

# Informationstechnologie: Grundlagen

---

Michael Sonntag

## Inhaltsübersicht

I.	Einleitung.....	2
1.	Was ist Informatik.....	3
2.	Daten vs Information.....	6
II.	Prinzipielles über den Computer.....	7
1.	Der von-Neumann-Rechner.....	8
2.	Äquivalenz von Hard- und Software.....	9
III.	Hardware.....	10
1.	Hauptgruppen von IKT-Hardware.....	10
2.	Betriebssicherheit, Wartung.....	11
3.	Mobile Geräte.....	12
4.	Eingebettete Systeme („Embedded systems“ ).....	12
IV.	Software.....	13
1.	Typologien für Software.....	14
2.	Die Entwicklung von Software.....	20
3.	Proprietäre und Open Source Software.....	25
4.	Digital Rights Management (DRM).....	27
V.	Netzwerke.....	27
1.	Kommunikation.....	28
2.	Sicherheit.....	29
3.	Internet.....	32
VI.	Geschäftsmodelle.....	44
VII.	Rechts- und sonstige Regeln.....	45

## Literaturauswahl

Eine Einführung bieten *Kersken*, IT-Handbuch für Fachinformatiker<sup>5</sup> (2011) und *Levi/Rembold*, Einführung in die Informatik<sup>4</sup> (2003) sowie *Herold/Lurz/Wohlrab*, Grundlagen der Informatik (2006).

Den Aufbau von Computern auf niedriger Ebene erläutert *Tanenbaum/Austin*, Rechnerarchitektur<sup>6</sup> (2014); Umfassend zur Software-Technik informiert *Balzert*, Lehrbuch der Software-Technik<sup>2</sup>, in mehreren Bänden (Softwaremanagement [2008], Basiskonzepte und Requirements Engineering [2009], Entwurf, Implementierung und Betrieb [2011]); Als Klassiker der Software-Entwicklung können herangezogen werden *Balzert*, Die Entwicklung von Software Systemen: Prinzipien, Methoden, Sprachen, Werkzeuge (1992); Details zu Netzwerken können in *Kurose/Ross*, Computer Networking. A Top-Down Approach<sup>6</sup>

(2012) gefunden werden, während über Betriebssysteme *Tanenbaum/Bos*, Moderne Betriebssysteme<sup>4</sup> (2016) Auskunft gibt.

Ein allgemeines und leicht verständliches Nachschlagewerk, gerade im IT-Bereich (sehr umfassend, wenn auch nicht immer vollständig oder ganz richtig) bietet Wikipedia: <de.wikipedia.org> sowie <en.wikipedia.org> (deutsche und englische Version sind unabhängig voneinander und nur teilweise eine Übersetzung).

## I. Einleitung

- 1/1** Ziel dieses Beitrages ist es, mit Informations- und Kommunikationstechnologie (im Folgenden: IKT) nicht vertrauten Juristen ausgewählte technische Themen dieses Bereiches so weit näher zu bringen, dass sie die jeweils relevanten **technischen** und **organisatorischen Tatsachen** bei der für die juristische Arbeit charakteristischen Bestimmung von **juristischen Sachverhalten** verstehen und angemessen zu berücksichtigen vermögen. Es geht also weder um eine Darlegung von **Informatik-** noch von **Benutzerwissen**: Dem Informatiker wird das Dargebotene lückenhaft und oft allzu vereinfachend erscheinen, und wer Rat zu Benutzerproblemen sucht, wird in anderen Quellen Hilfreicheres finden. Es soll vielmehr dazu dienen, die Kommunikation zwischen Technikern und Juristen durch grundlegendes Verständnis und einen Einblick in die Denkweise zu verbessern. Am ehesten wird das zu Vermittelnde als „**Beurteilungswissen von Sachverhalten mit IKT-Bezug**“ zu bezeichnen sein. Denn wer zB in einem konkreten Fall den (juristischen) Sachverhalt einer mangelhaften Leistung in Form von Qualitätsmängeln einer gelieferten Software, welche (technisch betrachtet) durch Programmierfehler und fehlerhaften Entwurf verursacht wurden, beurteilen soll, der benötigt ein **zumindest prinzipielles Verständnis** dafür, welche Rolle einerseits Design und andererseits Qualitätssicherung beim Herstellungsprozess von Software spielen, wie dieser grob abläuft, und welche Personen hierbei involviert sind – bzw sein sollten. Eine allfällige Zuschreibung von Unterlassungen der am Herstellungsprozess Beteiligten wird damit je nach dem ausgewählten Modell der Software-Entwicklung sehr unterschiedlich ausfallen müssen. Es geht im Folgenden daher darum, ein prinzipielles Verständnis für ausgewählte Themen aus dem Gebiet der IKT zu vermitteln.
- 1/2** Um ermüdende Wiederholungen von zwei Phrasen zu vermeiden, werden diese hier nur einmal an den Beginn gestellt (und müssen in der Folge immer implizit mitgelesen werden):
- 1/3** • „**Vereinfachend gesprochen, verhält es sich so ...**“  
Die durchgehend **stark vereinfachende Darstellungsweise** soll der soeben erklärten Zielsetzung dienen. Das Dargebotene wird dem Kenner

der Materie ohnedies über das hier erforderliche Detail hinaus bekannt sein und kann von ihm daher auch ohne Verlust überblättert werden, während dem mit IKT Unvertrauten mit einer zu detaillierten Einführung (kurzfristig) nicht gedient wäre. Darüber hinaus bleiben Sonderfälle, Ausnahmen, Umgehungsmöglichkeiten etc unerwähnt. Es werden lediglich die wichtigsten Elemente der Grundlagen erörtert, dh der „typische“, „übliche“ oder „häufigste“ Fall.

- „Für eine korrekte rechtliche Bewertung kann relevant sein, dass ...“ 1/4  
Da die dargelegten Gegenstände nach dem Kriterium ihrer möglichen Relevanz für die juristische Beschäftigung mit IKT-Sachverhalten ausgewählt wurden, ist auch dieser Satz immer mitzulesen.

## 1. Was ist Informatik

Informatik ist die **Wissenschaft der systematischen Verarbeitung von Informationen**. Sie wird vielfach jedoch mit der automatisierten Informationsverarbeitung mittels Universalcomputern gleichgesetzt. Es bestehen starke Verbindungen zur Mathematik, Elektrotechnik und Nachrichtentechnik, sowie dem jeweiligen Fachgebiet, in welchem die IKT eingesetzt werden soll. In Bezug auf die Herstellung von Computern spielt die Halbleiterphysik eine essentielle Rolle. Die (Kern-)Informatik teilt sich entsprechend in drei Hauptgebiete: Die **theoretische Informatik** (Berechenbarkeit, formale Sprachen...; was ist wie überhaupt möglich, abstrakte Strukturierung), die **praktische Informatik** (Programmierung im weitesten Sinne, inkl Algorithmen und Datenstrukturen; Bau konkreter Software) sowie die **technische Informatik** (Rechnerarchitektur, Mikroprozessortechnik; Bau der Hardware). Die **angewandte Informatik** beschäftigt sich damit, die Methoden der obigen Teile in anderen Gebieten anzuwenden (Rechtsinformatik, Wirtschaftsinformatik, Bioinformatik etc – auch „Bindestrich-Informatiken“ genannt). Wichtige Prinzipien der Informatik in allen Gebieten sind:

- **Determinismus:** Ist der Rechner in einem bestimmten Zustand und erfolgen bestimmte Eingaben, so ist das Ergebnis in allen Wiederholungen immer genau identisch und das Vorausgeplante. Dies beruht darauf, dass der Computer das, und nur das, macht, „was man ihm anschafft“. Diese exakte **Vorherbestimmung des Ergebnisses** ist in der Praxis dadurch stark eingeschränkt, dass die Eingabe oft eben nicht exakt bekannt ist, sondern nur ungefähr beschrieben werden kann (ein Textverarbeitungsprogramm soll beliebige Briefe verarbeiten, nicht nur einen einzigen vorbestimmten), sowie dass „sonstige“ Einflüsse vorliegen (zB ankomen-

de Daten aus der Umwelt, wie etwa dem Internet) bzw die Eingabe nicht exakt wiederholbar ist (ein Mensch ist nicht in der Lage, auf die Nanosekunde genau eine Taste zu drücken). Entsprechend lassen sich Fehler in Programmen (grob) klassifizieren: Richtige Befehle in falscher Anwendung (es wird ein anderes Problem gelöst, als das tatsächlich vorliegende, falsche Eingabe; der Sachbearbeiter bearbeitet einen Antrag mit dem falschen Gesetz), falsche Befehle (es wurde nicht verstanden, was das Programm tun soll; der Sachbearbeiter hat eine falsche Vorstellung von der Rechtslage bzw wendet das Gesetz falsch an) und fehlende Befehle (es wurde vergessen, bestimmte Umstände oder Probleme zu behandeln; der vorliegende Sonderfall ist in Gesetz/Verordnung/Dienstanweisung nicht enthalten). Nur äußerst selten verursacht der Computer (die Hardware) selbst den Fehler: Falscher Entwurf (fehlerhaftes Design/Produktion der CPU; Fehlen einer zuständigen Person) oder zufällige Fehler (Störungen in der Stromversorgung, kosmische Strahlung; Akt wird verlegt usf). Zufällige Fehler können nie ausgeschlossen werden und externe Umstände können ihre Wahrscheinlichkeit stark beeinflussen.

- 1/7 • **Abstraktion:** Moderne Programme bestehen aus einer Unzahl an Einzelteilen (ein einziges Programm besteht uU direkt und indirekt, zB Bibliotheken, aus mehreren Millionen Anweisungen) und führen hochkomplexe Aufgaben aus. Dies ist für keinen einzelnen Menschen in seiner Gesamtheit mit allen Details mehr überschaubar. Es ist daher Abstraktion notwendig, dh das Problem wie auch die zu entwerfende Lösung wird in allgemeinere Varianten überführt, um einen Überblick zu gewinnen und eine Lösung im Groben zu entwerfen, welche dann in kleinere Teile zerlegt wird, die separat gelöst werden können. Dadurch wird auch **Arbeitsteilung** möglich. Typische konkrete Fälle der Abstraktion sind Hierarchisierung (ein großes Teil wird in mehrere Teile zerlegt, welche selbst wiederum zerlegt werden usf; vergleichbar einer Betriebsanlagengenehmigung, wo verschiedene Aspekte von unterschiedlichen Abteilungen bearbeitet werden und schließlich die Teilergebnisse zu einem Gesamtergebnis kombiniert werden) und Modularisierung (Zusammenfassung von Operationen und Daten einer in sich geschlossenen Aufgabe mit einer definierten Schnittstelle nach außen; vergleichbar einer Behörde, die einen umgrenzten Aufgabenbereich besitzt, entsprechendes Personal vorhält und mit der Außenwelt zB über Formulare – auf Papier oder elektronisch, zB per Internet – kommuniziert).
- 1/8 • **Standardisierung & Anpassung:** Um die hohen Kosten der Entwicklung von Software zu reduzieren, wird diese nicht für jeden einzelnen Benutzer separat entwickelt, sondern es erfolgt eine Standardisierung auf

vielen Ebenen. Grundlegende Bausteine zur Umsetzung von Programmen werden ebenso wiederverwendet wie ganze Programmteile oder große Software-Systeme. Entsprechend findet später eine Anpassung des „allgemeinen“ Programms an die konkreten Gegebenheiten statt, in denen es eingesetzt werden soll. Die Anpassung kann umfangreiche Programmierung umfassen (Änderungen, Hinzufügungen), ebenso aber auch nur eine einfache Konfiguration sein (Variationen vorgesehener Einstellmöglichkeiten). Im Rechtsbereich kann dies mit Gesetzen (allgemeine Grundlagen, passend für alle Fälle) und Bescheiden (individuelle Anwendung auf den konkreten Einzelfall, ev detaillierte Auflagen) verglichen werden (siehe auch unten: Standard-/Individualsoftware).

- **Binärprinzip:** Alle Daten sind **binär codiert**. Dieses Prinzip ist ua deshalb von eminenter Bedeutung, weil binäre Zustände („Spannung liegt über/unter einem bestimmten Schwellenwert“, „Elektronen sind vorhanden/fehlen“, „Platz ist magnetisch in die eine/andere Richtung ausgerichtet“, allgemein: „Ein/Aus“ oder symbolisch „1/0“) einer einfachen technisch-physikalischen Realisierung sehr entgegenkommen. Alle modernen Universalrechner basieren intern vollständig auf dem Binärsystem. Die Codierung beruht auf dem trivialen Prinzip, dass sich durch Aneinanderfügen von  $n$  binären Zuständen (**Bits**) genau  $2^n$  verschiedene **Bytes** der Länge  $n$  Bit herstellen lassen, denen jeweils eindeutig ein Zeichen zugeordnet werden kann. Ein sog „**Coderahmen**“ von 8 Bit ergibt also 256 zuordenbare Zeichen (Bytes), die auch heute noch typische Einteilung von Daten (vergleiche: je nach Definition sind 1000 oder 1024 Byte ein Kilobyte,  $1000^3$  oder  $1024^3$  ein Megabyte usf; die offizielle Bezeichnung nach ISO-Standard ist k/M/G für Tausender-Potenzen und ki/Mi/Gi für Vielfache von 1024; letztere werden in der Praxis selten verwendet, sodass oft Unklarheiten verbleiben). Dies führt zur unangenehmen Konsequenz, dass mit vielen Fließkommazahlen, zB  $1/3$  oder  $\pi$ , in einem Computer niemals exakt gerechnet werden kann. Es ist jedoch (mit entsprechend steigendem Aufwand) eine beliebig hohe Genauigkeit oder überhaupt eine symbolische Bearbeitung (man rechnet mit dem Text „ $1/3$ “ anstatt der Zahl 0,33333) möglich. Eine Änderung dieses Prinzips ergibt sich bei **Quantencomputern**, bei denen „quantum bits“ (qubits) mehrere Zustände gleichzeitig einnehmen können: 0, 1, oder „gleichzeitig“ (Quanten-Überlagerung) 0 und 1. Quantencomputer existieren derzeit nicht in praktisch relevanter Größe (Real bis zu 53 qubits; Simulatoren etwas mehr), doch versprechen sie bei entsprechender Vergrößerung die rasche Berechnung von derzeit nicht effizient lösbaren Problemen. Dies bringt zB enorme Herausforderungen für die Krypto-

1/9

graphie, da sich viele, aber nicht alle, Verschlüsselungen dann trivial brechen lassen.

- 1/10 • **Laufende rasche Veränderung:** Die Informatik verändert sich in rasendem Tempo und was vor ein paar Jahren Stand der Technik und aktuell war, ist heute bereits veraltet. Gewisse Grundlagen besitzen naturgemäß eine deutlich längere Lebensdauer, aber **kontinuierliche Weiterbildung** ist in der Informatik unerlässlich. Dies bedeutet auch, dass Systeme rasch veralten (zB oft schon nach 4 Jahren Betriebszeit) und nicht mehr dem aktuellen Stand entsprechen. Dennoch ist ein „Wegwerfen“ und eine Neuimplementierung vielfach aus finanziellen Gründen, wie auch aufgrund erforderlicher Benutzerschulungen, langer Zeitdauer, neuer Fehler, fehlender/mangelhafter Dokumentation etc nicht möglich oder unerwünscht. Daher müssen diese Systeme permanent gewartet und aktualisiert werden.
  
- 1/11 • **Interdisziplinarität:** Informatik ist für sich alleine nur von sehr begrenztem Nutzen. Erst ihr Einsatz in verschiedensten Bereichen ist hilfreich. Hierzu ist es erforderlich, die Software an die jeweiligen Bedürfnisse anzupassen, was zumindest ein grundlegendes **Verständnis des Einsatzbereiches** erfordert. Daraus wird auch klar, dass Informatik mehr ist, als das bloße „Programmieren“, sondern die systematische Verarbeitung von Informationen für einen **bestimmten Zweck** in einem **konkreten Kontext**.

## 2. Daten vs Information

- 1/12 Zwischen **Daten** und **Informationen** besteht ein wichtiger Unterschied: Daten sind eine interpretierfähige, in einer formalisierten Art und Weise verfügbare **Repräsentation** von Informationen, nutzbar zur Kommunikation, Interpretation oder zur Verarbeitung. Es handelt sich daher um eine allgemeine Darstellung unabhängig von einer Bedeutung. So kann das Datum „4711“ eine Anzahl von Produkten darstellen (Zahl), eine Postleitzahl (Zahl mit Spezialbedeutung) repräsentieren oder ein Markenname (Text) sein. Erst in einem konkreten Kontext wird einem Datum eine bestimmte Bedeutung beigelegt, womit dieses zur **Information** wird: Für eine bestimmte Person besitzt das Datum in einem spezifischen Kontext eine **Bedeutung**. Ohne diesen Kontext kann etwa nicht entschieden werden, welche der obigen drei Interpretationen die richtige ist. Ebenso können verschiedene Personen dasselbe Datum unterschiedlich interpretieren (eine typische Fehlerquelle in der Informatik: Der Computer verarbeitet Daten, Menschen geben jedoch Informationen ein und erwarten solche als Ausgabe!). In der Praxis

ist in den meisten Fällen nur eine einzige (oder sehr wenig) Interpretation sinnvoll oder allgemeingültig, indem eine Vielzahl an Personen sie akzeptieren würde (so kann eine Bilddatei, als Daten angesehen, technisch problemlos als Musikstück, ie Informationen, interpretiert werden – das akustische Ergebnis wird jedoch von kaum jemandem als Musik angesehen werden!).

## II. Prinzipielles über den Computer

Als meistverbreitete Erscheinungsform von IKT gilt der **Computer (Rechner)**, eine universell einsetzbare Maschine, die zur **automatisierten Verarbeitung** (sowie heute meist auch **Übermittlung**) von **Daten** dient. 1/13

Umgangssprachlich wird oft nur eine bestimmte Gruppierung typischer **Hardwarekomponenten** als Computer bezeichnet, nämlich in einem gemeinsamen Gehäuse untergebrachte Bauteile (zu denen vor allem die **Zentraleinheit**, CPU oder **Prozessor** gehört) sowie ein Bildschirm und ggf eine Tastatur/Maus. Dies darf nicht darüber hinwegtäuschen, dass Computer heute vielfach in ganz anderen Formen in Erscheinung treten: Autos, Haushaltsgeräte, Industrieanlagen und Mobiltelefone sind ebenfalls mit Computern ausgestattet. Ohne Computer ist ein modernes Leben heute undenkbar und vieles würde ohne nicht mehr funktionieren (Beispiel: Ein modernes Auto enthält ca 25–100 Computer und kann ohne manche davon nicht einmal gestartet werden). Über diese Sicht der ubiquitären Hardware darf die entscheidende Rolle der **Software** nicht übersehen werden, durch welche der Computer erst seine Funktionsfähigkeit und seinen Nutzen erlangt (im positiven wie negativen Sinne, so kann die Motorsteuerung zB Autoabgase niedrig halten, dies aber auch nur in bestimmten Situation ausführen, um andere Parameter zu optimieren). Ohne Software ist die Hardware des Computers funktions- und nutzlos – wie ebenso Software ohne Hardware wertlos ist. 1/14

Die **universelle Einsetzbarkeit** des Computers macht den entscheidenden Unterschied zum bloß mechanischen oder fest verdrahteten Rechner aus: Sie beruht auf dem Umstand, dass die Hardware des Computers nicht für die Durchführung **bestimmter** Aufgaben gebaut ist, sondern für die Bewältigung **aller** Aufgaben, die **programmiert** werden können. Hierfür ist die Software zuständig, welche typischerweise, im Gegensatz zur Hardware, verhältnismäßig leicht veränderbar ist. Während die Hardware selbst unverändert bleiben kann, erbringt sie abhängig von – und im Zusammenspiel mit – dem jeweils geladenen **Programm** genau jene Leistungen, die durch das Programm angeordnet werden. Das allgemeine Schema nebst den dazuge- 1/15



hörigen Funktionsprinzipien für solche Universalrechner (Hardware) wurde 1946 von *John von Neumann* formuliert, der auf Arbeiten aus 1938 von *Konrad Zuse* aufbaute. Die theoretischen Grundlagen über die Berechenbarkeit von Problemen (Software), dh welche Klassen von Problemen ein Computer überhaupt theoretisch in der Lage sein könnte zu lösen bzw in welchem Zeitrahmen, stammen von *Alan Turing*.

## 1. Der von-Neumann-Rechner

- 1/16** Der von-Neumann-Rechner (auch „Universalrechner“) besteht aus vier groben Funktionseinheiten: Ein-/Ausgabewerk, Speicher, Rechenwerk und Steuerwerk.
- 1/17** 1. Über das **Ein-/Ausgabewerk** werden **Programme** geladen und mit **Eingabewerten** versorgt. Die Eingabe erfolgt über spezifische Eingabegeräte oder auch durch Zuführung von anderen Rechnern oder externen Speichern („Einlaufstelle“). Weiters gibt es errechnete **Ausgabedaten** an die **Umgebung** aus („Poststelle“). Dies beinhaltet alle Arten von Schnittstellen innerhalb des Computers (zB Massenspeicher wie eine Festplatte oder SSD) oder außerhalb (Bluetooth, WLAN, Kabelnetzwerke etc).
- 1/18** 2. Im **Rechenwerk** werden **arithmetische und logische Operationen** durchgeführt („Sachbearbeiter“). Arithmetische (Addieren, Dividieren) und logische („und“, „wenn – dann“, etc) Operationen beruhen letztlich auf der technischen Realisierung dreier **elementarer Grundsaltungen** (Und-/Oder-Verknüpfung, Negation). Das bisher Dargelegte (Eingabe – Berechnung – Ausgabe) kann man auch mit einem „gewöhnlichen“ (mechanischen oder fest verdrahteten) Rechner erreichen, denn dieser wird gleichfalls über ein Eingabewerk mit Eingabewerten versorgt und gibt nach Berechnung von Resultaten im Rechenwerk diese über ein Ausgabewerk aus. Der Unterschied liegt darin, dass die **Programme**, welche die Anweisungen für die vom Rechenwerk durchzuführenden Operationen (und deren Reihenfolge) enthalten, entweder selbst bereits Teil des Rechenwerks (also „fest“) sind oder sich „im Kopf des Bedieners“ befinden. Um **wechselnde** Programme eingeben und **automatisch abarbeiten** lassen zu können, sind die beiden weiteren Funktionseinheiten des Universalrechners erforderlich: Speicher und Steuerwerk.
- 1/19** 3. Der **Speicher** dient dazu, sowohl Daten („Aktenarchiv“) wie auch Programme („Bibliothek“) abzulegen. Abgesehen vom Massenspeicher, der über ein Ein-/Ausgabewerk angesprochen wird, interessiert uns hier lediglich der **Hauptspeicher** (engl **memory**, **RAM** von Random Access Memory). Das ist jener Speicher, der neben den zu bearbeitenden Daten auch die jeweils gerade auszuführenden Programme oder Programmteile



aufnimmt. Der Speicher ist so aufgebaut, dass jede Speicherzelle „**adressiert**“, dh numerisch gekennzeichnet, in **direktem Zugriff erreichbar**, ihr Inhalt **abrufbar** und beliebig oft wieder mit neuem Inhalt **überschreibbar** ist. Kennzeichen des von-Neumann-Rechners ist, dass Programm und Daten im **gleichen** Speicher abgelegt sind, daher Programme auch als Daten interpretiert werden können bzw umgekehrt. Dies ermöglicht erst die Vielfältigkeit der Software: Anweisungen können mittels eines Computers bearbeitet werden (= Daten), bevor sie „anders interpretiert“ und als Software (= Programme) ausgeführt werden. Aus Sicherheitsgründen wird diese **Dualität** bei modernen Rechnern eingeschränkt, da sie vielfältige Angriffe ermöglicht.

4. Das **Steuerwerk** („Management“) sorgt für die **Decodierung** und **Interpretation** der aus dem Hauptspeicher geholten **Programmbefehle** und sendet **Steuerbefehle** an alle an der Befehlsdurchführung beteiligten Funktionseinheiten, insbesondere an das Rechenwerk, sodass diese zeitlich und logisch korrekt ausgeführt werden. 1/20

Das Zusammenspiel der genannten Funktionseinheiten mit den jeweils geladenen Programmen ist durch die Prinzipien von-Neumanns vollständig bestimmt und garantiert die automatische Abarbeitung der Aufgabe, für die das Programm geschrieben ist. Damit ist gewährleistet, dass bei identischem Zustand und exakt (!) gleichen Eingaben jedes Mal das gleiche Ergebnis berechnet wird (Determinismus; siehe oben). Zu berücksichtigen ist, dass dies nur das Grundkonzept darstellt; zur Leistungssteigerung werden viele Tricks verwendet und Teile vervielfacht etc. So enthalten viele CPUs ebenso einen kleinen, dafür besonders schnellen Speicher (Cache; „Aktenstapel am Schreibtisch“), da die Kommunikation mit dem Hauptspeicher heute einen Flaschenhals darstellt. 1/21

## 2. Äquivalenz von Hard- und Software

Obwohl Hard- und Software anscheinend klar voneinander unterscheidbar sind (Hardware = kann man angreifen; Software = gedankliche, logische Dinge), besteht in der Realität eine enorme **Verschiebbarkeit der Grenze**. Liegt ein absolut minimaler (eine der ersten in Massen eingesetzten CPUs für PCs – 8086 – kam mit 29.000 Transistoren aus) Kern in der Hardware vor, so kann die gesamte „fehlende“ Funktionalität durch Software ersetzt werden. Umgekehrt kann jede beliebige Software in Hardware „gegossen“ werden (moderne CPUs bestehen aus bis zu 3,2 Milliarden Transistoren). Dies kann mit der Kompetenzverteilung zwischen Bund und Ländern verglichen werden: Bis auf einen minimalen Anteil (ansonsten existierten 9 un- 1/22

abhängige Staaten) können Aufgaben beliebig zwischen Bund und Ländern verschoben werden. Genau wie in diesem Beispiel ist die Verschiebung in der Praxis wegen Kosten und Effizienz deutlich eingeschränkt.

- 1/23** Für eine Umschichtung von Software nach Hardware stehen zwei Varianten zur Verfügung: Das Programm wird in einen festen und unveränderlichen Speicher programmiert (Wechsel des Programms nur mehr durch Austausch des entsprechenden Teils der Hardware oder spezielle Speicher möglich; typisch bei Steuerungsanlagen, aber heute selten verwendet) oder es wird die Funktionalität selbst in elektronische Schaltungen übersetzt. Letzteres findet in der Praxis in großem Umfang statt. So wird beispielsweise in modernen Prozessoren eine Division in Hardware ausgeführt, während sie früher durch umfangreiche und komplexe Programme „händisch“ erfolgen musste. Gleiches geschieht für oft verwendete Spezialaufgaben zB in Hochleistungs-Grafikkarten oder für die Verschlüsselung von Daten. Direkt von der Hardware ausführbare Befehle steigen in ihrer Komplexität, werden jedoch sehr viel rascher ausgeführt. Sinnvoll ist dies nur, wenn die Befehle auch oft und von vielen Programmen benötigt werden; daher müssen diese „allgemein genug“ bleiben.

### III. Hardware

- 1/24** Als **Hardware** bezeichnet man alle **physikalisch-materiellen Teile eines Datenverarbeitungssystems**. Zur Hardware werden also alle (zumindest prinzipiell) sinnlich erfassbaren Komponenten wie Zentraleinheit, Prozessoren, Peripheriegeräte, Netzkabel, etc gezählt, ebenso wie alle Datenträger, letztere jedoch nicht hinsichtlich ihres Inhalts.

#### 1. Hauptgruppen von IKT-Hardware

- 1/25** Als **CPU (Central Processing Unit; Zentraleinheit)** werden in der technischen Realisierung die bei der Schilderung des von-Neumann-Rechners genannten Funktionseinheiten Rechen- und Steuerwerk mit dem Hauptspeicher auf einer Leiterplatte (**Motherboard**) zusammengeführt, auf welcher auch **Hardwareschnittstellen** (zB als Steckplätze oder genormte Stecker) ausgebildet sind. In dem die Zentraleinheit aufnehmenden **Gehäuse** sind üblicherweise weitere Komponenten wie **Stromversorgung**, gegebenenfalls **Lüfter** und auch verschiedene externe Geräte wie Massenspeicher und Kommunikationsschnittstellen eingebaut.
- 1/26** **Externe Geräte** sind solche, die über Schnittstellen oder Leitungen (zB Kupfer, Glasfaser oder Funk) mit der Zentraleinheit in Verbindung stehen; sie

zählen zur sog „Peripherie“. Sie dienen zur Speicherung von Massendaten (**Festplatte, DVD-Laufwerk, SSD**), als Ein-/Ausgabegeräte (zB **Tastatur, Maus, Touchscreen, Bildschirm, Drucker**) oder – als **Steckkarten** ausgebildet – als sog „Steuerungperipherie“ zur Signalversorgung anderer Peripherie (**Grafikkarte, Netzwerkschnittstellen** etc). Verschiedenste Teile davon sind heute manchmal bereits auf Motherboards oder direkt in der CPU integriert, insbesondere bei sehr kompakten Geräten, zB Mobiltelefonen.

## 2. Betriebssicherheit, Wartung

Die Betriebssicherheit oder Zuverlässigkeit von Hardware stellt einen kritischen Faktor dar, der durch die sog „**mittlere Betriebsdauer zwischen Ausfällen**“ (**mean time between failures, MTBF**) gemessen wird. Abgesehen von die Betriebssicherheit verbessernden Maßnahmen an der Gerätekonfiguration selbst (zB Einsatz einer unterbrechungsfreien Stromversorgung, **USV**) bemisst sich der einzusetzende **Wartungsaufwand**, um die MTBF über einem dadurch zu erwartenden Schwellenwert zu halten, an den Kosten, die ein durch Hardwaredefekt bedingter Betriebsausfall verursacht. **1/27**

Die risikoreichste (und oft nur im Glücksfall billigste) Variante ist, ohne Absicherung durch einen Wartungsvertrag nur im Anlassfall einer bemerkten Störung diese beheben zu lassen. Man kann auch allein die Anlasswartung, dh Störungsbehebung bei Hardwareausfall, vertraglich absichern. Allerdings ist auch schon bei dieser Minimalvariante eine **Reaktionszeit** zu vereinbaren, innerhalb welcher nach Störungsmeldung mit der Störungsbehebung begonnen werden muss (und/oder bis wann diese abzuschließen ist: **Entstörzeit**). Auch der Ort der Behebung (Einsendung nötig oder Vor-Ort-Service) ist wichtig. **1/28**

Eine regelmäßige oder **geplante Wartung** schließt über die Störungsbehebung im Anlassfall hinaus auch **präventive** Maßnahmen mit ein: Die Hardware wird in regelmäßigen Intervallen (auch unter Einsatz von Prüfprogrammen) überprüft und allenfalls dabei entdeckte mangelhafte oder bedenkliche Komponenten ausgetauscht. Wartungsbesuche und -maßnahmen, insbesondere der Austausch von Komponenten, sind nachvollziehbar zu dokumentieren. **1/29**

In der Praxis findet heute **Redundanz** in großem Ausmaß Anwendung. Hierbei besitzt ein Computer beispielsweise zwei Netzgeräte, sodass bei Defekt oder Ausfall der Versorgung eines Gerätes das System durch das zweite Netzteil weiterhin versorgt wird. Gleiches erfolgt bei Massenspeichern, Verkabelung, Netzwerkinfrastruktur, Servern und sogar ganzen Re- **1/30**

chenzentren. Nachteil ist, dass die Wahrscheinlichkeit für einen Fehler in **irgendeinem** Teil aufgrund der höheren Anzahl an Teilen **steigt**, sowie dass Kosten im Vorhinein anfallen und sich laufende Kosten erhöhen. Durch die meist extrem rasche Umschaltung können jedoch Ausfallzeiten verkürzt oder meist sogar vollständig vermieden werden, sofern nur ein einzelner Defekt auftritt.

### 3. Mobile Geräte

- 1/31** Mobile Geräte steigen in ihrer Bedeutung. Hierbei handelt es sich um **Mobiltelefone**, bei denen der Computer-Charakter im Laufe der Zeit immer stärker wurde, sowie diverse Arten von **Tablets** und Laptops. Moderne Geräte zeichnen sich durch kleine Größe und geringes Gewicht sowie diverse Kommunikationsmöglichkeiten (Mobilfunknetz, WLAN, Bluetooth) aus. Wichtige Unterschiede zu „klassischen“ Computern sind, dass Erweiterungen nur extern (Docking-Stationen oder über Kabel) bzw gar nicht möglich sind (Anschluss von Grafikkarte, Drucker, Festplatte etc ist nicht vorgesehen). Fehlende Funktionalität wird ersetzt, indem entsprechende Geräte im Internet oder an anderen Computern bzw in der Cloud vorhanden sind und über die Kommunikationskanäle (WLAN, mobiles Internet) „mitbenutzt“ werden. Bei der Bedienung hat sich überwiegend der „Touchscreen“ durchgesetzt, teilweise mit Stift für höhere Genauigkeit. Sicherheit ist bei mobilen Geräten besonders wichtig, da umfangreiche und permanente Kommunikation erforderlich ist und sie leicht abhandenkommen.

### 4. Eingebettete Systeme („Embedded systems“)

- 1/32** Hierbei handelt es sich um Computer, die in einen technischen Kontext eingebunden und **Teil eines größeren Systems** sind. Typische Beispiele sind Steuerungs- und Überwachungscomputer für Industrieanlagen, Transportmittel etc. Kennzeichnendes Merkmal ist, dass sie für den Benutzer unsichtbar sind. Dementsprechend besitzen sie oft keinerlei Ein- oder Ausgabegeräte für Menschen, sondern kommunizieren lediglich über technische Schnittstellen, typischerweise mit einer Vielzahl anderer eingebetteter Systeme, und werden, wenn nicht überhaupt autonom, aus der Ferne gesteuert. Im Gegensatz zum klassischen Computer wird hier die Hardware vielfach spezifisch an die Aufgabe angepasst. Obwohl es sich weiterhin um Universalrechner handelt, sind diese für bestimmte Aufgaben optimiert und alles hierfür nicht Erforderliche wird weggelassen. Sie besitzen vergleichsweise wenig Speicher, benötigen sehr wenig Platz und Energie und sind aufgrund hoher Stückzahlen und Verzicht auf die allerneueste Technologie meist kos-