die Sicherheit im Rechenzentrum anschaulich erläutert, da die Schüler*innen zahlreiche Fragen zu den Sicherheitsmaßnahmen hatten. Zudem zeigten sie großes Interesse daran, wie das Rechenzentrum eigentlich funktioniert.

Die zweite Gruppe nahm an einer interaktiven Präsentation zur GWDG-Ausbildung und zum Thema Künstliche Intelligenz teil. Ein besonderes Highlight war dabei „Neura“, eine interaktive LED-Wand, die ein künstliches neuronales Netzwerk visualisiert (siehe Abbildung 1). Die Schüler*innen hatten die Möglichkeit, selbst aktiv zu werden: Auf Tablets zeichneten sie einfache Bilder, die anschließend von Neuras „Gehirn“ erkannt werden sollten. Über ein visuelles Feedback auf der LED-Wand konnten sie verfolgen,

Joint Future Day at the GWDG and the UMG IT Service Center

On April 23, 2026, around 25 students participated in the Future Day, visiting the GWDG and the IT Service Center of the University Medical Center Göttingen (UMG). The participants spent a full day exploring both medical and information technology environments, gaining hands-on insights into real world applications and career paths. The program included a visit to the UMG Biobank, an introduction to medical professions, and an interactive afternoon at the GWDG data center. There, students experienced modern IT infrastructure and engaged with artificial intelligence through a hands-on learning activity using a neural network visualization tool. The event aimed not only to provide knowledge but also to inspire curiosity, encourage exploration, and support young people in discovering their individual interests and future aspirations.

wie die KI lernte. Durch ihr eigenes Feedback entschieden sie, ob die Erkennung korrekt war, und trugen so aktiv zum Training des Systems bei. Viele der Schüler*innen kannten KI bereits aus Anwendungen wie Chatbots oder Suchmaschinen. Doch nun konnten sie erstmals erleben, wie ein neuronales Netzwerk tatsächlich funktioniert.

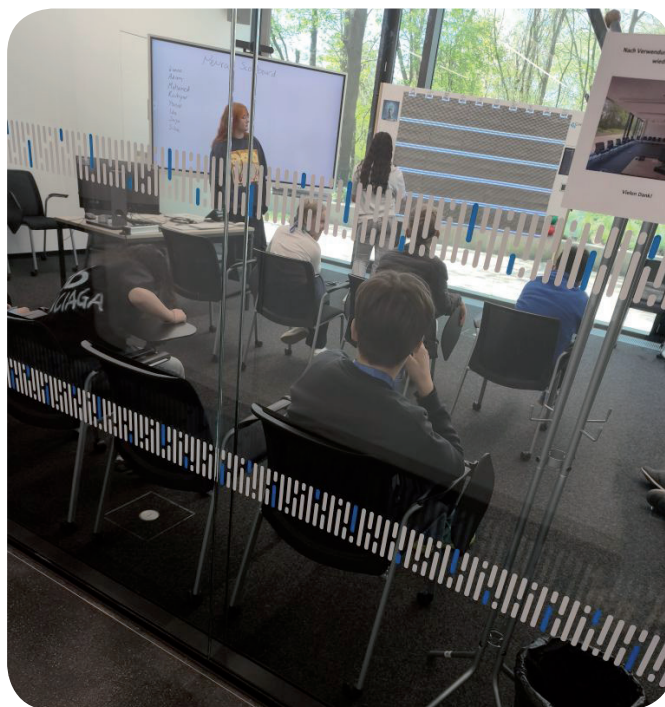
Nach dem ersten Durchlauf wechselten die Gruppen, sodass alle Teilnehmer*innen sowohl die Führung durch das Rechenzentrum als auch die KI-Session erleben konnten. Der Nachmittag war geprägt von Neugier, Lachen und vielen neuen Eindrücken. Am Ende des Tages war deutlich zu spüren, dass dieser Zukunftstag mehr war als nur ein Einblick in verschiedene Berufe. Für einige Schüler*innen hat sich vielleicht eine neue Vorstellung von ihrer Zukunft entwickelt. Andere haben möglicherweise festgestellt, dass bestimmte Bereiche nicht ihren Interessen entsprechen und genau das ist ebenso wertvoll. Nicht jede Erfahrung muss sofort zu einer klaren Entscheidung führen. Gerade in jungen Jahren ist es wichtig, verschiedene Wege kennenzulernen, auszuprobieren und eigene Interessen zu entdecken.

Der Zukunftstag bietet genau diesen Raum – einen geschützten Rahmen, in dem Fragen gestellt, Dinge ausprobiert und Eindrücke gesammelt werden können. Vielleicht hat dieser Tag bei einigen den Wunsch geweckt, später einmal im medizinischen oder technischen Bereich zu arbeiten. Vielleicht hat er aber auch einfach Mut gemacht, weiter zu entdecken.

FAZIT: INTERESSANTE EINBLICKE IN EINE VIELFÄLTIGE ARBEITSWELT

Veranstaltungen wie der Zukunftstag zeigen, wie wichtig es ist, frühzeitig Einblicke in unterschiedliche Berufsfelder zu ermöglichen. Sie schaffen Verbindungen zwischen Theorie und Praxis und geben jungen Menschen Orientierung in einer Welt voller Möglichkeiten. Denn vielleicht war dieser Tag für einige der erste Schritt in eine Richtung, die sie eines Tages zurück an genau diesen Ort führen könnte.

Und vielleicht hat dieser Tag bei dem einen oder der anderen den Wunsch geweckt, eines Tages selbst Teil dieses Arbeitsumfelds zu werden. ■



1_KI an der interaktiven LED-Wand „Neura“ erleben



2_Präsentation von Hardware



Filmrechercheplattform geht online – Projekt „AVefi“ vernetzt Filmbestände für Archive, Museen und Forschung

Text:
GWDG-Pressinformation
2/2026 vom 19.05.2026

AVefi, eine neue von der GWDG mitentwickelte Rechercheplattform, die Filmbestände institutionsübergreifend vernetzt und auffindbar macht, geht jetzt online. Die interessierte Öffentlichkeit bekommt hier die Möglichkeit, Sammlungen verschiedener Archive anhand von thematischen wie auch technischen Kriterien zu durchsuchen. Neben den klassischen Formaten sollen insbesondere auch Lehr-, Gebrauchs- und Amateurfilme auffindbar und zitierbar gemacht werden.

Mit AVefi geht eine neue Rechercheplattform online, die Filmbestände institutionsübergreifend vernetzt und auffindbar macht. Über ein rein werkorientiertes Nachweissystem hinausgehend, vernetzt AVefi (Automatisiertes Verbundsystem für audiovisuelle Bestände über einheitliche Filmidentifikatoren) unterschiedliche audiovisuelle Formate – von Spiel- und Dokumentarfilm bis hin zu Amateur- und Gebrauchsfilmen – und Filmdateien aus Archiven und Sammlungen in einer zentralen, offenen Datenbank und verknüpft sie über einheitliche Filmidentifikatoren.



Die Plattform ermöglicht es, Filmwerke und ihre vorhandenen Materialien über Archivgrenzen eindeutig aufzufinden und verlässlich zu zitieren. Sie hilft beispielsweise bei Fragen wie: Welche Exemplare zu Werken von Helma Sanders-Brahms oder Hermann Schlenker haben verschiedene Einrichtungen? Wo finde ich eine 35-mm-Kopie von „Die Mörder sind unter uns“? Welche dokumentarischen Produktionen zu den Themen „Reisen“ oder „Arbeit“ gibt es? Und wo sind sie zugänglich?

AVefi stellt dafür Daten und Verlinkungen bereit. Die offene Struktur ermöglicht perspektivisch auch eine internationale Erweiterung des Netzwerks.

FILME INSTITUTIONSÜBERGREIFEND RECHERCHIEREN

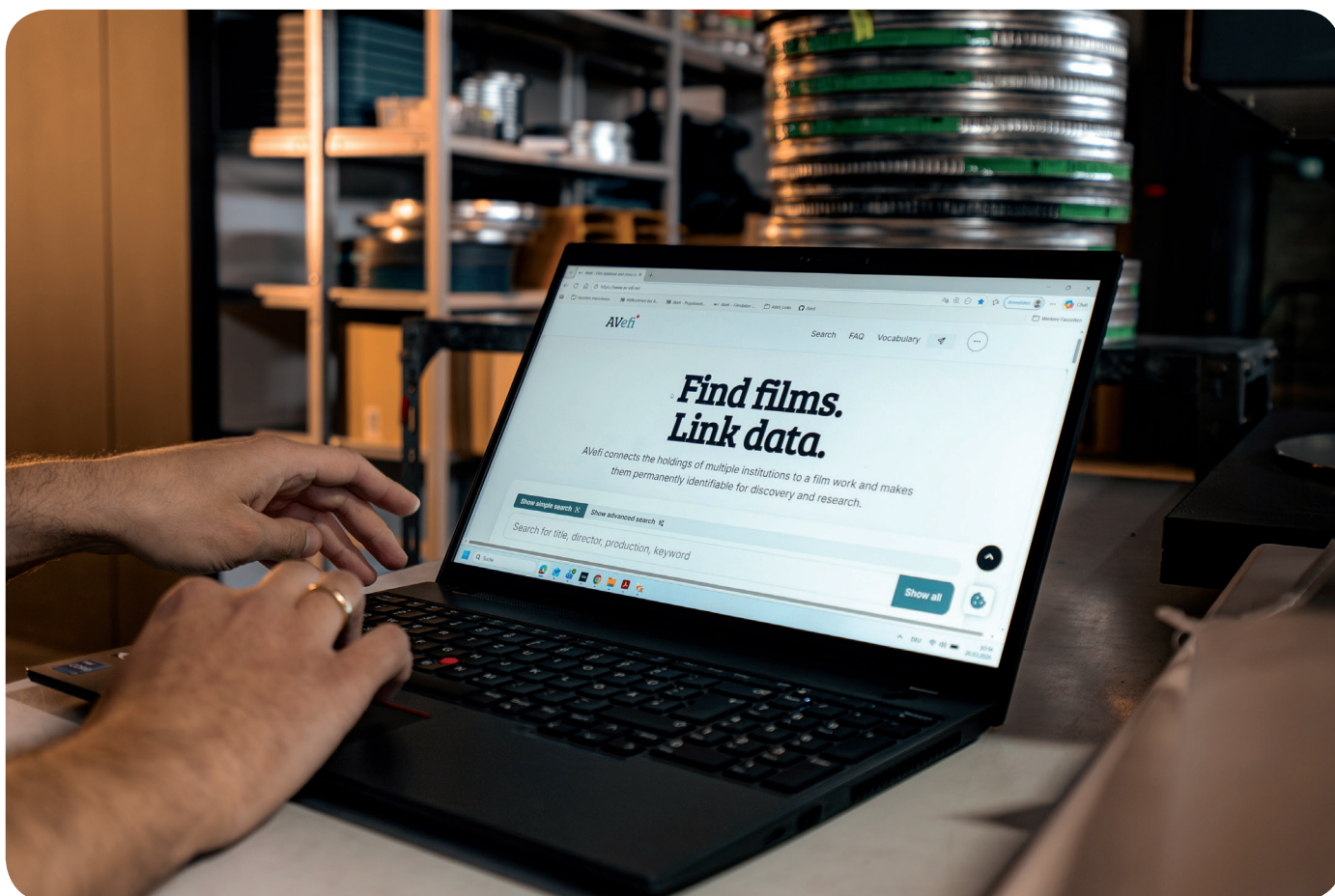
Filme sind heute in zahlreichen Archiven und Sammlungen überliefert, oft mit unterschiedlichen Datensätzen und Erschließungsständen. AVefi ermöglicht eine übergreifende Recherche nach Titeln, Themen, Orten und beteiligten Personen – über einzelne Archive hinaus. Produktionsdaten, Material- und Formatanlagen werden gebündelt dargestellt. Dabei entscheiden die jeweiligen Einrichtungen selbst, welche Bestände sie einbringen. Zu jedem Filmwerk zeigt die Datenbank, in welchen Archiven oder Sammlungen Exemplare vorhanden sind. So werden sowohl gut dokumentierte als auch bislang wenig erschlossene Bestände sichtbar und besser auffindbar.

EINHEITLICHE FILMIDENTIFIKATOREN ALS GRUNDLAGE

AVefi steht für „Automatisiertes Verbundsystem für audio-

Film Research Platform “AVefi” Goes Live

AVefi, a new research platform co-developed by the GWDG, that links film collections across institutions and makes them discoverable, is now going live. The interested public will have the opportunity to search the collections of various archives by subject matter as well as technical criteria. In addition to traditional formats, the aim is in particular to make educational, functional and amateur films searchable and citable.



Mit AVefi können Filme und vorhandene Materialien über Archivgrenzen hinweg gefunden und verlässlich zitiert werden. (Foto: Odbileg Nyamsuren/Deutsche Kinemathek)

visuelle Bestände über einheitliche Filmidentifikatoren“. Im Zentrum der Plattform steht die Vergabe sogenannter einheitlicher Filmidentifikatoren (efi). Jedes Filmwerk, jede Fassung und jedes einzelne Exemplar erhalten eine klare, dauerhafte Kennung. Diese persistenten Identifikatoren ermöglichen es, Datensätze verschiedener Einrichtungen miteinander zu verknüpfen, abgleichbar zu machen und langfristig stabil referenzierbar zu halten. Damit schafft AVefi eine verlässliche Grundlage für Forschung, Lehre und die nachhaltige Nutzung von Metadaten.

BREITES SPEKTRUM DER FILMKULTUR

AVefi bildet ein breites Spektrum filmischer Überlieferung ab – vom frühen Kino- über den Dokumentarfilm bis hin zu Gebrauchsfilm. Ein besonderer Fokus liegt derzeit auf Unterrichts-, Lehr- und Wissenschaftsfilm sowie auf Amateurproduktionen und auf bislang wenig erschlossenen Beständen. Ziel ist es, gerade kleinere Sammlungen in einem offenen und niedrigschwelligem System sichtbar zu machen. In der laufenden zweiten Projektphase wird die Plattform zudem um filmbegleitende Materialien mit Werkbezug wie Fotografien, Drehbücher oder Produktionsunterlagen erweitert.

GEMEINSCHAFTSPROJEKT MIT STARKEN PARTNERN

Die Infrastruktur wird seit 2023 in einem von der Deutschen Forschungsgemeinschaft geförderten Projekt entwickelt, das sich derzeit in der zweiten Phase befindet. Beteiligt sind als Verbundpartner die Stiftung Deutsche Kinemathek – Museum für Film und Fernsehen (SDK), die TIB – Leibniz-Informationszentrum Technik und Naturwissenschaften und Universitätsbibliothek, die Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG), das Filmmuseum der Landeshauptstadt Düsseldorf (FMD) sowie das Marburg Center for Digital Culture and Infrastructure (MCDI). Zahlreiche Film- und Medienarchive mit audiovisuellen Beständen unterstützen das Projekt als Praxispartner.

„Mit AVefi entsteht eine Infrastruktur, die das Filmerbe stärker vernetzt“, betont Dr. Sven Bingert (GWDG). „Die Plattform lebt vom Mitmachen. Wir verstehen AVefi als dynamisches Gemeinschaftsprojekt“, so Antje Dittmann (SDK) und Matti Stöhr (TIB) im Namen der Projektbeteiligten.

Die zum Public Release verfügbaren Bestände werden kontinuierlich erweitert. AVefi ist als offenes Verbundsystem angelegt – weitere Datenquellen und Kooperationspartner sind ausdrücklich eingeladen, sich zu beteiligen. Interessierte Institutionen und Sammlungen sind eingeladen, mit dem AVefi-Projektteam Kontakt über <https://projects.tib.eu/av-efi/kontakt/> aufzunehmen. ●

NEUER MITARBEITER DR. JAN CILLIÉ LOUW

Seit dem 1. Februar 2026 ist Herr Dr. Jan Cillié Louw als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe „Computing“ (AG C) beschäftigt. Er hat im Juli 2023 seine Doktorarbeit zu stark korrelierten Quantensystemen in der Gruppe von Stefan Kehrein an der Universität Göttingen abgeschlossen. Danach arbeitete er am Max-Planck-Institut für Quantenoptik (MPQ) sowie an der Ludwig-Maximilians-Universität München an der Entwicklung von q-dit-Quantenschaltungen. Seine Aufgaben in der AG C nutzen diese Kompetenzen im Rahmen des QUICS-Projekts (Quantum Innovation and Computing for SMEs), in dem er als Berater und Trainer tätig ist. Derzeit umfasst seine Tätigkeit die Erstellung neuer Schulungen, die Entwicklung von Jupyter-Notebooks für einen QC-Digital-Twin sowie die Kartierung der Quanten-Hardware-Landschaft. Herr Dr. Louw ist per E-Mail unter jancillie.louw@gwdg.de zu erreichen.



Kunkel



NEUER MITARBEITER TOBIAS MÜLLER

Seit dem 1. April 2026 ist Herr Tobias Müller als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe „eScience“ (AG E) beschäftigt und verstärkt dort als Serviceentwickler die Aktivitäten der GWDG im Bereich Open Educational Resources (OERs). Er hat zuvor seinen Masterabschluss in Angewandter Informatik an der Georg-August-Universität Göttingen erworben und sich in seiner Masterarbeit mit dem Schnittpunkt von Data Science und Privatheit beschäftigt. Herr Müller ist telefonisch unter 0551 39-30385 und per E-Mail unter tobias.mueller@gwdg.de zu erreichen.

Wieder

INFORMATIONEN:
support@gwdg.de
0551 39-30000

Juni bis
Dezember 2026

Academy



| KURS | DOZENT*IN | TERMIN | ANMELDEN BIS | FORMAT |
|--|----------------------------|---|--------------|---------|
| GRUNDLAGEN DER BILDBEARBEITUNG MIT PHOTOSHOP | Töpfer | 02.06. – 03.06.2026 9:30 – 15:30 Uhr | 26.05.2026 | Online |
| INTRODUCTION TO NEUROMORPHIC COMPUTING AND THE SPINNAKER HARDWARE PLATFORM | Dr. Luboewski, Dr. Paleico | 02.06.2026 13:00 – 17:00 Uhr | 26.05.2026 | Online |
| 3D-MODELLIERUNG UND 3D-DRUCK – KURS FÜR AUZUBILDENDE | Jendrysseck | 03.06.2026 9:30 – 16:00 Uhr | 27.05.2026 | Präsenz |
| SUPERCOMPUTING FOR EVERY SCIENTIST | Eulert, Lüdemann | 04.06.2026 9:00 – 17:00 Uhr | 28.05.2026 | Online |
| KI IN DER VERWALTUNG: EINE EINFÜHRUNG IN DIE NUTZUNG FÜR ALLE MITARBEITER*INNEN | Eulert | 09.06.2026 9:00 – 12:00 Uhr | 02.06.2026 | Präsenz |
| USING THE GWDG DATA POOLS FOR SCIENTIFIC DATA SHARING | L. Quentin | 15.06.2026 9:00 – 12:00 Uhr | 08.06.2026 | Online |
| DEEP DIVE INTO CONTAINERS | Dr. Nordsiek | 16.06.2026 9:00 – 17:00 Uhr | 09.06.2026 | Online |
| WORKING WITH GRO.DATA | Dr. Király | 16.06.2026 14:00 – 15:30 Uhr | 09.06.2026 | Online |

| KURS | DOZENT*IN | TERMIN | ANMELDEN BIS | FORMAT |
|--|--|--|--------------|---------|
| QUICKSTARTING R: EINE ANWENDUNGSORIENTIERTE EINFÜHRUNG IN DAS STATISTIKPAKET R | Cordes | 17.06. – 18.06.2026 9:00 – 15:30 Uhr | 10.06.2026 | Online |
| FOUNDATIONS OF SQL AND DATA MANAGEMENT | Sharma | 18.06.2026 13:00 – 17:00 Uhr | 11.06.2026 | Online |
| EFFECTIVELY UTILIZE AI TOOLS IN RESEARCH | Eulert, Lewis | 23.06.2026 9:00 – 12:00 Uhr | 16.06.2026 | Präsenz |
| GRUNDLAGEN DER BILDBEARBEITUNG MIT AFFINITY PHOTO – SCHNUPPERKURS FÜR EINSTEIGER*INNEN | Töpfer | 24.06. – 25.06.2026 9:30 – 12:30 Uhr | 17.06.2026 | Online |
| USING THE GWDG SCIENTIFIC COMPUTE CLUSTER – AN INTRODUCTION | Eulert, Dr. Lüdemann | 30.06.2026 9:00 – 17:00 Uhr | 23.06.2026 | Präsenz |
| STATISTIK MIT R FÜR TEILNEHMER*INNEN MIT VOR-KENNTNISSEN – VON DER ANALYSE ZUM BERICHT | Cordes | 01.07. – 02.07.2026 9:00 – 15:30 Uhr | 24.06.2026 | Online |
| DEEP LEARNING BOOTCAMP: BUILDING AND DEPLOYING AI MODELS | Lewis | 01.07. – 02.07.2026 14:30 – 16:30 Uhr | 24.06.2026 | Online |
| GETTING STARTED WITH LLVM – BUILD YOUR OWN COMPILER (NHR-SW AT NHR-NORD) | Meyer | 02.07.2026 | 25.06.2026 | Online |
| GETTING STARTED WITH LINUX BASH | Eulert, Dr. Lüdemann | 13.08.2026 9:00 – 12:00 Uhr | 06.08.2026 | Online |
| GETTING STARTED WITH THE AI TRAINING PLATFORM | Eulert, Dr. Lüdemann | 13.08.2026 13:00 – 17:00 Uhr | 06.08.2026 | Online |
| POSTGRESQL – GRUNDKURS | Groh | 19.08.2026 9:30 – 16:00 Uhr | 12.08.2026 | Präsenz |
| POSTGRESQL FÜR FORTGESCHRITTENE | Groh | 20.08.2026 9:30 – 16:00 Uhr | 13.08.2026 | Präsenz |
| INTRODUCTION TO PANGENOMICS | Dr. Paleico, Hussein | 24.08.2026 13:00 – 17:00 Uhr | 17.08.2026 | Online |
| SECURE HPC – PARALLEL COMPUTING WITH HIGHEST SECURITY | Dieterle, L. Quentin | 25.08.2026 9:00 – 12:00 Uhr | 18.08.2026 | Online |
| DEEP LEARNING WITH GPU CORES | Kirchner, Meisel, Biniaz, Doost Hosseini | 26.08.2026 13:00 – 17:00 Uhr | 19.08.2026 | Online |
| USING JUPYTER NOTEBOOKS ON HPC | Khuziyakhmetov | 27.08.2026 9:00 – 12:00 Uhr | 20.08.2026 | Online |
| SQL – KURS FÜR AUSZUBILDENDE | Groh | 01.09. – 02.09.2026 9:30 – 16:00 Uhr | 25.08.2026 | Präsenz |
| INTRODUCTION TO SHINY – A SIMPLE WAY TO CREATE WEB APPS WITH R AND PYTHON | Dr. Paleico | 02.09.2026 13:00 – 17:00 Uhr | 26.08.2026 | Online |

| KURS | DOZENT*IN | TERMIN | ANMELDEN BIS | FORMAT |
|--|--------------------------------------|--|--------------|---------|
| DATA MANAGEMENT CONCEPTS FOR EFFICIENT AND USER-FRIENDLY HPC | L. Quentin | 03.09.2026 13:00 – 17:00 Uhr | 27.08.2026 | Online |
| INDESIGN GRUNDKURS – SCHWERPUNKT POSTER-GESTALTUNG | Töpfer | 09.09. – 10.09.2026 9:30 – 15:30 Uhr | 02.09.2026 | Online |
| DEEP LEARNING BOOTCAMP: BUILDING AND DEPLOYING AI MODELS | Lewis | 09.09. – 10.09.2026 14:30 – 16:30 Uhr | 02.09.2026 | Online |
| SUPERCOMPUTING FOR EVERY SCIENTIST | Eulert, Lüdemann | 10.09.2026 9:00 – 17:00 Uhr | 03.09.2026 | Online |
| KI IN DER VERWALTUNG: EINE EINFÜHRUNG IN DIE NUTZUNG FÜR ALLE MITARBEITER*INNEN | Eulert, Rafi | 14.09.2026 9:00 – 12:00 Uhr | 07.09.2026 | Präsenz |
| AFFINITY PUBLISHER – SCHNUPPERKURS FÜR EINSTEIGER*INNEN | Töpfer | 22.09.2026 10:30 – 15:30 Uhr | 15.09.2026 | Online |
| INTRODUCTION TO ALPHA-FOLD | Dr. Paleico | 24.09.2026 13:00 – 17:00 Uhr | 17.09.2026 | Online |
| EFFECTIVELY UTILIZE AI TOOLS IN RESEARCH | Eulert, Lewis, Rafi | 28.09.2026 9:00 – 12:00 Uhr | 21.09.2026 | Präsenz |
| USING THE GWDG SCIENTIFIC COMPUTE CLUSTER – AN INTRODUCTION | Eulert, Dr. Lüdemann | 01.10.2026 9:00 – 17:00 Uhr | 24.09.2026 | Präsenz |
| ANSYS ON CLUSTER AND POST-PROCESSING OF SIMULATION RESULTS | Dr. Höhn, Dr. Kanning | 05.10.2026 9:00 – 17:00 Uhr | 28.09.2026 | Online |
| PERFORMANCE ENGINEERING TOOLS FOR AI AND HPC WORKLOADS | Masih, Dr. Lüdemann | 06.10.2026 9:00 – 17:00 Uhr | 29.09.2026 | Online |
| INDESIGN – AUFBAUKURS | Töpfer | 07.10. – 08.10.2026 9:30 – 16:00 Uhr | 30.09.2026 | Online |
| PRACTICAL: HIGH-PERFORMANCE COMPUTING SYSTEM ADMINISTRATION | Prof. Kunkel, Decker | 19.10. – 23.10.2026 9:00 – 17:00 Uhr | 12.10.2026 | Online |
| USING THE GÖDL DATA CATALOG FOR SEMANTIC DATA ACCESS ON THE GWDG HPC SYSTEMS | L. Quentin | 02.11.2026 9:00 – 12:00 Uhr | 26.10.2026 | Online |
| ADMINISTRATION IM ACTIVE DIRECTORY | S. Quentin, Dawidowski | 03.11. – 04.11.2026 9:00 – 15:30 Uhr | 27.10.2026 | Präsenz |
| PROGRAMMANWENDUNG IN AFFINITY PHOTO – SCHNUPPERKURS FÜR EINSTEIGER*INNEN | Töpfer | 05.11.2026 10:00 – 15:30 Uhr | 29.10.2026 | Online |
| QUANTUM COMPUTING WITH SIMULATORS ON HPC | Dr. Boehme, Kayi, Altpeter, Dr. Louw | 08.11.2026 9:00 – 17:00 Uhr | 01.11.2026 | Online |

| KURS | DOZENT*IN | TERMIN | ANMELDEN BIS | FORMAT |
|---|--------------------------------|--|--------------|---------|
| MONITORING HPC SYSTEMS IN THE GWDC | Merz | 09.11.2026 9:00 – 12:00 Uhr | 02.11.2026 | Online |
| VIRTUELLE CLOUD-INFRA-STRUKTUREN – KURS FÜR AUSZUBILDENDE | Kopp | 10.11. – 12.11.2026 9:00 – 16:00 Uhr | 03.11.2026 | Präsenz |
| INTRODUCTION TO QUANTUM COMPUTING FOR LIFE SCIENCES | Dr. Paleico, Dr. Louw, Hussein | 12.11.2026 13:00 – 17:00 Uhr | 05.11.2026 | Online |
| PERFORMANCE ENGINEERING IN ML WORKLOADS USING SCORE-P AND VAMPIR | Masih | 16.11.2026 13:00 – 17:00 Uhr | 09.11.2026 | Online |
| AFFINITY DESIGNER – SCHNUPPERKURS FÜR EINSTEIGER*INNEN | Töpfer | 17.11.2026 10:30 – 15:30 Uhr | 10.11.2026 | Online |
| EINFÜHRUNG IN DIE STATISTISCHE DATENANALYSE MIT SPSS | Cordes | 18.11. – 19.11.2026 9:00 – 15:30 Uhr | 11.11.2026 | Online |
| GETTING STARTED WITH LINUX BASH | Eulert, Dr. Lüdemann | 19.11.2026 9:00 – 12:00 Uhr | 12.11.2026 | Online |
| GETTING STARTED WITH THE AI TRAINING PLATFORM | Eulert, Dr. Lüdemann | 19.11.2026 13:00 – 17:00 Uhr | 12.11.2026 | Online |
| USING THE GWDC DATA POOLS FOR SCIENTIFIC DATA SHARING | L. Quentin | 25.11.2026 9:00 – 12:00 Uhr | 18.11.2026 | Online |
| DEBUGGING SCIENTIFIC APPLICATIONS – ILLUSTRATION ON OPENFOAM | Dr. Höhn | 26.11.2026 13:00 – 17:00 Uhr | 19.11.2026 | Online |
| GRUNDLAGEN DER BILDBEARBEITUNG MIT PHOTOSHOP | Töpfer | 01.12. – 02.12.2026 9:30 – 15:30 Uhr | 24.11.2026 | Online |
| 3D-MODELLIERUNG UND 3D-DRUCK – KURS FÜR AUSZUBILDENDE | Jendrysseck | 02.12.2026 9:30 – 16:00 Uhr | 25.11.2026 | Präsenz |
| SUPERCOMPUTING FOR EVERY SCIENTIST | Eulert, Dr. Lüdemann | 03.12.2026 9:00 – 17:00 Uhr | 26.11.2026 | Online |
| DEEP DIVE INTO CONTAINERS | Dr. Nordsiek | 15.12.2026 9:00 – 17:00 Uhr | 08.12.2026 | Online |
| DEEP LEARNING BOOTCAMP: BUILDING AND DEPLOYING AI MODELS | Lewis | 16.12. – 17.12.2026 14:30 – 16:30 Uhr | 09.12.2026 | Online |

Teilnehmerkreis

Das Angebot der GWDG Academy richtet sich an die Beschäftigten aller Einrichtungen der Universität Göttingen, der Max-Planck-Gesellschaft sowie aus wissenschaftlichen Einrichtungen, die zum erweiterten Kreis der Nutzer*innen der GWDG gehören. Studierende am Göttingen Campus zählen ebenfalls hierzu. Für manche Kurse werden spezielle Kenntnisse vorausgesetzt, die in den jeweiligen Kursbeschreibungen genannt werden.

Anmeldung

Für die Anmeldung zu einem Kurs müssen Sie sich zunächst mit Ihrem Benutzernamen und Passwort in der GWDG Academy (<https://academy.gwdg.de>) einloggen. Wenn Sie zum Kreis der berechtigten Nutzer*innen der GWDG gehören, erhalten Sie anschließend automatisch Zugang zu unserem Kursprogramm. Sollten Sie noch keinen Account besitzen, können Sie sich unter <https://id.academiccloud.de> registrieren und müssen ggf. auf Anfrage für die Anmeldung zu unseren Kursen freigeschaltet werden. Bei Online-Kursen kann das Anmeldeverfahren abweichen. Genauere Informationen dazu finden Sie in der jeweiligen Kursbeschreibung. Einige Online-Angebote stehen Ihnen jederzeit und ohne Anmeldung zur Verfügung.

Absage

Absagen können bis zu sieben Tagen vor Kursbeginn erfolgen. Bei kurzfristigeren Absagen werden allerdings die für den Kurs angesetzten Arbeitseinheiten (AE) vom AE-Kontingent der jeweiligen Einrichtung abgezogen.

Kursorte

Die Kurse finden entweder in einem geeigneten Online-Format oder als Präsenzkurs statt. Nähere Informationen dazu finden Sie bei den jeweiligen Kursen. Auf Wunsch und bei ausreichendem Interesse führen wir auch Kurse vor Ort in einem Institut durch, sofern dort ein geeigneter Raum mit entsprechender Ausstattung zur Verfügung gestellt wird.

Kosten bzw. Gebühren

Die Academy-Kurse sind – wie die meisten anderen Leistungen der GWDG – in das interne Kosten- und Leistungsrechnungssystem der GWDG einbezogen. Die den Kursen zugrundeliegenden AE werden vom AE-Kontingent der jeweiligen Einrichtung abgezogen. Für alle Einrichtungen der Universität Göttingen und der Max-Planck-Gesellschaft sowie die meisten der wissenschaftlichen Einrichtungen, die zum erweiterten Kreis der Nutzer*innen der GWDG gehören, erfolgt keine Abrechnung in EUR. Dies gilt auch für die Studierenden am Göttingen Campus.

Kontakt und Information

Wenn Sie Fragen zum aktuellen Academy-Kursangebot, zur Kursplanung oder Wünsche nach weiteren Kursthemen haben, schicken Sie bitte eine E-Mail an support@gwdg.de. Falls bei einer ausreichend großen Gruppe Interesse besteht, könnten u. U. auch Kurse angeboten werden, die nicht im aktuellen Kursprogramm enthalten sind.

Kurz & knapp

Kursprogramm der GWDG Academy für das zweite Halbjahr 2026

Das Kursprogramm der GWDG Academy für das zweite Halbjahr 2026 wurde jetzt unter <https://academy.gwdg.de> veröffentlicht und ist auch in dieser Ausgabe der GWDG-Nachrichten zu finden. Es enthält einige neue Kurse, insbesondere aus den Bereichen Scientific Computing bzw. High-Performance Computing und Life Sciences. Alle Kurse finden entweder in einem geeigneten Online-Format oder als Präsenzkurs statt. Nähere Informationen dazu finden Sie bei den jeweiligen Kursen.

Otto

Umfrage zum Bedarf der Industrie an Quantencomputing

DWährend Quantencomputing (QC) den Sprung vom Labor-Durchbruch zum industriellen Werkzeug vollzieht, stellt sich für viele Unternehmen nicht mehr die Frage nach dem „Ob“, sondern nach dem „Wie“. Um sicherzustellen, dass die deutsche Wirtschaft und insbesondere kleine und mittlere Unternehmen (KMU) für diesen Wandel gerüstet sind, ruft eine neue strategische Studie – die „Umfrage zum Bedarf der Industrie an Quantencomputing“ – offiziell zur Teilnahme auf.

Diese Initiative wird im Rahmen des Fraunhofer INQUBATOR (<https://www.fraunhofer-inqubator.de>) und der gemeinsamen QUICS-Zentren der GWDG und der LUH (<https://quics.gwdg.de>) durchgeführt. Die gewonnenen Daten fließen direkt in die Entwicklung regionaler Unterstützungsangebote und spezialisierter Beratungsleistungen ein.

Die Umfrage ist prägnant und effizient gestaltet:

- Zeitaufwand: Die Bearbeitung dauert etwa 6–8 Minuten.
- Schwerpunkte: Bewertet werden Motivation, aktueller Wissensstand, Adoptionsstadium sowie spezifische „Problemklassen“, die für Ihr Unternehmen relevant sind.
- Strategischer Input: Sie haben die Möglichkeit zu definieren, welche Leistungen ein QC-Test- und Beratungszentrum bieten sollte, um Ihr Unternehmen effektiv zu unterstützen.
- Anonymität: Die Teilnahme erfolgt unter höchsten Datenschutzstandards – es werden keine identifizierbaren Informationen gespeichert.

Die Teilnahme an der Umfrage ist unter <https://survey.academiccloud.de/ff/596263?lang=de> möglich.

Leuw



Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen