



Physikalische Chemie, Universität Rostock

**Vorlesung: Forschungsdatenmanagement
im Sommersemester 2025**

Dr. habil. Till Biskup

— Glossar zu Vorlesung 12: „Prinzipien“ —

Hinweis: Die nachfolgend genannten Begriffe und Definitionen erheben keinen Anspruch auf formale Korrektheit, sondern dienen lediglich dem besseren Verständnis der in der Vorlesung behandelten Themen und sind im jeweiligen Kontext zu sehen. Mehrfache, voneinander abweichende Definitionen in unterschiedlichen Kontexten sind daher möglich. Fremdsprachige Begriffe werden nach Möglichkeit übersetzt, erscheinen aber ggf. unter ihrem ursprünglichen Namen in der Liste. Verweise auf andere Begriffe innerhalb des Glossars sind durch das vorangestellte Symbol ↑ gekennzeichnet.

Automat von gr. *autómatos* = sich selbst bewegend; Apparat, der einen eingeleiteten technischen Vorgang ohne weiteres menschliches Zutun steuert oder regelt. Vgl. ↑Automatisierung

Automatisierung *automation* Strategie, um sich die manuelle Durchführung repetitiver und meist langweiliger Prozesse zu ersparen, indem sie an Maschinen ausgelagert wird. Automatisierung sorgt für Konsistenz (aber nicht Fehlerfreiheit) und ermöglicht es den menschlichen Akteuren, ihre dadurch freiwerdende Kapazität auf die eigentlichen intellektuellen Aufgaben, die weder automatisiert noch von Algorithmen übernommen werden können, zu verwenden.

dauerhafte Kennung ↑PID

eindeutige Kennung ↑UID

Erkenntnis Aneignung des Sinngehalts von erlebten bzw. erfahrenen Sachverhalten, Zuständen oder Vorgängen, Ergebnis des Vorgangs des Erkennens. Erkenntnis beinhaltet immer eine auf die Erfahrung gestützte Beurteilung und setzt notwendiger Weise ein Subjekt voraus, das erkennt. Neue Erkenntnisse, die von innerer und äußerer Erfahrung unabhängig sind, sind immer Ergebnis einer schöpferischen Phantasie. Bei der Erkenntnis stehen

sich Subjekt und Objekt als Erkennendes und Erkanntes gegenüber. Die Erkenntnis führt zu einem Abbild des Objekts im Subjekt. Die grundsätzliche Unvollständigkeit dieses Abbilds ist die Triebkraft hinter dem Erkenntnisgewinn und letztlich der ↑Wissenschaft. Vgl. [1]; wesentliche Beiträge zur Erkenntnistheorie und ihrer Anwendung auf die Naturwissenschaft kommen von Kant [2, 3].

Forschungsdaten zunächst einmal Daten, die im Zuge wissenschaftlicher Vorhaben im Rahmen von Forschung z.B. durch Digitalisierung, Quellenforschungen, Experimente, Messungen, Erhebungen oder Befragungen entstehen. Forschungsdaten im weiteren Sinn umfassen darüber hinaus (physische) Objekte und Werkzeuge (z.B. Fragebögen, Software und Simulationen). Forschungsdaten können grundsätzlich analog oder digital vorliegen. Sie sind Ausgangspunkt der (empirischen) Wissenschaft.

Forschungsdatenmanagement Umgang mit ↑Forschungsdaten über ihren gesamten Lebenszyklus hinweg mit dem Fokus auf ↑Nachvollziehbarkeit und Nachnutzbarkeit; wird meist auf die digitale Welt bezogen, ist letztlich aber nichts anderes als sauberes wissenschaftliches Arbeiten; notwendige, aber nicht hinreichende Bedingung für den

wissenschaftlichen Erkenntnisgewinn.

hinreichend mathematisches Konzept, das eine Bedingung beschreibt, deren Erfüllung ausreicht, um ein gegebenes Ziel zu erreichen. Vgl. ↑notwendig

Idempotenz Eigenschaft eines Programms, bei mehrfacher Ausführung zum gleichen Zustand des Systems zu führen wie bei einmaliger Ausführung. Wichtig insbesondere für die Fehler-toleranz mehrschrittiger Abläufe, wenn sie in einem Durchlauf nicht erfolgreich waren.

Infrastruktur personelle, sachliche und finanzielle Ausstattung, um ein angestrebtes Ziel zu erreichen.

intellektuelle Beherrschbarkeit *intellectual manageability*, nach Edsger Dijkstra [4] das Hauptziel der Softwaretechnik (*software engineering*) – und letztlich des Projektmanagements. Unterschiedliche Lösungsansätze für ein Problem sind unterschiedlich gut intellektuell beherrschbar. Entsprechend ist die intellektuelle Beherrschbarkeit das zentrale Kriterium für die Entscheidung, welche Lösung für ein Problem bevorzugt wird.

Komplexität Eigenschaft der Realität, dass selbst sehr wenige einfache Regeln in ihrer Kombination ein nichttriviales Verhalten erzeugen können, die den Menschen für gewöhnlich überfordern. ↑Größere Projekte in der Softwareentwicklung sind immer komplex, ein ↑System zur Datenverarbeitung allemal. Eine zentrale Strategie zum Umgang mit Komplexität ist ↑Abstraktion und in der Folge ↑Modularisierung. Nach Fred Brooks [5] lassen sich zwei Arten von Komplexität unterscheiden: ↑vermeidbare Komplexität und ↑unvermeidliche Komplexität.

Kopplung *coupling*, Grad der Verbindung zweier Gegenstände oder Systeme. Systemarchitektur zielt generell auf eine lose Kopplung (*loose coupling*) einzelner Komponenten ab, um die ↑Modularität zu erhalten.

Metadaten wörtlich „Daten über Daten“, Informationen zu den numerischen Daten, notwendige

Voraussetzung für eine sinnvolle Verarbeitung der Daten im Kontext eines ↑Systems zur Datenverarbeitung und für ↑nachvollziehbare Wissenschaft.

Modularisierung Aufteilung der Gesamtaufgabe in kleinere Abschnitte. Die Aufteilung wird so lange fortgesetzt, bis die Lösung für den aktuellen Abschnitt unmittelbar in Form von Quellcode offensichtlich ist. Setzt die Definition von ↑Schnittstellen voraus.

Modularität Eigenschaft eines Systems, aus lauter separaten, durch ↑Schnittstellen miteinander verbundenen Teilen zu bestehen. I.d.R. Folge der ↑Modularisierung und einzig erfolgversprechende Strategie für die ↑intellektuelle Beherrschbarkeit komplexer Systeme.

nachvollziehbare Wissenschaft *reproducible science*, seit der Etablierung rechnergestützter Datenauswertung eigentlich nie mehr erreicht, aber für die Wissenschaft konstituierender Aspekt, dass sich Ergebnisse und Auswertungen unabhängig nachvollziehen lassen, weil alle dazu notwendigen Aspekte vollständig und ausreichend beschrieben wurden (↑Nachvollziehbarkeit). Motivation für die Vorlesung, deren Ziel es ist, die Hörer mit Konzepten vertraut zu machen, die letztlich eine ernstzunehmende nachvollziehbare Wissenschaft ermöglichen. Die ↑Nachvollziehbarkeit geht dabei weit über ↑Replizierbarkeit und ↑Reproduzierbarkeit hinaus.

Nachvollziehbarkeit zentraler Aspekt der ↑Wissenschaft und der wissenschaftlichen Methode, die die intersubjektivität ihrer Aussagen ermöglicht. Setzt in der Regel eine ↑hinreichende Beschreibung (und Dokumentation) der einzelnen Schritte voraus, die von einem gegebenen Ausgangspunkt zu einem (neuen) Ergebnis oder auch einer Erkenntnis kommt.

notwendig mathematisches Konzept, das eine Bedingung beschreibt, die zwar erfüllt sein muss, um ein bestimmtes Ergebnis zu bekommen, aber für die Erfüllung nicht ausreicht. Vgl. ↑hinreichend

numerische Stabilität Eigenschaft eines Algorithmus, unempfindlich gegenüber kleinen Stö-

rungen zu sein. Insbesondere Rundungsfehler durch die endliche numerische Genauigkeit wirken sich nicht zu stark auf die Berechnung aus.

Persistenz Fähigkeit, Daten oder logische Verbindungen über lange Zeit (insbesondere über einen Programmabbruch hinaus) bereitzuhalten; benötigt ein nichtflüchtiges Speichermedium.

PID *persistent identifier*, dt. dauerhafte Kennung, i.d.R. eindeutige und langzeitstabile Kennung für physische oder digitale Objekte. Bekannte und weit verbreitete PIDs sind z.B. der *digital object identifier* (DOI), aber auch die *International Standard Book Number* (ISBN) oder die *Open Researcher and Contributor ID* (ORCID). Vgl. ↑Persistenz

Replizierbarkeit *replicability*, unabhängige Wiederholbarkeit der (Roh-)Datenerhebung, meist in Form von Experimenten und Beobachtungen, entsprechend nicht in jedem Fall durchführbar. Vgl. ↑Reproduzierbarkeit, ↑Robustheit, ↑Verallgemeinerbarkeit.

Reproduzierbarkeit *reproducibility*, vollständige Wiederholbarkeit einer beschriebenen Datenverarbeitung und -Analyse. Ausgangspunkt sind existierende Daten, entsprechend sollte sie in jedem Fall möglich sein. Vgl. ↑Replizierbarkeit.

Robustheit *robustness*, im Kontext der Datenverarbeitung die Tatsache, dass unterschiedliche, unabhängige Analysen derselben Daten zum gleichen Ergebnis führen. Vgl. ↑Reproduzierbarkeit, ↑Replizierbarkeit, ↑Verallgemeinerbarkeit

Schnittstelle der Teil eines Systems, der der Kommunikation und dem Austausch z.B. von Information dient. Systeme werden von außen als abgeschlossen (*black box*) betrachtet und kommunizieren ausschließlich über ihre Schnittstelle(n). Die explizite Definition, Dokumentation und Implementation von Schnittstellen sind wesentliche Voraussetzungen für ↑modulare ↑Systemarchitekturen. Schnittstellen ermöglichen die ↑Trennung der

Belange. Oft genug stimmen Schnittstellen in Systemen mit Organisationsgrenzen beteiligter Gruppen überein [6]. In jedem Fall ist es essentiell, mit Systemen nur über deren Schnittstellen zu kommunizieren und *keine* Annahmen über die innere Organisation dieser Systeme zu treffen.

System zur Datenverarbeitung hier: Gesamtsystem für wissenschaftliche Datenverarbeitung von der Datenaufnahme bis zur fertigen Publikation, das alle Aspekte umfasst und das ↑nachvollziehbare Wissenschaft möglich macht und gewährleistet. Definitiv ein größeres Projekt, das nicht nur eine ↑monolithische Anwendung umfasst, sondern viele Aspekte darüber hinaus. Setzt entsprechende ↑Infrastruktur und in der Umsetzung der einzelnen Komponenten sauberen Code und eine solide Softwarearchitektur voraus.

Trennung der Belange *separation of concerns*, grundlegendes Prinzip für ↑Modularisierung, nach Edsger Dijkstra [7] die einzig effektive Möglichkeit, seine Gedanken zu ordnen, indem man sich auf einen Aspekt eines ↑komplexen Problems fokussiert, ohne dabei zu vergessen, dass es lediglich ein Teilaspekt ist.

UID *unique identifier*, dt. eindeutige Kennung, (in einem gegebenen Kontext) eindeutiger Verweis auf eine beliebige Ressource. Vgl. ↑PID

Unix-Philosophie griffige Formulierung dreier Prinzipien, die wesentlich zum Erfolg des Unix-Betriebssystems und seiner Nachfolger (u.a. Linux, macOS, Android, iOS) beigetragen haben. Nach Salus [8, S. 53]: „Write programs that do one thing and do it well. Write programs to work together. Write programs that handle text streams, because that is a universal interface.“ Diese Prinzipien gelten für die Softwareentwicklung genauso wie für die analoge Welt: ↑Modularität und Interoperabilität sind entscheidend, um mit begrenzten Ressourcen Werkzeuge zu entwickeln und zu etablieren.

unvermeidliche Komplexität *essential complexity*, nach Fred Brooks [5] jener Teil der ↑Kom-

plexität eines Systems, der in der Komplexität der Fragestellung begründet ist und der sich nicht verkleinern lässt. Eine gute Systemarchitektur zielt auf die Beherrschung dieser unvermeidlichen Komplexität u.a. durch Einsatz von ↑Abstraktion und ↑Modularisierung. Vgl. ↑vermeidbare Komplexität.

Verallgemeinerbarkeit auch: Generalisierbarkeit, *generalisability*, im Kontext der Datenverarbeitung die Tatsache, dass sowohl unabhängig erhobene Daten als auch voneinander unabhängige Analysemethoden zum gleichen Ergebnis führen. Baustein zur unabhängigen Bestätigung wissenschaftlicher Hypothesen. Vgl. ↑Reproduzierbarkeit, ↑Replizierbarkeit, ↑Robustheit

vermeidbare Komplexität *accidental complexity*, nach Fred Brooks [5] jener Teil der ↑Komplexität eines Systems, der *nicht* in der Komplexität der Fragestellung begründet ist und der sich durch geschickten Einsatz von (etablier-

ten) Strategien beheben lässt. Ein wesentlicher Baustein zur Verringerung dieser vermeidbaren Komplexität ist die Verwendung guter ↑Abstraktionen. Vgl. ↑unvermeidbare Komplexität.

Wissenschaft Auf den Erkenntnisgewinn ausgerichtetes, systematisches menschliches Unterfangen, das in der Regel eine Reihe von Kriterien erfüllt bzw. erfüllen sollte: Unabhängigkeit vom Beobachtenden bzw. Durchführenden, gegründet auf den Erkenntnissen früherer Generationen, sowie überprüfbar, nachvollziehbar und ggf. reproduzierbar. Für Einführungen vgl. u.a. [9, 10].

Zerbrechlichkeit *fragility* In der Softwareentwicklung durch zu enge ↑Kopplung von Modulen entstehendes Phänomen, dass Änderungen an einer Stelle zu Problemen an ganz anderen, davon konzeptionell unabhängigen Stellen führt [11, S. 88].

Literatur

- [1] Heinrich Schmidt. *Philosophisches Wörterbuch*. 22. Aufl. Neu bearbeitet von Georgi Schischkoff. Stuttgart: Kröner, 1991.
- [2] Immanuel Kant. *Kritik der reinen Vernunft*. Herausgegeben von Wilhelm Weischedel. Frankfurt am Main: Suhrkamp, 1974.
- [3] Immanuel Kant. *Metaphysische Anfangsgründe der Naturwissenschaft*. Mit einer Einleitung herausgegeben von Konstantin Pollok. Hamburg: Felix Meiner Verlag, 1997.
- [4] Edsger W. Dijkstra. The humble programmer. *Communications of the ACM* 15 (1972), S. 859–865.
- [5] Frederick P. Brooks. *The Mythical Man Month*. Anniversary edition with four new chapters. Boston: Addison Wesley Longman, 1995.
- [6] Melvin E. Conway. How do committees invent? *Datamation* 14.4 (1968), S. 28–31.
- [7] Edsger W. Dijkstra. „On the Role of Scientific Thought (EWD447)“. In: *Selected Writings on Computing: A Personal Perspective*. New York: Springer-Verlag, 1982, S. 60–66.
- [8] Peter H. Salus. *A Quarter Century of UNIX*. Reading, MA: Addison-Wesley, 1994.
- [9] Alan F. Chalmers. *What is this thing called Science?* Third edition. Berkshire, UK: Open University Press, 1999.
- [10] Hans Poser. *Wissenschaftstheorie*. Stuttgart: Reclam, 2001.
- [11] Robert C. Martin. *Agile Software Development. Principles, Patterns, and Practices*. Upper Saddle River: Prentice Hall, 2003.