Abschlussprüfung Teil 1 Lernzettel

Autor: u/DeFyuseOW, 01.03.2025

Diese Zusammenfassung deckt die wesentlichen theoretischen Themen der Abschlussprüfung Teil 1 für alle IT-Berufe gemäß dem ab Januar 2025 gültigen Prüfungskatalog des U-Form Verlags ab. Die Themen sind in größere Überthemen grob sortiert. Allgemeine Grundlagen stehen am Anfang.

Nicht enthalten sind folgende praxisorientierte Themenbereiche:

- Situationsgerechte Kundenkommunikation
- Interpretation englischsprachiger Texte
- Technische und kaufmännische Texte in deutscher und englischer Sprache
- Präsentation und Medienkompetenz
- Fehler in gegebenem Quellcode finden

Diese Bereiche wurden ausgelassen, da sie eher praktische Fertigkeiten und Kompetenzen abfragen und weniger theoretisches Fachwissen erfordern.

- 11.25 Personal Firewall

1. Grundlagen

1.1 Künstliche Intelligenz

KI

KI steht für "Künstliche Intelligenz" und umfasst alles, was Computer und Maschinen befähigt, menschenähnliche Intelligenzleistungen zu erbringen - von der Mustererkennung über das Verarbeiten natürlicher Sprache bis hin zum selbstständigen Lernen und Problemlösen.

Neuronales Netzwerk

Ein **Neuronales Netzwerk** ist ein Algorithmus, welcher Daten verarbeitet - "nachgebaut" nach dem biologischen Neuronalen Netzwerk (Gehirn).

Überwachtes Lernverhalten / Supervised Learning

Die Trainingsdaten werden vor dem Trainieren beschriftet und so lernt das Netzwerk aufgrund dieser Daten. Benötigt mehr Vorarbeit, aber erzielt bessere Ergebnisse.

Unüberwachtes Lernverhalten/unsupervised learning

Die Trainingsdaten werden NICHT klassifiziert vor dem Trainieren. Das Netzwerk lernt dadurch z.B. die Daten nur zu clustern, also in Gruppen zusammenzufassen.

Overfitting

Von **Overfitting** spricht man, wenn ein Neuronales Netzwerk zu gut auf die Trainingsdaten trainiert wurde und somit NUR auf die Trainingsdaten angewendet werden kann.

Maschinelles Lernen

Maschinelles Lernen ist ein Teilbereich der KI.

Deep Learning / Deep Neural Network

Ein **Deep Neural Network** ist ein Neuronales Netzwerk mit vielen Hidden/Zwischen-Layern

Schwache KI

Eine **schwache KI** ist eine Künstliche Intelligenz, welche nicht selbstständig handeln kann und Eingabe eines Menschen braucht

Starke KI

Eine **starke KI** ist eine selbst handelnde KI, die nicht auf irgendeine menschliche Interaktion angewiesen ist

Generative AI

Eine **Generative KI** ist eine KI, welche kunstvolle Sachen generieren kann (Text, Bild, Ton, etc.)

Deep Fake

Ein **Deep Fake** ist ein durch KI bearbeitetes Video/Bild. Meistens wird das Gesicht und die Stimme einer anderen Person ersetzt

KI Software

Alltag:

- **DeepL** - Einer der präzisesten KI-Übersetzer mit natürlich klingenden Übersetzungen und Dokumentenverarbeitung
- **Grammarly** - Prüft Rechtschreibung und Grammatik mit KI und gibt Verbesserungsvorschläge
- **Google Lens** - Erkennt Objekte, Text und Landmarken in Bildern
- **Google Fotos** - Kategorisiert automatisch Bilder und erkennt Gesichter

Entwicklung:

- **GitHub Copilot** - Unterstützt bei der Code-Entwicklung durch KI-gestützte Vorschläge
- **ChatGPT** - Hilft bei Code-Erklärungen, Debugging und Programmierkonzepten

1.2 Bits & Bytes

Bits und Bytes sind die grundlegenden Maßeinheiten für digitale Daten. Ein Bit (b) ist die kleinste Informationseinheit und kann zwei Zustände annehmen: 0 oder 1. Acht Bits ergeben ein Byte (B).

Einheiten

Es gibt zwei unterschiedliche Systeme zur Angabe von Speichergrößen:

Im **dezimalen System (SI)** basieren die Einheiten auf Vielfachen von 1.000 (Basis 10):

- 1 Kilobyte (KB) = 1.000 Byte
- 1 Megabyte (MB) = 1.000 KB = 1.000.000 Byte
- 1 Gigabyte (GB) = 1.000 MB = 1.000.000.000 Byte
- 1 Terabyte (TB) = 1.000 GB = 1.000.000.000.000 Byte

Im **binären System (IEC)** basieren die Einheiten auf Vielfachen von 1.024 (Basis 2):

- 1 Kibibyte (KiB) = 1.024 Byte
- 1 Mebibyte (MiB) = 1.024 KiB = 1.048.576 Byte
- 1 Gibibyte (GiB) = 1.024 MiB = 1.073.741.824 Byte
- 1 Tebibyte (TiB) = 1.024 GiB = 1.099.511.627.776 Byte

1.3 Zahlensysteme

Zahlensysteme sind Methoden zur Darstellung von Zahlen. Verschiedene Kulturen und technologische Bedürfnisse haben zur Entwicklung einer Vielfalt von Zahlensystemen geführt. Sie reichen von einfachen Strichzählungen bis hin zu komplexen digitalen Codierungen.

Grundlegende Zahlenformen

Ganzzahlen

- **Beispiele:** -3, 0, 42. Ganze Zahlen ohne Bruchteil.

Festkommazahlen

- **Beispiele:** 3.14, -0.001. Zahlen mit einer festen Anzahl von Dezimalstellen.

Gleitkommazahlen

- **Beispiele:** 1.23e4 (entspricht 12300). Zahlen mit variabler Anzahl von Dezimalstellen.

Buchstaben

- **Beispiele:** A, b, Я. Zeichen, die keine Zahlen sind, aber in manchen Zahlensystemen verwendet werden.

Wichtige Zahlensysteme

Dezimalsystem

- **Anzahl Zeichen:** 10 (0 bis 9)
- **Grundlage:** 10 Finger, daher weit verbreitet.

Binärsystem

- **Anzahl Zeichen:** 2 (0 und 1)
- **Verwendung:** In der Computertechnik für digitale Zustände (An/Aus, 0/5V, 0/3.3V).

Oktalsystem

- **Anzahl Zeichen:** 8 (0 bis 7)
- **Besonderheit:** Eine Oktalstelle umfasst genau 3 Bit.

Hexadezimalsystem

- **Anzahl Zeichen:** 16 (0 bis 9 und A bis F)
- **Besonderheit:** Eine Hexadezimalstelle umfasst genau 4 Bit.

Dezimal	Binär	Hexadezimal	Oktal
0	0000	0	0
1	0001	1	1
2	0010	2	2

Dezimal	Binär	Hexadezimal	Oktal
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	10000	10	20

Ganzzahlen und ihre Darstellung

Dezimal

- **Bereich:** 0 bis n
- **Beispiel:** 42

Binär

- **1 Byte:** 0 bis 255
- **2 Bytes:** 0 bis 65535
- **4 Bytes:** 0 bis etwa 4 Milliarden (Unsigned Integer)

Binary Coded Decimal (BCD)

- **Beispiel:** 77 = 0111 0111

Buchstaben und Zeichencodierung

ASCII

- **Verwendung:** Standardisierte Codierung für lateinische Buchstaben und Zeichen.

Unicode

- **Verwendung:** Umfassende Codierung für Schriftzeichen vieler Sprachen und Symbole.

2. Hardware

2.1 CPU

Die CPU (Central Processing Unit) ist das Hauptrechenwerk eines Computers und führt Befehle aus. Sie bestimmt maßgeblich die Geschwindigkeit und Leistungsfähigkeit eines Systems.

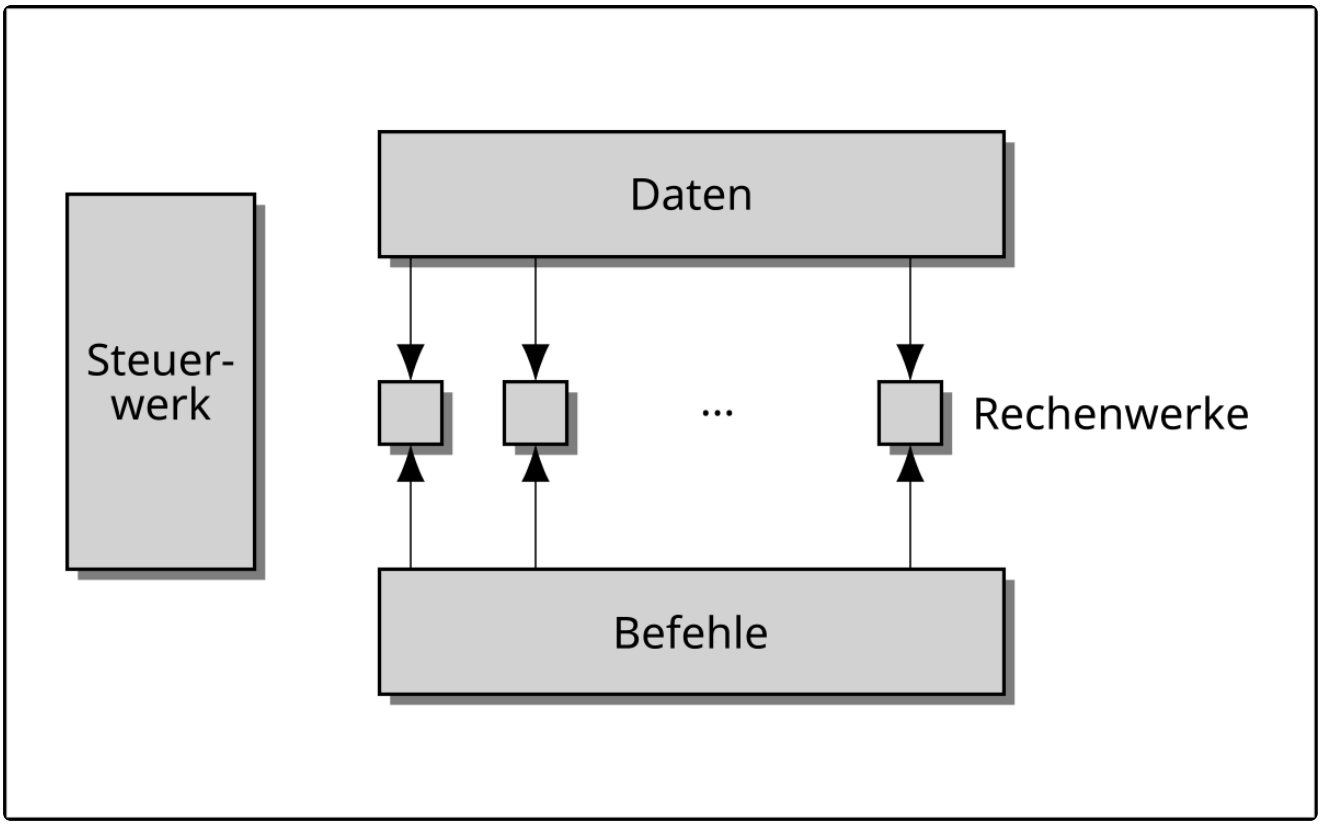
Harvard-Architektur

Funktionsblöcke

1. Rechenwerk (ALU - Arithmetical Logical Unit)
    - **Aufgaben:** Führt arithmetische und logische Operationen durch.
2. Leit- & Steuerwerk (CU - Control Unit)
    - **Aufgaben:** Kontrolliert den Ablauf der Befehle.
3. Speicherwerk
    - **Aufgaben:** Getrennte Speicher für Daten (Daten-Speicher) und Programme (Programm-Speicher).
4. Ein-/Ausgabewerk (IO Unit)
    - **Aufgaben:** Zuständig für die Ein- und Ausgabe von Daten und die Interaktion mit dem Benutzer.

Verbindungen

Die Harvard-Architektur verwendet getrennte Verbindungswege (Busse) für Daten und Programme, was zu einer erhöhten Verarbeitungsgeschwindigkeit führen kann.



Vorteile und Nachteile

Vorteile	Nachteile
Höhere Geschwindigkeit	Komplexere Struktur
Weniger Zugriffskonflikte	Höherer Hardwareaufwand
Bessere Sicherheit	Geringere Flexibilität

Anwendungsbereiche

Die Harvard-Architektur findet häufig Anwendung in eingebetteten Systemen und Mikrocontrollern, wo Effizienz und Geschwindigkeit entscheidend sind.

Von-Neumann-Architektur

Funktionsblöcke

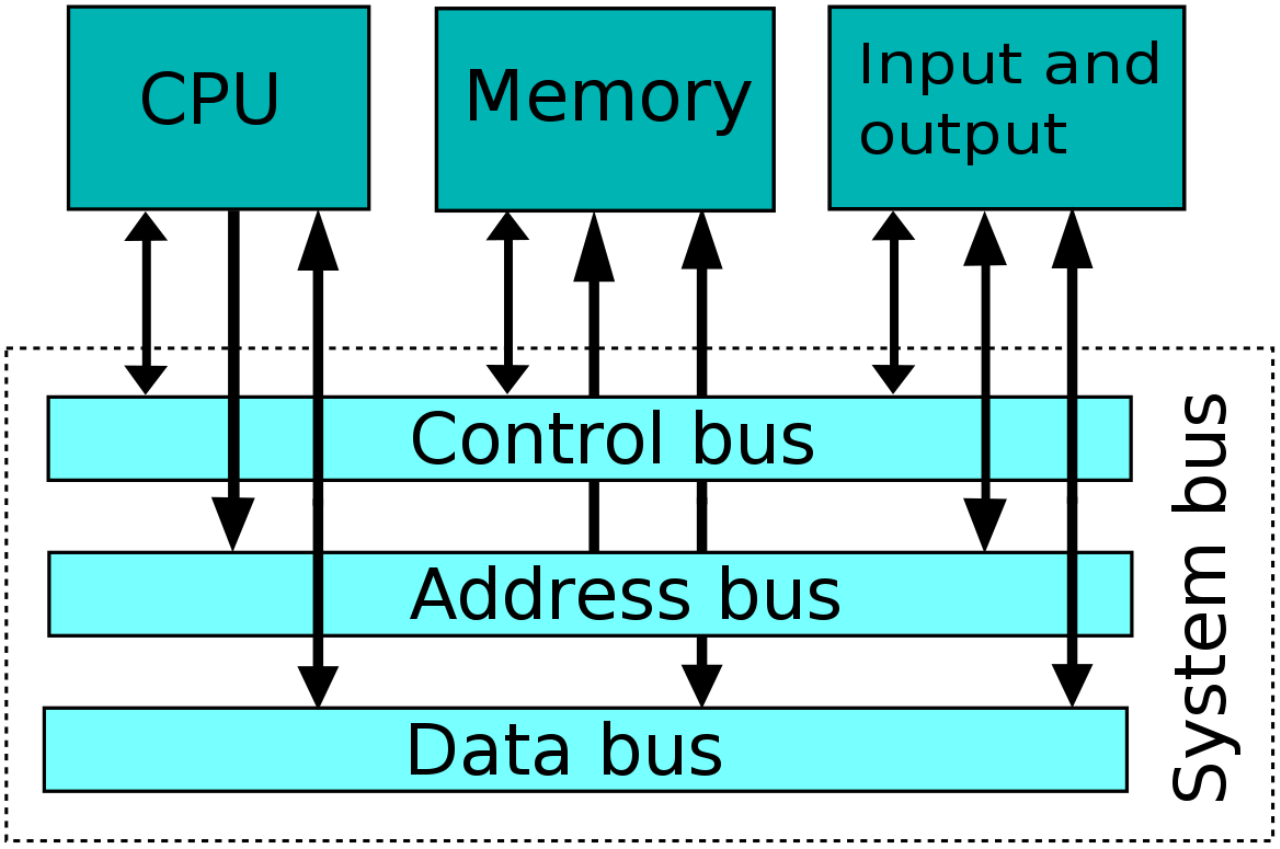
1. Rechenwerk (ALU - Arithmetical Logical Unit)
    - **Aufgaben:** Führt arithmetische und logische Operationen durch.
2. Leit- & Steuerwerk (CU - Control Unit)
    - **Aufgaben:** Kontrolliert den Ablauf der Befehle.
3. Speicherwerk (Memory - RAM)

- **Aufgaben:** Speichert sowohl Daten als auch Programme.

4. Ein-/Ausgabewerk (IO Unit)
    - **Aufgaben:** Zuständig für die Ein- und Ausgabe von Daten und die Interaktion mit dem Benutzer.

Verbindungen

1. Kontrollbus / Steuerbus
    - **Funktion:** Übermittelt Steuersignale zwischen den Funktionsblöcken.
2. Datenbus
    - **Funktion:** Transportiert Daten zwischen den Funktionsblöcken.
3. Adressbus
    - **Funktion:** Übermittelt die Adressen von Speicherzellen, auf die zugegriffen wird.



Zusammenarbeit bei Lese- und Schreibzugriffen

Bei Lese- und Schreibzugriffen kommuniziert das Steuerwerk mit dem Speicherwerk, um die notwendigen Daten zu erhalten oder zu speichern. Der Adressbus gibt die Speicheradresse an, der

Datenbus transportiert die Daten und der Kontrollbus steuert den Ablauf.

CISC

CISC (Complex Instruction Set Computing) ist eine Prozessorarchitektur, die viele komplexe und spezialisierte Befehle direkt in Hardware implementiert. Diese Befehle können in einem einzigen Maschinenbefehl komplexe Operationen ausführen, was zwar die Programmierung vereinfacht, aber zu einer komplexeren und langsameren CPU-Architektur führt.

RISC

RISC (Reduced Instruction Set Computing) verwendet einen minimalen Satz einfacher Befehle, die in genau einem Taktzyklus ausgeführt werden können. Durch diese Vereinfachung wird die CPU schneller und energieeffizienter, auch wenn Programme dadurch mehr einzelne Befehle benötigen.

Bus

Der Bus ist das System zur Datenübertragung innerhalb der CPU und zwischen Komponenten.

- **Datenbus**: Transportiert Daten zwischen CPU, RAM und anderen Komponenten
- **Adressbus**: Bestimmt Speicheradressen für Lese- und Schreibzugriffe
- **Steuerbus**: Überträgt Steuerbefehle zur Synchronisation

2.2 Speicher (RAM)

Der Arbeitsspeicher (RAM - Random Access Memory) dient als kurzfristiger Speicher für laufende Prozesse und Programme. Er ist schneller als Festplattenspeicher, verliert aber seinen Inhalt beim Ausschalten des Computers.

Arten

- **SRAM (Static RAM)**: Schnell, aber teuer (z. B. Cache-Speicher)
- **DRAM (Dynamic RAM)**: Günstiger, aber langsamer (z. B. Hauptspeicher)
- **DDR (Double Data Rate)**: Aktuelle Versionen sind DDR4 und DDR5, unterscheiden sich in Geschwindigkeit und

Energieeffizienz

Merkmale

- **Kapazität**: Bestimmt, wie viele Programme gleichzeitig laufen können
- **Taktfrequenz**: Misst die Geschwindigkeit der Datenübertragung (z. B. 3200 MHz)
- **Latenz (CL - CAS Latency)**: Verzögerung bei Speicherzugriffen

Umrechnung

DDR-RAM (Double Data Rate) überträgt pro Takt zwei Datenpakete, weshalb die effektive Datenrate doppelt so hoch ist wie die Taktrate.

Taktfrequenz → Datenrate:

$$\text{Datenrate (MB/s)} = \text{Taktfrequenz (MHz)} \times 2 \times \text{Busbreite (Byte)}$$

Typische Busbreite für DDR-RAM: **64 Bit = 8 Byte**

Beispiel:
DDR4-3200 (3200 MHz)

$$3200 \times 2 \times 8 = 51200\ \text{MB/s} \quad (= 51{,}2\ \text{GB/s})$$

Datenrate → Taktfrequenz

$$\text{Taktfrequenz (MHz)} = \frac{\text{Datenrate (MB/s)}}{2 \times \text{Busbreite (Byte)}}$$

2.3 Datenspeicher

Der Datenspeicher dient der langfristigen Speicherung von Daten und Programmen.

HDD (Hard Disk Drive)

- Mechanische Festplatte mit magnetischen Platten
- **Vorteile**: Günstig, hohe Speicherkapazität
- **Nachteile**: Langsam, empfindlich gegen Erschütterungen

SSD (Solid State Drive)

- Speicherchips ohne bewegliche Teile

- **Vorteile**: Schnell, robust, leise
- **Nachteile**: Teurer als HDD, begrenzte Schreibzyklen

Flash-Speicher

- Speichertechnologie für SSDs, USB-Sticks, SD-Karten
- **Arten**: NAND-Flash (günstig, langlebig) und NOR-Flash (schnell, für Firmware)

Dateisysteme

- **FAT32**: Kompatibel, aber auf 4 GB Dateigröße begrenzt
- **NTFS**: Windows-Standard, unterstützt große Dateien
- **exFAT**: Optimiert für externe Speichergeräte
- **EXT4**: Standard für Linux-Systeme

2.4 Netzwerk

RJ45

- Standardstecker für Ethernet-Kabel (LAN-Verbindungen)
- Unterstützt verschiedene Geschwindigkeiten (z. B. 100 Mbit/s, 1 Gbit/s, 10 Gbit/s)
-

LWL (Lichtwellenleiter)

- Glasfasertechnik für schnelle Datenübertragung
- **Vorteile**: Hohe Bandbreite, geringe Latenz, störungsfrei
- **Nachteile**: Teurer als Kupferkabel, empfindlicher

Netzwerkkarte

- Hardware-Schnittstelle zur Verbindung eines Geräts mit einem Netzwerk
- Gibt es für **Ethernet (kabelgebunden)** und **WLAN (kabellos)**

WLAN (Wireless Local Area Network)

- Kabellose Netzwerktechnologie basierend auf IEEE 802.11-Standards

- Verschiedene Standards (z. B. Wi-Fi 5, Wi-Fi 6) bestimmen Geschwindigkeit und Reichweite

LAN (Local Area Network)

- Lokales Netzwerk innerhalb eines Hauses oder Unternehmens
- Verbindung über Ethernet oder WLAN

Gateway/Router

- **Router**: Verbindet verschiedene Netzwerke (z. B. Heimnetz mit Internet)
- **Gateway**: Vermittelt zwischen verschiedenen Protokollen oder Netzwerken

Switch

- Verteilt Netzwerkverkehr innerhalb eines LANs
- Arbeitet auf Schicht 2 (Data Link Layer) des OSI-Modells

Access Point

- Erweitert WLAN-Abdeckung in einem Netzwerk
- Oft in Router integriert oder als eigenständiges Gerät

DHCP (Dynamic Host Configuration Protocol)

- Automatische Vergabe von IP-Adressen innerhalb eines Netzwerks
- Erleichtert Netzwerkkonfiguration, da Geräte keine feste IP-Adresse benötigen

NAT (Network Address Translation)

NAT übersetzt private IP-Adressen in eine öffentliche IP-Adresse für den Internetzugriff.

- **SNAT** (Source NAT): Private → Öffentliche IP
- **DNAT** (Destination NAT): Öffentliche → Private IP
- **PAT** (Port Address Translation): Mehrere private Adressen teilen eine öffentliche IP

IPv4

Internet Protocol Version 4 (IPv4) ist ein verbindungsloses, paketvermitteltes Protokoll zur Adressierung von Netzwerkteilnehmern. Es verwendet 32-Bit-Adressen, die in vier Oktette aufgeteilt sind (z. B. `192.168.1.1`) und ist dadurch auf ca. 4,3 Milliarden Adressen begrenzt.

Aufbau einer IPv4-Adresse

Eine IPv4-Adresse besteht aus zwei Teilen:

- **Netzwerkanteil**: Identifiziert das Netzwerk.
- **Hostanteil**: Identifiziert das Gerät innerhalb des Netzwerks.

Subnetting

Subnetting unterteilt ein Netzwerk in kleinere Teilnetze, um Adressen effizienter zu nutzen und Broadcast-Domänen zu begrenzen.

Subnetzmaske

Eine Subnetzmaske bestimmt den Netzwerk- und Hostanteil. Beispiele:

- `/8` = `255.0.0.0`
- `/16` = `255.255.0.0`
- `/24` = `255.255.255.0`

Beispiel für Subnetting

Netzwerk `192.168.1.0/24` soll in zwei Subnetze unterteilt werden:

- `/25` (`255.255.255.128`) → Erste Hälfte: `192.168.1.0 - 192.168.1.127`
- `/25` (`255.255.255.128`) → Zweite Hälfte: `192.168.1.128 - 192.168.1.255`

Berechnung der Subnetze

1. **Anzahl der Subnetze:** `2^n` (n = zusätzliche Subnetz-Bits)
2. **Anzahl der Hosts pro Subnetz:** `2^h - 2` (h = verbleibende Host-Bits)
3. **Subnetzgröße:** `256 - letzter Wert der Subnetzmaske`

Beispiel: `/26` → `255.255.255.192`

- 4 Subnetze (`/24` → `/26` → `2^2 = 4`)
- 62 Hosts pro Subnetz (`2^6 - 2 = 62`)

Private IPv4-Adressen

Diese Adressen sind nicht im Internet routbar und werden in privaten Netzwerken genutzt.

- **Klasse A:** `10.0.0.0 - 10.255.255.255`
- **Klasse B:** `172.16.0.0 - 172.31.255.255`
- **Klasse C:** `192.168.0.0 - 192.168.255.255`

Wichtige Sonderadressen

- **Netzwerkadresse:** Erste Adresse eines Subnetzes (z. B. `192.168.1.0`)
- **Broadcast-Adresse:** Letzte Adresse eines Subnetzes (z. B. `192.168.1.255`)
- **Loopback-Adresse:** `127.0.0.1` für lokale Tests
- **APIPA:** `169.254.0.0/16` (wird automatisch zugewiesen, wenn kein DHCP verfügbar ist)

IPv6

Internet Protocol Version 6 (IPv6) ist der Nachfolger von IPv4 und bietet eine größere Adresskapazität sowie verbesserte Netzwerkfunktionen. IPv6 verwendet 128-Bit-Adressen, die in acht Gruppen zu je vier hexadezimalen Ziffern dargestellt werden (z. B. `2001:0db8:85a3:0000:0000:8a2e:0370:7334`).

Aufbau einer IPv6-Adresse

IPv6-Adressen bestehen aus:

- **Netzwerkpräfix:** Identifiziert das Netzwerk (meist 64 Bit)
- **Interface Identifier:** Identifiziert das Gerät im Netzwerk (meist 64 Bit)

Beispiel-Adresse:

```
2001:0db8:85a3:0000:0000:8a2e:0370:7334
```

Verkürzt (führende Nullen können weggelassen werden):

```
2001:db8:85a3::8a2e:370:7334
```

Adresstypen

- **Unicast**: Kommunikation mit einem einzigen Empfänger
  - **Global Unicast (GUAs)**: Öffentlich routbare Adressen (z. B. `2000::/3`)
  - **Link-Local**: Nur im lokalen Netzwerk gültig (z. B. `FE80::/10`)
  - **Unique Local (ULA)**: Private IPv6-Adressen (`FC00::/7`)
- **Multicast**: Kommunikation mit mehreren Empfängern (`FF00::/8`)
- **Anycast**: Mehrere Geräte teilen sich eine Adresse, das nächstgelegene antwortet

IPv6-Präfixe

IPv6 verwendet die **CIDR-Notation** zur Angabe des Netzwerkteils:

- `/64` → Standard für Netzwerke (z. B. Heim- und Firmennetze)
- `/48` → Zuweisung für Unternehmen
- `/32` → ISPs erhalten oft ein `/32`-Netz und teilen es auf

IPv6-Adresszuweisung

1. **Statische Zuweisung**: Manuelle Vergabe einer IPv6-Adresse
2. **DHCPv6**: Automatische Vergabe durch einen DHCPv6-Server
3. **SLAAC (Stateless Address Autoconfiguration)**: Selbstkonfiguration der IP-Adresse anhand des Netzpräfixes und der MAC-Adresse

Subnetting in IPv6

IPv6 benötigt kein klassisches Subnetting wie IPv4. Ein **/64**-Präfix wird standardmäßig für ein Netzwerk verwendet, kleinere Subnetze sindeher selten.

Beispiel für Subnetting mit `/48`:

- `2001:db8:1234::/48` → Basisnetzwerk
- `2001:db8:1234:0001::/64` → Erstes Subnetz

- `2001:db8:1234:0002::/64` → Zweites Subnetz

Wichtige Sonderadressen

- **Loopback-Adresse:** `::1` (entspricht `127.0.0.1` in IPv4)
- **Unspecified-Adresse:** `::` (wird verwendet, wenn keine Adresse zugewiesen ist)
- **All Nodes Multicast:** `FF02::1` (alle Geräte im lokalen Netzwerk)
- **All Routers Multicast:** `FF02::2` (alle Router im lokalen Netzwerk)

2.5 Grafikkarte

Die Grafikkarte (GPU – Graphics Processing Unit) ist für die Bildberechnung und Darstellung verantwortlich.

Arten von Grafikkarten:

- **Integrierte GPU:** In der CPU enthalten, energiesparend (z. B. Intel UHD, AMD Vega)
- **Dedizierte GPU:** Eigenständige Karte für hohe Leistung (z. B. NVIDIA GeForce, AMD Radeon)

Wichtige Merkmale:

- **VRAM (Videospeicher):** Speichert Texturen und Bilddaten (z. B. 8 GB GDDR6)
- **CUDA / OpenCL:** Technologien für parallele Berechnungen
- **Raytracing:** Echtzeit-Lichtsimulation für realistische Grafiken

Anwendungsbereiche:

- Gaming
- 3D-Modellierung
- Künstliche Intelligenz (Deep Learning)

2.6 Netzwerkprotokolle und OSI-Modell

Netzwerkprotokolle

Netzwerkprotokolle sind standardisierte Regeln und Verfahren zur Kommunikation zwischen Geräten in einem Netzwerk. Sie definieren, wie Daten gesendet, empfangen und interpretiert werden.

Wichtige Netzwerkprotokolle:

- **TCP (Transmission Control Protocol) [Port: abhängig vom Dienst]**
  Stellt eine zuverlässige, verbindungsorientierte Datenübertragung sicher. Nutzt Mechanismen wie Fehlerkorrektur und Paket-Reihenfolge.
- **UDP (User Datagram Protocol) [Port: abhängig vom Dienst]**
  Verbindungsloses, schnelles Protokoll ohne Fehlerkorrektur. Geeignet für Echtzeitanwendungen wie Streaming oder VoIP.
- **IP (Internet Protocol) [Kein Port, da es auf Layer 3 arbeitet]**
  Zuständig für die Adressierung und Weiterleitung von Datenpaketen im Netzwerk. IPv4 (32-Bit-Adressen) und IPv6 (128-Bit-Adressen) sind gängige Versionen.
- **HTTP (Hypertext Transfer Protocol) [Port 80]**
  Ermöglicht die Übertragung von Webseiten. HTTP/2 und HTTPS (verschlüsselt mit TLS) bieten verbesserte Sicherheit und Effizienz.
- **HTTPS (Hypertext Transfer Protocol Secure) [Port 443]**
  Verschlüsselte Version von HTTP mit TLS/SSL zur sicheren Datenübertragung.
- **FTP (File Transfer Protocol) [Ports 20 (Datenübertragung) & 21 (Steuerung)]**
  Dient der Dateiübertragung zwischen Client und Server. Unterstützt verschiedene Übertragungsmodi (aktiv/passiv).
- **SFTP (Secure File Transfer Protocol) [Port 22]**
  Sicheres Dateiübertragungsprotokoll, das SSH für Verschlüsselung nutzt.
- **DNS (Domain Name System) [Port 53 (UDP/TCP)]**
  Wandelt menschenlesbare Domainnamen (z. B. www.hstin.io) in IP-Adressen um.

- **SMTP (Simple Mail Transfer Protocol) [Port 25 (alt), 465 (SSL), 587 (TLS)]**
  Verantwortlich für den Versand von E-Mails zwischen Mailservern.
- **IMAP (Internet Message Access Protocol) [Port 143, mit SSL/TLS 993]**
  Ermöglicht den serverbasierten Abruf von E-Mails, sodass sie auf mehreren Geräten synchron bleiben.
- **POP3 (Post Office Protocol 3) [Port 110, mit SSL/TLS 995]**
  Ermöglicht den Abruf von E-Mails, wobei sie nach dem Download standardmäßig vom Server gelöscht werden.
- **DHCP (Dynamic Host Configuration Protocol) [Port 67 (Server), 68 (Client)]**
  Vergibt automatisch IP-Adressen an Geräte in einem Netzwerk.

OSI-Modell

Das OSI-Modell (Open Systems Interconnection Model) ist ein Schichtenmodell zur Standardisierung der Netzwerkkommunikation. Es besteht aus sieben Schichten, die jeweils spezifische Aufgaben übernehmen.

Die 7 Schichten des OSI-Modells:

1. **Bitübertragungsschicht (Physical Layer)**
   - Überträgt rohe Bitströme über physikalische Medien (z. B. Kabel, Funk).
   - Beispiel: Ethernet, WLAN, Glasfaser.
2. **Sicherungsschicht (Data Link Layer)**
   - Stellt fehlerfreie Übertragung zwischen zwei direkt verbundenen Geräten sicher.
   - MAC-Adressen und Switches arbeiten auf dieser Schicht.
   - Beispiel: Ethernet (802.3), WLAN (802.11).
3. **Vermittlungsschicht (Network Layer)**
   - Verantwortlich für die logische Adressierung und das Routing von Paketen.
   - Arbeitet mit IP-Adressen und Routern.
   - Beispiel: IPv4, IPv6, ICMP (Ping).
4. **Transportschicht (Transport Layer)**
   - Sorgt für die zuverlässige oder unzuverlässige Übertragung von Daten zwischen Endgeräten.

    - Wichtige Protokolle: TCP (zuverlässig), UDP (schnell, aber unzuverlässig).
5. **Sitzungsschicht (Session Layer)**
    - Steuert und verwaltet Sitzungen zwischen Anwendungen.
    - Beispiel: NetBIOS, RPC.
6. **Darstellungsschicht (Presentation Layer)**
    - Wandelt Daten in ein für Anwendungen verständliches Format um.
    - Verschlüsselung und Komprimierung erfolgen hier.
    - Beispiel: TLS/SSL, JPEG, ASCII.
7. **Anwendungsschicht (Application Layer)**
    - Stellt Netzwerkdienste für Anwendungen bereit.
    - Beispiel: HTTP, FTP, SMTP, DNS.

2.7 Strukturierte Verkabelung

Strukturierte Verkabelung ist ein standardisiertes Verkabelungssystem für Netzwerke in Gebäuden, das eine flexible, skalierbare und einheitliche Infrastruktur bereitstellt.

Merkmale:

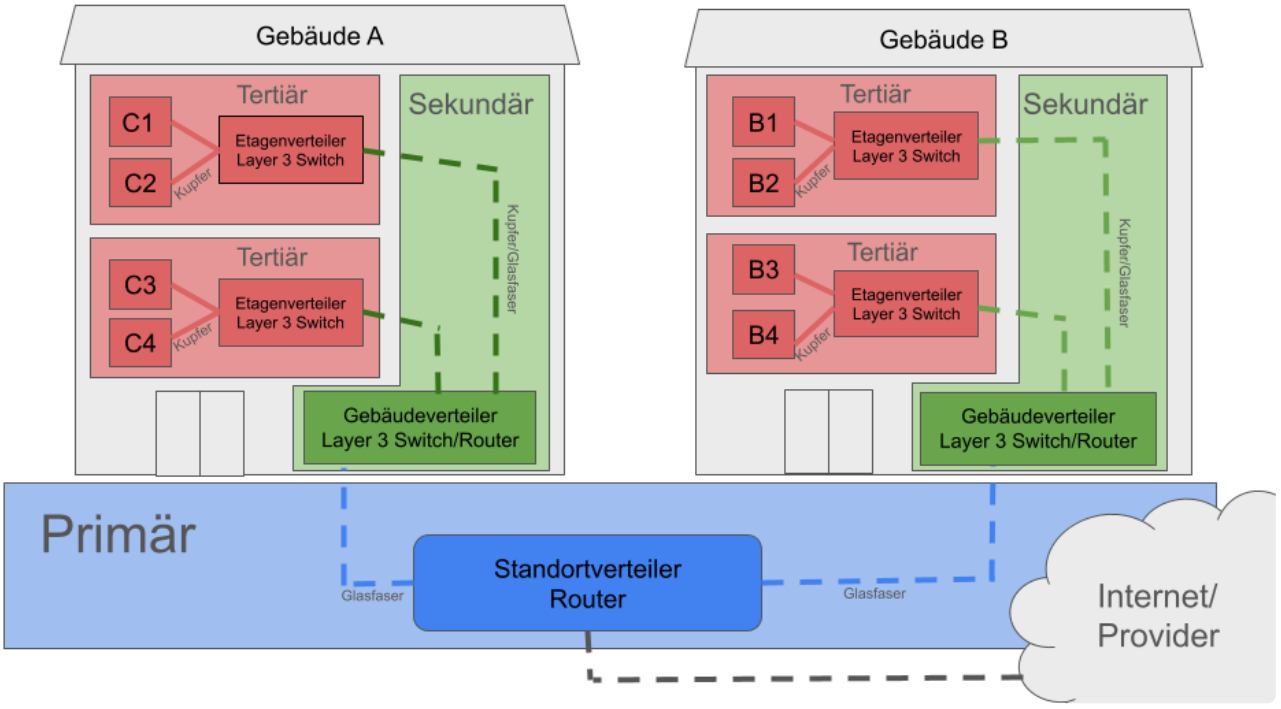
- Modular aufgebaut und hierarchisch strukturiert
- Unabhängig von spezifischen Netzwerktechnologien
- Unterstützt verschiedene Dienste
- Genormt durch Standards

Bestandteile der strukturierten Verkabelung:

1. **Primärbereich (Campus-Verkabelung)**
    - Verbindet verschiedene Gebäude eines Standorts
    - Nutzt Glasfaser oder Kupferkabel
2. **Sekundärbereich (Gebäude-Verkabelung)**
    - Verbindet die Etagen eines Gebäudes mit dem Hauptverteiler
    - Oft Glasfaser oder Kupfer (Cat 6/7)
3. **Tertiärbereich (Etagen-Verkabelung)**
    - Verbindet Etagenverteiler mit den einzelnen Netzwerkdosen
    - Meist Kupferkabel (Twisted-Pair, z. B. Cat 6a, Cat 7)

Vorteile der strukturierten Verkabelung:

- **Zukunftssicherheit:** Unterstützt neue Technologien durch standardisierte Infrastruktur
- **Flexibilität:** Anpassbar an verschiedene Netzwerkarchitekturen
- **Einfache Wartung:** Klare Struktur erleichtert Fehlersuche und Erweiterungen
- **Höhere Übertragungsgeschwindigkeiten:** Optimiert für moderne Netzwerke (Gigabit-Ethernet, PoE)



2.8 USV (Unterbrechungsfreie Stromversorgung)

Eine USV (Unterbrechungsfreie Stromversorgung) ist ein System, das bei Stromausfällen oder -schwankungen die Stromversorgung für angeschlossene Geräte aufrechterhält. Sie schützt vor Datenverlust, Hardware-Schäden und Systemausfällen.

Funktionen einer USV:

- Schutz vor Stromausfällen, Spannungsspitzen und -einbrüchen
- Überbrückung von kurzen Netzausfällen
- Ermöglicht geordnetes Herunterfahren kritischer Systeme
- Filterung von Netzstörungen (z. B. Frequenzschwankungen)

USV-Arten:

1. **Offline-USV (VFD - Voltage and Frequency Dependent)**
   - Schaltet bei Stromausfall auf Batterie um (Umschaltzeit ca. 10 ms)
   - Keine aktive Spannungsregelung im Normalbetrieb
   - Geeignet für einfache Geräte (z. B. PCs, kleine Router)
2. **Line-Interactive-USV (VI - Voltage Independent)**
   - Verfügt über einen Spannungsregler (AVR – Automatic Voltage Regulation)
   - Reduziert Netzschwankungen ohne Umschaltung auf Batterie
   - Geeignet für Server, Netzwerktechnik und kleine Rechenzentren
3. **Online-USV (VFI - Voltage and Frequency Independent / Doppelwandler-USV)**
   - Wandelt Netzstrom in Gleichstrom und wieder zurück in saubere Wechselspannung
   - Keine Umschaltzeit, konstante Spannung und Frequenz
   - Ideal für kritische IT-Systeme, Rechenzentren und medizinische Geräte
4. **Hybrid-USV (Dynamische USV, Rotary UPS)**
   - Kombination aus Online-USV und rotierendem Schwungrad für Energiepuffer
   - Höhere Effizienz und längere Lebensdauer als reine Online-USVs
   - Eingesetzt in großen Rechenzentren oder Industrieanlagen

2.9 Energie

Energie

Energie ist die Fähigkeit eines Systems, Arbeit zu verrichten oder Wärme abzugeben. In der IT und Elektrotechnik spielt sie eine zentrale Rolle für den Betrieb von Geräten und Rechenzentren.

Energieeffizienz

- Beschreibt das Verhältnis von nutzbarer Energie zur eingesetzten Energie.

- Ziel ist es, Energieverluste zu minimieren und Betriebskosten zu senken.
- Maßnahmen: Effiziente Netzteile, Kühlungssysteme, Lastmanagement.

Strom (Elektrische Stromstärke, I)

- Einheit: **Ampere (A)**
- Gibt an, wie viele elektrische Ladungsträger pro Sekunde durch einen Leiter fließen.
- Formel: I = U/R (Stromstärke = Leistung / Spannung).

Spannung (Elektrische Potenzialdifferenz, U)

- Einheit: **Volt (V)**
- Beschreibt die treibende Kraft, die den Strom durch einen Leiter bewegt.
- Typische Spannungen: 230V (Haushalt), 12V/5V (IT-Geräte).
- Formel: U = P / I (Spannung = Leistung / Stromstärke)

Leistung (Elektrische Leistung, P)

- Einheit: **Watt (W)**
- Gibt an, wie viel Energie pro Sekunde umgesetzt wird.
- Formel: P = U x I (Leistung = Spannung × Stromstärke).

Wirkungsgrad (n)

- Verhältnis von abgegebener zu zugeführter Leistung.
- Formel:

$$\eta = \frac{P_{ab}}{P_{zu}} \times 100\%$$

- Hohe Wirkungsgrade reduzieren Energieverluste und Betriebskosten.
- Beispiel: Ein Netzteil mit **80 Plus Gold**-Zertifizierung hat einen Wirkungsgrad von ca. 90%.

2.10 Geräteklassen

Geräteklasse	Beschreibung
Desktops	Stationäre PCs ohne Mobilitätsfokus
Notebooks	Mobile Laptops mit integrierter Hardware
All-in-One	Integrierte Systeme mit Monitor und PC in einem Gehäuse
Thin Clients	Schlanke Clients, die auf zentrale Server zugreifen
Tablets	Touch-basierte mobile Geräte
Smartphones	Kompakte mobile Endgeräte mit Kommunikationsfunktion

2.11 Barrierefreiheit

Barrierefreiheit bezeichnet Maßnahmen zur Ermöglichung der Nutzung von IT-Systemen für Menschen mit Einschränkungen. Zusätzliche Hardware kann dabei helfen, die Bedienung zu erleichtern.

Beispiele für unterstützende Hardware:

- **Größerer Monitor** – Erleichtert das Lesen für Menschen mit Sehschwäche durch größere Darstellung von Inhalten.
- **Breitere Tastatur** – Hilft Menschen mit motorischen Einschränkungen durch größere Tastenabstände und spezielle Layouts.
- **Lautsprecher/Mikrofon** – Unterstützt Sprachsteuerung und Sprachausgabe für Personen mit Seh- oder Mobilitätseinschränkungen.

3. Software

3.1 Softwarearten

Anwendungssoftware

Unter **Anwendungssoftware** versteht man Programme die für bestimmte Anwendungsbereiche entwickelt wurden und spezifische Aufgaben für den Endnutzer erfüllen. Darunter fallen z.B. die Office Programme, Browser, Grafikprogramme oder Spiele

Standardsoftware

Standardsoftware sind Fertige Programme für die Breite Masse wie z.B. Word oder Excel.

Branchensoftware

Branchensoftware sind Programme welche speziell für eine gewisse Branche entwickelt worden sind.

Individualsoftware

Unter **Individualsoftware** fällt alles was speziell für einen Kunden entwickelt worden ist.

Proprietäre Software

Wenn eine Software Closed Source ist, beschränkte Nutzungsrechte hat und zudem noch oft kostenpflichtig ist, handelt es sich um **Proprietäre Software**. Beispiele hierfür sind Windows oder die Adobe Suite.

Open Source

Open Source Software ist Software bei der der Source Code öffentlich zugänglich ist und eingesehen werden kann.

Betriebssysteme

Ein **Betriebssystem** ist die Grundsoftware für Computer. Sie Verwaltet die Hardware und Programme. Die gängigsten **Betriebssysteme** sind Windows, Linux und macOS.

Treiber

Ein **Treiber** ist ein spezielles Programm, das als Übersetzer zwischen Hardware (wie Drucker, Grafikkarte oder Maus) und einem Betriebssystem dient.

3.2 Beurteilungskriterien

- Benutzerfreundlichkeit
- Stabilität/Zuverlässigkeit
- Support/Updates
- Kosten
- Systemanforderungen
- Kompatibilität

3.3 BIOS

Das BIOS (Basic Input/Output System) ist die Firmware eines Computers, die beim Start die Hardware initialisiert und das Betriebssystem lädt. Es ist auf einem nichtflüchtigen Speicherchip (ROM, EEPROM oder Flash-Speicher) auf dem Mainboard abgelegt.

Funktionen des BIOS:

- **POST (Power-On Self-Test):** Überprüft die grundlegende Hardware auf Fehler
- **CMOS-Konfiguration:** Speichert Einstellungen wie Boot-Reihenfolge und Systemzeit
- **Bootloader:** Lädt den Bootsektor des Betriebssystems
- **Hardware-Steuerung:** Verwaltung von Tastatur, Festplatten, Lüftern und anderen Komponenten

BIOS vs. UEFI:

Merkmal	BIOS	UEFI
Speicherort	ROM/EEPROM/Flash	Flash-Speicher
Boot-Modus	MBR (max. 2 TB)	GPT (größere Laufwerke)
Oberfläche	Textbasiert	Grafisch, Maus-Support

Merkmal	BIOS	UEFI
Sicherheit	Keine Secure Boot-Funktion	Secure Boot, TPM-Unterstützung

Wichtige BIOS-Tasten (je nach Hersteller):

- **DEL / F2**: BIOS-Setup aufrufen
- **F12 / ESC**: Boot-Menü öffnen

BIOS wird zunehmend durch **UEFI** ersetzt, das moderner, sicherer und leistungsfähiger ist.

3.4 Enwicklungssoftware

Compiler

Ein **Compiler** ist ein Programm, das Quellcode in Maschinencode übersetzt. Der Compiler überprüft den Quellcode auf Fehler, übersetzt ihn in Assembler-Code und erstellt ein Objektmodul. Der Compiler wird nur einmal während der Kompilierungsphase aufgerufen.

Linker

Ein **Linker** (auch **Linker-Loader** genannt) ist ein Programm, das mehrere Objektmodule zu einem ausführbaren Programm zusammenfügt. Der Linker löst externe Verweise (Libraries) auf und ersetzt sie durch die tatsächlichen Speicheradressen.

Interpreter

Ein **Interpreter** ist ein Programm, das Quellcode direkt ausführt, ohne ihn vorher in Maschinencode zu übersetzen. Der Interpreter liest den Quellcode zeilenweise ein und führt ihn sofort aus. Beispiele für interpretierte Sprachen sind Python, JavaScript und Ruby.

Hybrid-Modelle

Einige Programmiersprachen, wie **Java** und **C#**, verwenden ein Hybrid-Modell, das sowohl Compile- als auch Interpret-Techniken kombiniert. Der Quellcode wird zunächst in einen Zwischencode (Bytecode) kompiliert, der dann von einer virtuellen Maschine (JVM oder CLR) interpretiert wird. Dies ermöglicht eine Plattformunabhängigkeit und eine verbesserte Sicherheit.

Integrierte Entwicklungsumgebung (IDE)

Eine **IDE (Integrated Development Environment)** ist eine Software für Programmierer. Das Programm enthält alles was für die Softwareentwicklung nötig ist: Editor, Compiler, Debugger, etc… Beispiele dafür sind Visual Studio, Eclipse oder die IntelliJ Programme.

Debugger

Ein **Debugger** unterstützt einen Entwickler während des Entwicklungsprozesses. Nachdem das Programm kompiliert und ausgeführt wurde, bietet der Debugger detaillierte Analysen des laufenden Programms. Beispielsweise können Laufzeitvariablen ausgelesen oder das Programm während der Ausführung unterbrochen und zu einem späteren Zeitpunkt fortgesetzt werden.

3.5 Cloud

Cloudlösungen bieten den Zugriff auf IT-Ressourcen über das Internet

Software as a Service (SaaS)

- Nutzung von Anwendungen über das Internet ohne lokale Installation
- Beispiel: Online-Office-Pakete, CRM-Systeme in der Cloud

Desktop as a Service (DaaS)

- Bereitstellung von virtuellen Desktop-Umgebungen
- Ermöglicht den Zugriff auf einen kompletten Desktop über verschiedene Endgeräte
- Zentralisierte Verwaltung und erhöhte Sicherheit

Infrastructure as a Service (IaaS)

- Bereitstellung von IT-Infrastrukturen wie virtuelle Maschinen, Speicher, Netzwerke und Rechenzentren über das Internet.
- Bietet eine Hohe Flexibilität, Skalierbarkeit und keine Notwendigkeit eigener physischer Hardware.

Platform as a Service (PaaS)

- Bereitstellung von Entwicklungsplattformen und -umgebungen, auf denen Entwickler Anwendungen erstellen, testen und bereitstellen können, ohne sich um die zugrunde liegende Infrastruktur kümmern zu müssen.
- Hat den Vorteil der beschleunigte Entwicklung, integrierte Entwicklungs- und Laufzeitumgebungen sowie vereinfachte Verwaltung von Anwendungen.

3.6 Virtuelle Desktops

Virtuelle Desktops ermöglichen das gleichzeitige Arbeiten mit mehreren separaten Desktop-Umgebungen auf einem einzigen physischen Bildschirm.

Vorteile:

- Bessere Organisation von Fenstern und Anwendungen
- Erhöhte Produktivität durch klare Trennung von Aufgaben
- Reduzierte Ablenkung und mehr Übersicht

Beispiele:

- Windows: „Task View“ (Win + Tab)
- macOS: „Spaces“ (Mission Control)
- Linux: Mehrere Workspaces je nach Distribution

3.7 Funktionale, ökonomische und ökologische Aspekte

Die Bewertung von IT-Systemen erfolgt häufig unter Berücksichtigung mehrerer Kriterien:

- **Funktionale Aspekte**
  - **Ergonomie:** Benutzerfreundlichkeit, intuitive Bedienung, Design und Bedienoberfläche
  - **Leistungsparameter:** Geschwindigkeit, Verarbeitungsleistung, Reaktionszeiten, Skalierbarkeit
- **Ökonomische Aspekte**

    - **Einmalige Kosten:** Anschaffungskosten, Investitionsausgaben
    - **Laufende Kosten:** Betriebskosten, Wartung, Updates, Lizenzgebühren
    - **Nutzungsdauer:** Geplante Lebensdauer des Systems, Abschreibungsdauer, Wiederbeschaffungszyklus
- **Ökologische Aspekte**
    - **Energieverbrauch:** Effizienz, Stromverbrauch im Dauerbetrieb
    - **Recyclingfähigkeit:** Umweltgerechte Entsorgung, Wiederverwendbarkeit von Komponenten, CO_2-Fußabdruck

> ⚠ Hinweis!
>
> Bei der Bewertung sollten alle drei Aspekte im Zusammenspiel betrachtet werden, da sie oft miteinander in Konflikt stehen (z. B. hohe Leistung vs. hoher Energieverbrauch).

3.8 Kommunikationssysteme

Moderne Unternehmen nutzen verschiedene Systeme zur internen und externen Kommunikation. **Videokonferenzsysteme** wie Teams oder Zoom ermöglichen ortsunabhängige Meetings und Zusammenarbeit. **Social-Media-Systeme** dienen der Kundenkommunikation und dem Marketing. Dazu gehören auch Messaging-Dienste für schnelle interne Abstimmungen.

3.9 Client-Server

In einem Client-Server-System stellt ein zentraler Server Dienste für mehrere Clients bereit. Die Clients greifen über das Netzwerk auf diese Dienste zu. Typische Beispiele sind:

- **Dateiserver:** Zentrale Datenspeicherung
- **Mailserver:** E-Mail-Verwaltung
- **Datenbankserver:** Zentrale Datenhaltung
- **Webserver:** Bereitstellung von Webseiten

3.10 Domäne

Die **Domäne** ist ein zentrales Verwaltungskonzept für Netzwerke. Ein **Domänencontroller** verwaltet dabei:

- Benutzerkonten und Berechtigungen
- Gruppenrichtlinien
- Gemeinsame Ressourcen
- Zentrale Authentifizierung

4. Installation und Konfiguration

4.1 Hardware

Physische Installation

- Einbau von Komponenten (z.B. CPU, RAM, Festplatten, Grafikkarten)
- Anschluss von Peripheriegeräten (Monitor, Tastatur, Maus, Drucker)
- Verkabelung und Stromversorgung prüfen

Hardware-Konfiguration

- BIOS/UEFI-Einstellungen anpassen (Boot-Reihenfolge, Sicherheitsoptionen)
- Hardware-Diagnose (Fehlersuche bei nicht erkannten Komponenten)
- Firmware-Updates durchführen

4.2 Betriebssystems

Betriebssystem-Installation

- Auswahl des Installationsmediums (DVD, USB-Stick, Netzwerkinstallation)
- Partitionierung der Festplatte
- Auswahl und Installation von Treibern (Hardware-spezifische Software)

Betriebssystem-Konfiguration

- Lokale und Netzwerkeinstellungen (Zeit, Sprache, Benutzerkonten)
- Installation von Sicherheitsupdates und Patches
- Einrichtung von automatischen Update-Prozessen

4.3 Kommandozeile

Grundlagen der Kommandozeile

- Unterschiedliche Shells/Terminals (z.B. cmd, PowerShell, Bash)
- Aufbau einer Befehlszeile: Befehl, Parameter, Optionen, Argumente

Befehlssyntax und Parameter

- Beispiel: `dir` (Windows) oder `ls` (Linux/Mac) zur Anzeige von Dateien
- Nutzung von Parametern zur Anpassung der Ausgabe (z.B. `-l`, `/w`)

Praktische Anwendung

- Navigation im Dateisystem (`cd`, `pwd`)
- Erstellen und Löschen von Dateien/Verzeichnissen (`mkdir`, `del`, `rm`)
- Nutzung von Aliasen zur Vereinfachung von Befehlen (z.B. `alias ll='ls -la'` in Bash)

4.4 Anpassung von Software

Softwarekonfiguration

- Änderung von Konfigurationsdateien (z.B. .conf, .ini, .xml)
- Anpassen von Einstellungen über grafische Benutzeroberflächen oder CLI
- Installation von Software-Paketen und Updates (Paketmanager wie apt, yum, Chocolatey)
- **Anwendungsfälle:**
  - Anpassung an spezifische Anforderungen (z.B. Sicherheit, Performance)
  - Integration in bestehende Systeme

4.5 Konfiguration, Test, Troubleshooting und Dokumentation von

Netzwerkverbindungen

Netzwerkkonfiguration

- **IP-Adressen und Subnetze:**
  - Statische vs. dynamische (DHCP) IP-Zuweisung
  - Manuelle Konfiguration über Betriebssystem-Einstellungen oder CLI-Befehle (z.B. `ipconfig`, `ifconfig`, `ip`)
- **WLAN-Zugang:**
  - Einrichtung von WLAN-Profilen (SSID, Verschlüsselung, Pre Shared Key/Enterprise)
  - Verbindungstest und Signalstärke messen
- **VPN:**
  - Einrichtung und Konfiguration von VPN-Verbindungen
  - Authentifizierungsverfahren und Verschlüsselungsstandards

Test und Troubleshooting

- **Diagnosewerkzeuge:**
  - `ping`: Prüfen der Erreichbarkeit eines Hosts
  - `traceroute`/`tracert`: Ermittlung des Pfades zu einem Ziel
  - `nslookup`: DNS-Abfragen durchführen
  - `arp`: Anzeigen und Ändern der ARP-Tabelle
- **Dokumentation:**
  - Festhalten von Konfigurationseinstellungen und Änderungen
  - Erstellung von Protokollen (Logs) und Fehlerberichten
  - Nutzung von Tools zur Netzwerküberwachung (z.B. Wireshark)

4.6 Konsolenbefehle

Dateioperationen

- **Windows:**
  - `dir`: Anzeigen des Inhalts eines Verzeichnisses
  - `copy`: Kopieren von Dateien
  - `del`: Löschen von Dateien
- **Linux/Mac:**
  - `ls`: Anzeigen des Inhalts eines Verzeichnisses

- `cp`: Kopieren von Dateien
- `rm`: Löschen von Dateien
- `mkdir`: Erstellen von Verzeichnissen
- `chmod`: Ändern von Dateiberechtigungen

Netzwerktroubleshooting

- `ipconfig` (Windows) und `ifconfig`/`ip` (Linux): Anzeigen und Konfigurieren von Netzwerkschnittstellen
- `alias`: Erstellen von Befehlsaliasen (insbesondere in Unix/Linux-Umgebungen)
- `iproute2`: Erweiterte Netzwerkadministration unter Linux
- `arp`: Anzeigen und Verwalten der ARP-Tabelle
- `ping`: Testen der Netzwerkverbindung
- `traceroute`/`tracert`: Verfolgen von Netzwerkrouten
- `nslookup`: Überprüfen der Namensauflösung

5. Lizenzen

5.1 Grundlagen des Urheberschutzes

Der Urheberschutz regelt die Rechte von Schöpfern geistiger Werke und schützt diese vor unbefugter Nutzung.

- **Gesetzliche Grundlage:** In Deutschland im **Urheberrechtsgesetz (UrhG)** geregelt.
- **Geschützte Werke:** Texte, Musik, Bilder, Software, Filme, wissenschaftliche Werke etc.
- **Entstehung:** Automatisch mit der Schöpfung des Werks, keine Registrierung erforderlich.
- **Dauer:** In der Regel **70 Jahre nach dem Tod** des Urhebers.
- **Rechte des Urhebers:**
  - **Urheberpersönlichkeitsrechte:** Schutz vor Entstellung, Namensnennung
  - **Verwertungsrechte:** Vervielfältigung, Verbreitung, öffentliche Wiedergabe
  - **Nutzungsrechte:** Können anderen durch Lizenzen eingeräumt werden
- **Einschränkungen:**
  - **Privatkopie:** Erlaubt, sofern keine Kopierschutzmaßnahmen umgangen werden
  - **Zitate:** Zulässig bei Angabe der Quelle
  - **Schrankenregelungen:** Bildung, Wissenschaft, Berichterstattung

> ⚠ Info
>
> **Verstöße gegen das Urheberrecht** können Abmahnungen, Schadensersatz oder strafrechtliche Folgen nach sich ziehen.

5.2 Lizenzarten

Creative-Commons (CC)

Einführung

Creative Commons (CC) ist eine Non-Profit-Organisation, die es Urhebern ermöglicht, ihre Werke unter bestimmten Bedingungen frei zu geben. Die Creative-Commons-Lizenzen bieten eine Alternative zu den traditionellen Urheberrechten und ermöglichen es anderen, Werke zu teilen, zu bearbeiten und weiterzugeben.

Attribution (BY)

Eine **Attribution-Lizenz** (BY) erlaubt die Verwendung, Veränderung und Weitergabe von Werken, solange die Urheberin oder der Urheber genannt wird.

ShareAlike (SA)

Eine **ShareAlike-Lizenz** (SA) erlaubt die Verwendung, Veränderung und Weitergabe von Werken, solange die neue Arbeit unter denselben Bedingungen lizenziert wird.

NoDerivatives (ND)

Eine **NoDerivatives-Lizenz** (ND) erlaubt die Verwendung und Weitergabe von Werken, aber nicht die Veränderung oder Bearbeitung.

NonCommercial (NC)

Eine **NonCommercial-Lizenz** (NC) erlaubt die Verwendung, Veränderung und Weitergabe von Werken, solange keine kommerzielle Nutzung erfolgt.

Zero (CC0)

Eine **CC0-Lizenz** (Zero) dediziert ein Werk dem öffentlichen Domain, so dass es frei von Urheberrechten ist und ohne Einschränkungen verwendet werden kann.

Kombination von Lizenzen

Die Creative-Commons-Lizenzen können kombiniert werden, um genau anzugeben, wie ein Werk verwendet werden darf. Zum Beispiel: **CC BY-SA** erlaubt die Verwendung, Veränderung und Weitergabe von

Werken, solange die Urheberin oder der Urheber genannt wird und die neue Arbeit unter denselben Bedingungen lizenziert wird.

Weitere Lizenzen

Nodelocked License

Eine **Nodelocked-Lizenz** ist eine Lizenz, die an eine bestimmte Node (Client/Server/Computer) gebunden ist, so dass die Software nur auf diesem bestimmten Knoten verwendet werden kann.

Named User License

Eine **Benutzerlizenz**, oder **NUL**, ist eine Lizenz, die einer bestimmten Person zugewiesen wird, und diesem damit Zugriff auf die Software ermöglicht, unabhängig vom verwendeten Gerät. Diese Person muss Namentlich in der Lizenz erwähnt werden.

Floating License / Concurrent Use License

Eine **Floating-License**, auch bekannt als **Concurrent-Use-License**, ist eine Lizenz, die eine bestimmte Anzahl von Benutzern gleichzeitig Zugriff auf die Software ermöglicht, aber nicht die Gesamtzahl der Benutzer beschränkt, die die Software verwenden können.

Pay per Use (Metered Service)

Eine **Pay-per-Use**-, oder **Metered-Service**-Lizenz, ist ein Lizenzmodell, bei dem Benutzer nur für die tatsächliche Nutzung der Software bezahlen, anstatt eine feste Gebühr für eine Lizenz zu zahlen.

6. Wirtschaft

6.1 Kosten

Kosten sind finanzielle Aufwendungen, die im Rahmen betrieblicher Tätigkeiten entstehen. Sie lassen sich in verschiedene Kategorien unterteilen.

Anschaffungskosten

Anschaffungskosten sind die einmaligen Kosten für den Erwerb eines Gutes oder einer Dienstleistung. Dazu gehören:

- Kaufpreis
- Transportkosten
- Installationskosten
- Schulungskosten

Betriebskosten

Betriebskosten sind laufende Kosten für den Betrieb eines Gutes oder einer Dienstleistung. Beispiele:

- Strom, Wasser, Heizung
- Wartung und Reparaturen
- Lizenzen und Abonnements
- Personalkosten

Variable und fixe Kosten

- **Variable Kosten:** Abhängig von der Nutzung oder Produktion (z. B. Materialkosten, Stromverbrauch).
- **Fixe Kosten:** Unabhängig von der Nutzung oder Produktion (z. B. Miete, Gehälter, Versicherungen).

Finanzierungskosten

Finanzierungskosten entstehen durch die Bereitstellung von Kapital für Investitionen. Beispiele:

- Zinsen für Kredite

- Gebühren für Finanzierungsmodelle
- Kosten für Bürgschaften oder Sicherheiten

Kostenvergleich: Leasing, Kauf, Finanzierung, Pay-per-Use

Modell	Vorteile	Nachteile
Leasing	Planbare Raten, keine hohe Anfangsinvestition	Keine Eigentumsrechte, langfristige Kosten
Kauf	Eigentum, keine laufenden Verpflichtungen	Hohe Anfangsinvestition, Wertverlust
Finanzierung	Eigentum nach Abzahlung, flexiblere Zahlung	Zinsen und Gebühren, Kreditabhängigkeit
Pay-per-Use	Kosten nur bei tatsächlicher Nutzung	Unvorhersehbare Kosten, langfristig teurer

6.2 Qualitativer und quantitativer Angebotsvergleich und -bewertung

IT-Systeme werden nach quantitativen (Preis, Leistung) und qualitativen Aspekten (Benutzerfreundlichkeit, Service) verglichen. Eine strukturierte Bewertung erfolgt durch die Nutzwertanalyse.

6.3 Nutzwertanalyse

Die **Nutzwertanalyse** ist eine Methode zur Entscheidungsfindung, die sowohl qualitative als auch quantitative Faktoren berücksichtigt. Sie wird verwendet, um verschiedene Optionen zu bewerten und die beste Alternative auszuwählen, indem unterschiedliche Kriterien systematisch gewichtet und bewertet werden.

Beispiel:

		Anbieter A		Anbieter B		Anbieter C	
Kriterium	**Gewichtung**	Bewertung (ungewichtet)	Bewertung (gewichtet)	Bewertung (ungewichtet)	Bewertung (gewichtet)	Bewertung (ungewichtet)	Bewertung (gewichtet)
Referenzen	0,15 / 15 %	4	0,6	5	0,75	2	0,3
Kosten	0,15 / 15 %	3	0,45	4	0,6	3	0,45
Erreichbarkeit	0,2 / 20 %	4	0,8	3	0,6	5	1,0
Qualität	0,5 / 50 %	2	1,0	2	1,0	4	2,0
SUMME		13	2,85	14	2,95	14	3,75

6.4 Wertschöpfung

IT-Systeme schaffen Wert durch direkte Effizienzsteigerungen und indirekte Vorteile wie Qualitätsverbesserungen. Zusätzlich entstehen strategische Vorteile durch neue Geschäftsmöglichkeiten. Eine reine Kostenbetrachtung reicht daher für die Bewertung nicht aus.

6.5 Projekte

Machbarkeitsanalyse mit Budgetvorgabe

Machbarkeitsanalyse bewertet, ob ein Projekt technisch, wirtschaftlich und rechtlich umsetzbar ist. Aspekte:

- Technische Machbarkeit: Verfügbarkeit von Technologien
- Wirtschaftliche Machbarkeit: Rentabilität und Kosten-Nutzen-Analyse
- Rechtliche Machbarkeit: Einhaltung gesetzlicher Vorgaben

Eine **Budgetvorgabe** stellt sicher, dass das Projekt innerhalb der finanziellen Rahmenbedingungen bleibt. Es umfasst:

- Kosten für Personal, Material und Infrastruktur
- Reserve für unvorhergesehene Ausgaben

Vor- und Nachkalkulation

- **Vorkalkulation:**
  - Schätzung der anfallenden Kosten vor Projektstart
  - Dient der Budgetplanung und Entscheidungsgrundlage
  - Berücksichtigt alle relevanten Kostenfaktoren (Material, Personal, externe Dienstleistungen(!!!))
- **Nachkalkulation:**

    - Erfassung und Analyse der tatsächlichen Kosten nach Projektabschluss
    - Vergleich mit der Vorkalkulation, um Abweichungen zu identifizieren
    - Ableitung von Optimierungspotenzialen für zukünftige Projekte

Einfluss der Stakeholder beurteilen können

Stakeholder sind alle Personen oder Gruppen, die ein Interesse am Projekt haben. Wichtige Stakeholder sind:

- Auftraggeber
- Kunden
- Projektteam
- Lieferanten
- Behörden

Die Beurteilung erfolgt durch:

- Stakeholder-Analyse (Identifikation und Priorisierung)
- Kommunikationsstrategie zur Einbindung relevanter Akteure
- Konfliktmanagement zur Minimierung negativer Einflüsse

Risikoanalyse

- Definition und Identifikation potenzieller Projektrisiken:
    - Technische Risiken (z.B. unvorhergesehene technische Herausforderungen)
    - Wirtschaftliche Risiken (z.B. Budgetüberschreitungen, Marktveränderungen)
    - Organisatorische Risiken (z.B. Ressourcenknappheit, Zeitverzögerungen)
- Bewertung der Risiken hinsichtlich Eintrittswahrscheinlichkeit und Auswirkung
- Entwicklung von Maßnahmen zur Risikovermeidung, -minimierung oder -akzeptanz
- Kontinuierliche Überwachung und Anpassung der Risikomanagementmaßnahmen während des Projektverlaufs

ⓘ Eine gründliche Machbarkeits- und Wirtschaftlichkeitsanalyse bildet die Grundlage für fundierte Projektentscheidungen und ermöglicht die frühzeitige Erkennung von Problemen.

6.6 Markformen

Monopol

Ein **Monopol** ist eine Marktform, bei der nur **ein** Anbieter auf dem Markt tätig ist. Dieser Anbieter hat die vollständige Kontrolle über den Markt und kann die Preise und Mengen selbst bestimmen.

Beispiel: Ein Stromversorger, der der einzige Anbieter in einer Region ist.

Oligopol

Ein **Oligopol** ist eine Marktform, bei der nur wenige **Anbieter** auf dem Markt tätig sind. Diese Anbieter haben eine starke Marktmacht und können die Preise und Mengen beeinflussen.

Beispiel: Die Automobilindustrie, in der nur wenige große Hersteller wie Volkswagen, Toyota und General Motors tätig sind.

Polypol

Ein **Polypol** ist eine Marktform, bei der **viele** Anbieter auf dem Markt tätig sind. Kein einzelner Anbieter hat eine dominante Marktmacht, und die Preise und Mengen werden durch den Marktmechanismus bestimmt.

Beispiel: Der Markt für Lebensmittel, auf dem viele verschiedene Anbieter wie Supermärkte, Discounter und Einzelhändler tätig sind.

Käufer- und Verkaufsmarkt

Ein **Käufermarkt** ist eine Marktform, bei der die Nachfrage nach einem Produkt oder einer Dienstleistung hoch ist und die Anbieter versuchen, Käufer zu gewinnen. Ein **Verkäufermarkt** ist eine Marktform, bei der das Angebot an einem Produkt oder einer Dienstleistung hoch ist und die Käufer sich um die verfügbaren Angebote bemühen.

Beispiel: Ein Käufermarkt für Wohnungen in einer beliebten Stadt, wo viele Käufer um wenige Wohnungen konkurrieren. Ein Verkaufsmarkt für Gebrauchtwagen, wo viele Verkäufer um wenige

Käufer konkurrieren.

6.7 Vertriebsformen

Direkter Vertrieb

Vom **direktem Vertrieb** spricht man, wenn ein Hersteller direkt an den Endkunden Verkauft, ohne einen Zwischenhändler.

Indirekter Vertrieb

Indirekter Vertrieb ist, wenn der das Produkt erst an einen oder mehrere Zwischenhändler vertrieben wird, und erst nach `n` Verschiedenen Zwischenstopps bei dem Endkunden ankommt.

6.8 Zielgruppendefinition und -abgrenzung

Die **Zielgruppendefinition und Abgrenzung** ist der Prozess, bei dem ein Unternehmen seine Zielgruppe identifiziert und von anderen Gruppen abgrenzt. Dies hilft dem Unternehmen, seine Marketingstrategie auf die Bedürfnisse und Präferenzen der Zielgruppe auszurichten.

6.9 Verträge

Verträge sind rechtsverbindliche Vereinbarungen zwischen zwei oder mehr Parteien, die gegenseitige Rechte und Pflichten regeln.

Kaufvertrag

Kaufvertrag regelt den entgeltlichen Erwerb von Waren oder Dienstleistungen.

- Pflichten des Verkäufers: Lieferung, mangelfreie Ware
- Pflichten des Käufers: Zahlung des Kaufpreises

Mietvertrag

Mietvertrag ermöglicht die Nutzung einer Sache gegen eine regelmäßige Mietzahlung.

- Vermieter: Bereitstellung und Instandhaltung
- Mieter: Zahlung und sorgfältiger Gebrauch

Leasing

Leasing ist eine spezielle Mietform, oft für Fahrzeuge oder Maschinen.

- Leasingnehmer: Nutzung gegen Leasingrate
- Leasinggeber: Eigentümer, trägt meist Restwertrisiko

Lizenzvertrag

Lizenzvertrag erlaubt die Nutzung von Rechten (z. B. Software, Patente) gegen Gebühr.

- Lizenzgeber: Erteilt Nutzungsrechte
- Lizenznehmer: Hält sich an Lizenzbedingungen

Werkvertrag

Werkvertrag verpflichtet zur Herstellung eines konkreten Werks (z. B. Bau, Reparatur).

- Auftragnehmer: Herstellung des Werks
- Auftraggeber: Zahlung nach Abnahme

Werkliefervertrag

Werkliefervertrag kombiniert Kauf- und Werkvertrag, Herstellung und Lieferung eines Werkes.

- Beispiel: Anfertigung von Möbeln nach Maß

Dienstvertrag

Dienstvertrag regelt die Erbringung von Dienstleistungen ohne Erfolgsgarantie.

- Beispiel: Arbeitsvertrag, Beratungsleistungen

Servicevertrag

Servicevertrag sichert regelmäßige Wartung oder Supportleistungen zu.

- Beispiel: IT-Supportvertrag, Wartungsverträge

6.10 Vertragsbestandteile

- **Leistungsbeschreibung:** Detaillierte Beschreibung der geschuldeten Leistung
- **Termine:** Festlegung von Fristen und Lieferzeiten
- **Entgelte:** Vereinbarte Vergütung für die Leistungserbringung
- **Sanktionen/Konventionalstrafen:** Vertragsstrafen bei Nichterfüllung oder Verzug

6.11 Vertragsstörungen

Vertragsstörungen treten auf, wenn eine Partei ihre vertraglichen Verpflichtungen nicht erfüllt.

Arten der Vertragsstörung

- **Leistungsverzug:** Verspätete Erfüllung der vertraglichen Leistung
- **Schlechtleistung:** Erfüllung der Leistung mit Mängeln
- **Nichterfüllung:** Vollständiges Ausbleiben der Leistung
- **Störung der Geschäftsgrundlage:** Unerwartete Änderungen der Rahmenbedingungen

6.12 Zielsetzungen

Zielsetzungen von Verträgen können verschiedene Schwerpunkte haben:

- **Ökonomisch:** Wirtschaftliche Vorteile, Kostenoptimierung, Gewinnmaximierung
- **Ökologisch:** Nachhaltigkeit, Ressourcenschonung, Umweltfreundlichkeit
- **Sozial:** Arbeitnehmerrechte, Fairness, gesellschaftliche Verantwortung

6.13 Aufbauorganisation

Aufbauorganisation beschreibt die Struktur eines Unternehmens und die Verteilung von Aufgaben, Kompetenzen und Verantwortlichkeiten.

Mehrliniensystem

Mehrliniensystem ist eine Organisationsform, in der Mitarbeiter mehreren Vorgesetzten unterstellt sind.

- Vorteile: Spezialisierung, kürzere Kommunikationswege
- Nachteile: Unklare Verantwortlichkeiten, Konfliktpotenzial

Einliniensystem

Einliniensystem ist eine hierarchische Organisationsform, in der jeder Mitarbeiter genau einen direkten Vorgesetzten hat.

- Vorteile: Klare Zuständigkeiten, einfache Kontrolle
- Nachteile: Lange Entscheidungswege, hohe Belastung der Führungskräfte

Stabliniensystem

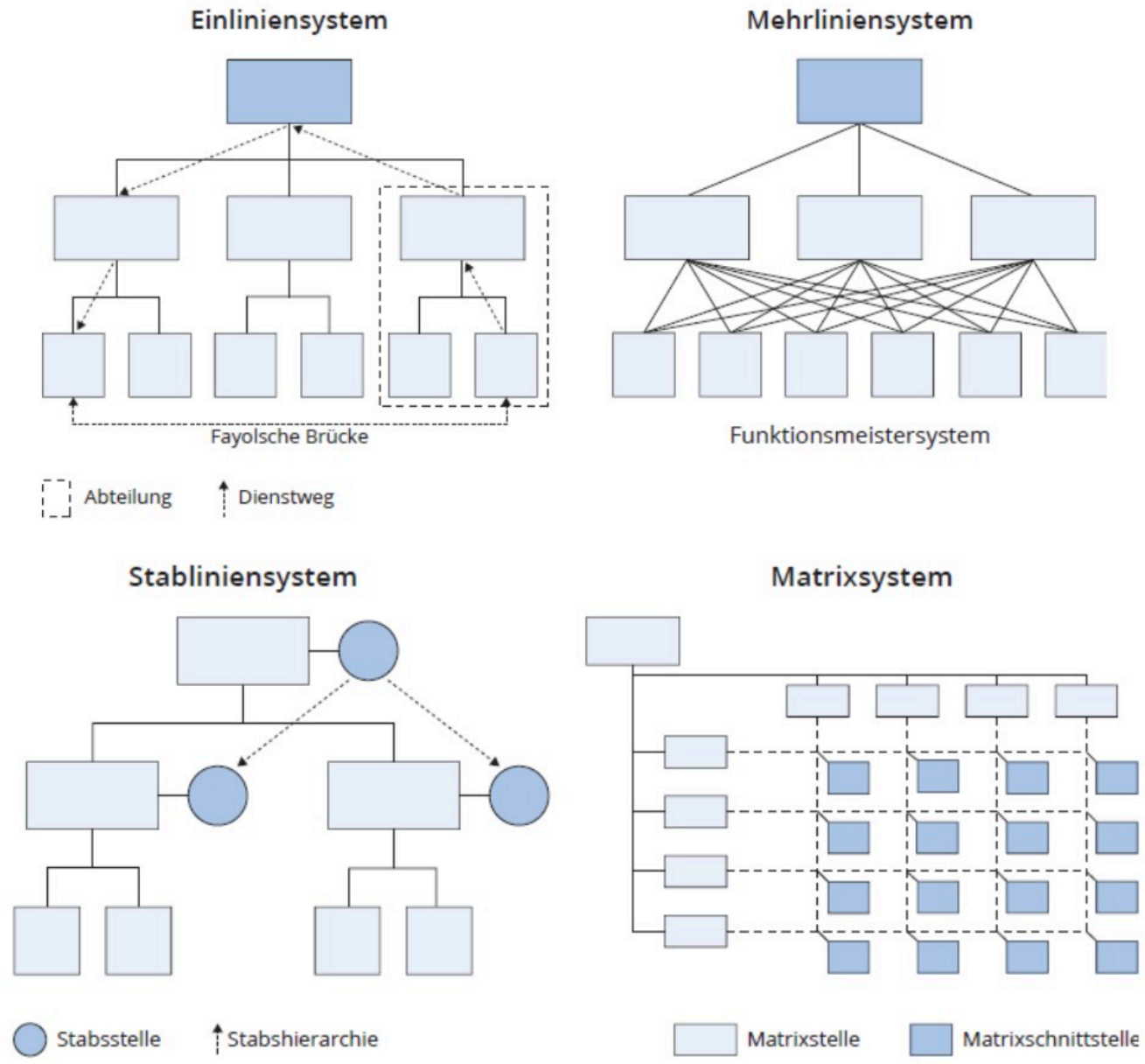
Stabliniensystem erweitert das Einliniensystem durch Stabsstellen, die beratende Funktionen übernehmen, jedoch keine Weisungsbefugnis haben.

- Vorteile: Entlastung der Führungskräfte, bessere Entscheidungsgrundlage
- Nachteile: Höhere Kosten, mögliche Interessenkonflikte zwischen Linie und Stab

Matrixorganisation

Matrixorganisation kombiniert Elemente aus dem Mehr- und Einliniensystem, indem Mitarbeiter nach Funktion und Projektzugehörigkeit organisiert werden.

- Vorteile: Hohe Flexibilität, bessere Ressourcennutzung
- Nachteile: Hoher Kommunikationsaufwand, Konflikte zwischen Linien- und Projektverantwortlichen



6.14 Handlungs- und Entscheidungsspielräume und -vollmachten

Handlungs- und Entscheidungsspielräume definieren die Befugnisse, die Mitarbeiter und Führungskräfte innerhalb einer Organisation besitzen.

- **Handlungsvollmacht:** Befugnis zur Durchführung bestimmter Geschäfte im Rahmen der Unternehmensrichtlinien
- **Einzelvollmacht:** Berechtigung zur Durchführung einer einmaligen Handlung
- **Gesamtvollmacht:** Mehrere Personen sind gemeinsam zur Entscheidung befugt
- **Prokura:** Umfassende geschäftliche Vertretungsbefugnis für kaufmännische Tätigkeiten

7. Projektmanagement

7.1 Merkmale

- Einmaligkeit
- Zielgerichtet
- Zeitlich begrenzt
- Begrenzte Ressourcen
- Komplexität

7.2 Projektplanung

Netzplan

Der **Netzplan** ist eine grafische Methode zur Darstellung von Projektabläufen und deren zeitlichen Abhängigkeiten.

Zentrale Elemente:

1. Vorgänge
2. Pfeile/Kanten (zeigen Abhängigkeiten)
3. Knoten (Ereignisse)

Kürzel	Bedeutung
Vorgang	Vorgangs-ID (A, B, C …)
Dauer	Dauer in Arbeitstagen
FAZ	Frühester Anfangszeitpunkt
FEZ	Frühester Endzeitpunkt
SAZ	Spätester Anfangszeitpunkt
SEZ	Spätester Endzeitpunkt
GP	Gesamtpuffer, GP = SAZ - FAZ oder GP = SEZ - FEZ

Kürzel	Bedeutung
FP	Freier Puffer, FP = FAZ des Nachfolgers - FEZ des Vorgangs

Der **kritische Weg** stellt den längsten Pfad durch den Netzplan dar und bestimmt somit die minimale Projektdauer. (Der weg ohne Puffer)

Ereignisgesteuerte Prozesskette (EPK)

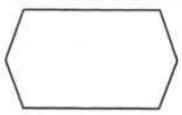
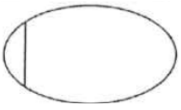
Die **Ereignisgesteuerte Prozesskette (EPK)** ist eine grafische Modellierungssprache zur Darstellung von Geschäftsprozessen. Sie besteht aus einer Aneinanderreihung von Knoten, die durch Kanten verbunden sind.

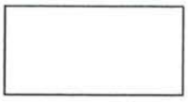
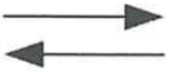
Grundelemente der EPK:

- Ereignisse
- Funktionen
- Verknüpfungsoperatoren (AND, OR, XOR)

Zentrale Regeln:

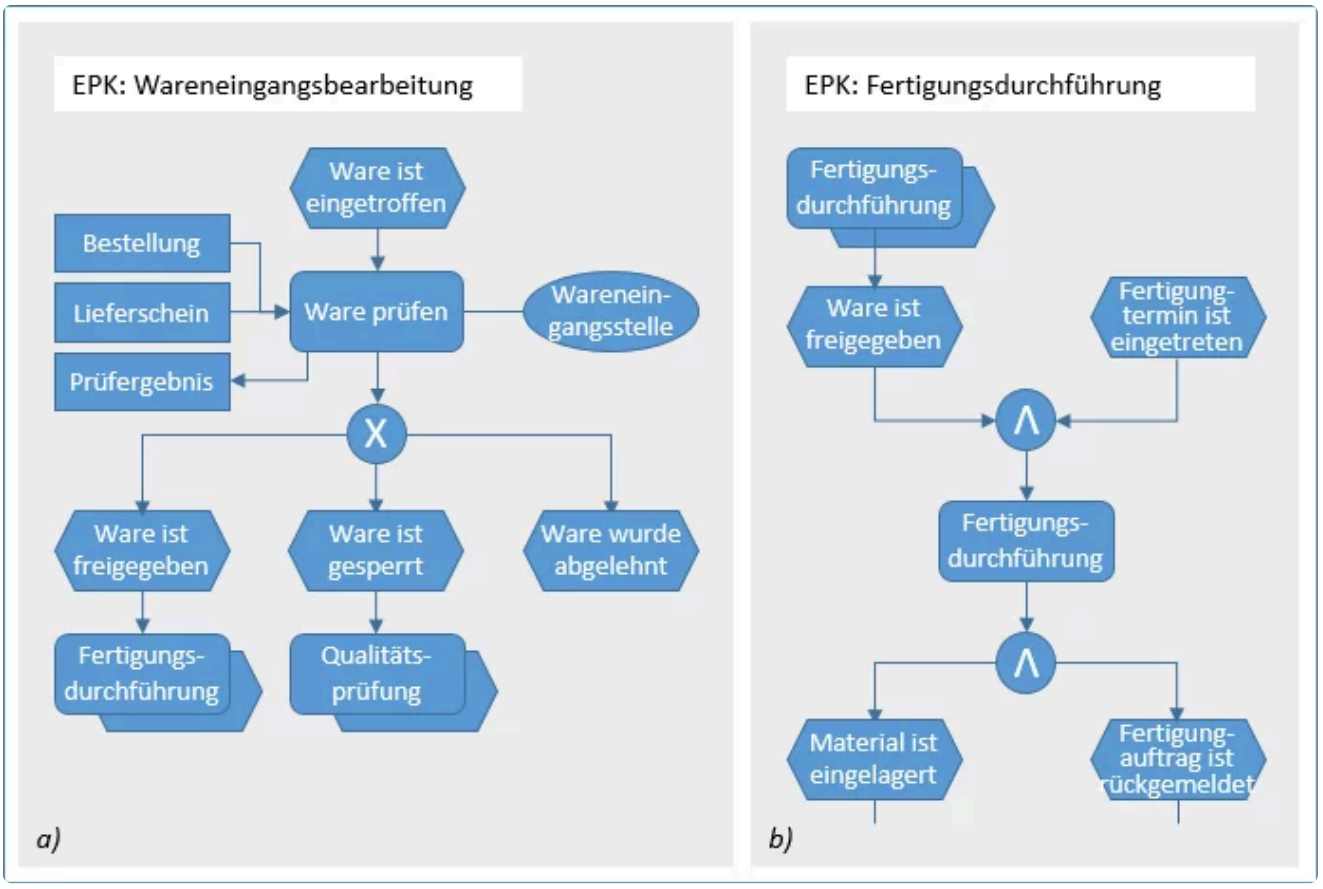
- Ereignisse und Funktionen wechseln sich immer ab.
- Die EPK startet mit einem Ereignis und endet mit einem Ereignis.
- Kanten verbinden die Knoten und zeigen die Flussrichtung.
- Verknüpfungsoperatoren steuern den Prozessfluss.

Symbol	Beschreibung
	**Ereignis** Eingetretener Zustand, der auf den weiteren Ablauf festlegt.
	**Funktion** Betrieblicher Vorgang, der einen Eingangszustand in einen Zielzustand umwandelt.
	**Organisationseinheit** Benennt die Abteilung, der eine Funktion oder ein Ereignis zugeordnet ist.

Symbol	Beschreibung
	**Objekt** Ein Objekt kann ein Informationsobjekt, ein Material, eine Ressource oder ein Produkt sein.
	**Kontrollfluss** Gerichteter, zusammenhängender Graph, dessen Knoten Ereignisse, Funktionen und Verknüpfungsoperatoren sind.
	**Informationsfluss** Gibt an, dass anlässlich einer Funktion Informationen von einem Informationsobjekt gelesen oder auf ein Informationsobjekt geschrieben werden. **ODER** **Material-/Ressourcenfluss** Gibt an, dass in einer Funktion Material bzw. Ressourcen verbraucht werden.
	Zuordnung Gibt an, welche Organisationseinheit oder welche Objekte einer Funktion zugeordnet sind.
	**Konnektor "UND"** Eine Funktion wird ausgeführt, wenn mehrere Ereignisse eingetreten sind. **ODER** Nach einer Funktion treten mehrere Ereignisse ein. **ODER** Ein Ereignis tritt ein, nachdem alle direkt vorangestellten Funktionen ausgeführt wurden.

Symbol	Beschreibung
	**Konnektor "Exklusives Oder" (XOR)** Eine Funktion wird ausgeführt, wenn genau ein Ereignis von mehreren eingetreten ist. **ODER** Nach einer Funktion tritt genau eins von mehreren Ereignissen ein. **ODER** Ein Ereignis tritt ein, nachdem eine von mehreren direkt vorangestellten Funktionen ausgeführt wurde.
	**Konnektor "Offenes Oder" (OR)** Eine Funktion wird ausgeführt, wenn mindestens ein Ereignis von mehreren eingetreten ist. **ODER** Nach einer Funktion tritt mindestens eins von mehreren Ereignissen ein. **ODER** Ein Ereignis tritt ein, nachdem mindestens eine von mehreren direkt vorangestellten Funktionen ausgeführt wurden.

Beispiel:



7.3 SMART

Mit dem **SMART-Prinzip** werden Ziele auf klare und konkrete Formulierungen überprüft. Die Abkürzung steht für:

- **S**-pezifisch
- **M**-essbar
- **A**-ttraktiv (oder Achievable)
- **R**-ealistisch
- **T**-erminiert

7.4 Projektphasen

Grobe Projektphasen

1. Projektdefinition
2. Projektplanung
3. Projektrealisierung
4. Projektabschluss

Wasserfallmodell

Das **Wasserfallmodell** ist ein **lineares** Vorgehensmodell, welches Entwicklungsprozesse in aufeinanderfolgende Projektphasen unterteilt. Hier wird jede Phase nur **einmal** durchlaufen:

1. Analyse
2. Design
3. Implementierung
4. Test
5. Betrieb

Obwohl die Phasen theoretisch nur einmal durchlaufen werden, gibt es in der Praxis oft Sprünge zwischen den Phasen, wenn Probleme oder Änderungsbedarf auftreten. Wird zurückgesprungen, muss man von diesem Punkt aus die Phasen wieder linear durchlaufen.

SCRUM

SCRUM ist ein **agiles** Vorgehensmodell, das iterative und inkrementelle Entwicklungen ermöglicht.

Zentrale Elemente:

- **Sprints:** Kurze, fest definierte Arbeitszyklen (2–4 Wochen), in denen ein Teilprodukt erstellt wird.
- **Rollen:**
  1. **Product Owner:** Legt die Produktvision fest und priorisiert die Anforderungen.
  2. **Scrum Master:** Unterstützt das Team, beseitigt Hindernisse und sorgt für die Einhaltung der SCRUM-Regeln.
  3. **Entwicklungsteam:** Setzt die Aufgaben selbstorganisiert um.
- **Meetings:**
  - **Sprint Planning:** Planung des kommenden Sprints.
  - **Daily Stand-up:** Kurzes tägliches Meeting zum Informationsaustausch und zur Abstimmung.
  - **Sprint Review:** Vorstellung und Bewertung der Arbeitsergebnisse am Ende eines Sprints.
  - **Sprint Retrospective:** Rückblick und kontinuierliche Verbesserung der Arbeitsprozesse.

7.5 Teambildungs-Phasen

Die Teambildung verläuft in mehreren Phasen, die das Zusammenwachsen und die Effizienz des Teams fördern:

1. **Forming:** Kennenlernen und erste Orientierung.
2. **Storming:** Erste Konflikte und Auseinandersetzungen, da Rollen und Verantwortlichkeiten geklärt werden.
3. **Norming:** Entwicklung gemeinsamer Regeln und Normen.
4. **Performing:** Effektive, zielgerichtete Zusammenarbeit.

7.6 Reflektionsmethoden

Lessons Learned

Unter **Lessons Learned** versteht man die nachträgliche Analyse des Projektes, um aus den gemachten Erfahrungen und Ereignissen systematisch zu lernen und diese Erkenntnisse für zukünftige Projekte nutzbar zu machen.

Feedback-Kultur

Eine **Feedback-Kultur** bedeutet, dass Rückmeldungen regelmäßig und offen gegeben werden. Wichtig ist, dass das Feedback ehrlich, respektvoll und hilfreich ist. So können Teams aus Fehlern lernen, sich verbessern und besser zusammenarbeiten.

7.7 Termine

Fristgerechte Terminierung

Fristgerechte Terminierung zielt darauf ab, alle Projektaufgaben rechtzeitig abzuschließen.

Lösungsmöglichkeiten bei Terminproblemen

Bei Terminproblemen können folgende Maßnahmen ergriffen werden:

- **Prioritäten anpassen**
- **Ressourceneinsatz optimieren:** Zusätzliche Mitarbeiter oder externe Unterstützung hinzuziehen.
- **Prozessoptimierung:** Analyse und Verbesserung von Arbeitsabläufen, um Engpässe zu beseitigen.
- **Umfang reduzieren**
- **Kommunikation intensivieren**

Termin und Erfüllungsort

Der **Termin** legt fest, bis wann eine Leistung oder Aufgabe erbracht werden muss. Der **Erfüllungsort** bestimmt den Ort, an dem die Leistung vertragsgemäß zu erbringen ist. Beide Aspekte sind in Verträgen oder Projektplänen geregelt und beeinflussen die rechtzeitige Lieferung sowie mögliche Haftungsfragen.

7.8 Bedarfsanalyse

Eine Bedarfsanalyse ermittelt den tatsächlichen Bedarf und Anforderungen als Basis für Entscheidungen und Projektplanung **Vorgehen:**

- **Erhebung:** Daten sammeln (z.B. Interviews, Umfragen)
- **Analyse:** Anforderungen bewerten und priorisieren
- **Dokumentation:** Ergebnisse festhalten und mit relevanten Stakeholdern abstimmen

7.9 Lasten- und Pflichtenheft

Lastenheft

Ein Lastenheft ist ein Dokument des Auftraggebers, welches dieser alleine, oder in Zusammenarbeit mit dem Auftragnehmer erstellt. Es stellt die Anforderungen und Wünsche des Kunden dar und enthält technische und nicht-technische Bestandteile aus der Sicht des Kunden. Das Lastenheft ist zudem oft die Grundlage eines Werkvertrags.

Pflichtenheft

Im Gegensatz zum Lastenheft ist das Pflichtenheft ein Dokument, welches der Auftragnehmer aufgrund des Lastenhefts erstellt. Es stellt die konkrete technische Lösung dar mit detaillierten Spezifikationen.

7.10 Kundenvorgaben bei der Leistungserbringung

Kundenvorgaben sind spezifische Anforderungen, die ein Kunde an ein Produkt oder eine Dienstleistung stellt. Sie beeinflussen maßgeblich die Projektplanung, -durchführung und -kontrolle.

Arten von Kundenvorgaben:

- **Funktionale Anforderungen**: Beschreiben die gewünschten Funktionen und Eigenschaften des Produkts.
- **Nicht-funktionale Anforderungen**: Definieren Qualitätsmerkmale wie Leistung, Sicherheit oder Skalierbarkeit.
- **Technische Vorgaben**: Spezifikationen zu Hardware, Software oder Schnittstellen.
- **Rechtliche & regulatorische Vorgaben**: Compliance-Anforderungen, Datenschutz (z. B. DSGVO), Zertifizierungen.
- **Budget- und Zeitvorgaben**: Festgelegte Kostenrahmen und Liefertermine.

Bedeutung für das Projektmanagement:

- **Anforderungsanalyse**: Klärung und Dokumentation der Vorgaben zu Beginn des Projekts.
- **Pflichtenheft**: Übersetzung der Kundenanforderungen in technische und organisatorische Spezifikationen.
- **Risikomanagement**: Identifikation und Bewertung von Risiken im Zusammenhang mit den Vorgaben.
- **Qualitätsmanagement**: Sicherstellung, dass die Anforderungen erfüllt werden (z. B. durch Tests, Reviews).
- **Change Management**: Umgang mit nachträglichen Änderungen der Kundenvorgaben.

Herausforderungen & Lösungsansätze:

Herausforderung	Lösungsansatz
Unklare oder unvollständige Anforderungen	Frühzeitige Klärung, Workshops mit dem Kunden
Widersprüchliche Vorgaben	Abstimmung und Priorisierung mit dem Kunden
Änderungen während des Projekts	Flexibles Change Management, dokumentierte Anpassungen
Enge Zeit- und Budgetvorgaben	Realistische Planung, regelmäßige Fortschrittskontrollen

7.11 Technische Voraussetzungen

Technische Voraussetzungen sind die notwendigen Hard- und Softwareanforderungen, die erfüllt sein müssen, damit ein System, eine Anwendung oder ein Dienst ordnungsgemäß funktioniert.

Wichtige Aspekte technischer Voraussetzungen

Betriebssystem

Das Betriebssystem muss mit der Software kompatibel sein. Wichtige Faktoren:

- **Typ:** Windows, Linux, macOS, Unix, Android, iOS
- **Version:** Mindestanforderungen an die OS-Version
- **Architektur:** 32-Bit oder 64-Bit
- **Treiber & Updates:** Notwendige Aktualisierungen oder spezielle Treiber

Hersteller

Spezifische Anforderungen an Hardware oder Software von bestimmten Herstellern:

- **Kompatibilität:** Bestimmte Software läuft nur auf zertifizierter Hardware (z. B. Apple-Produkte)
- **Support & Garantie:** Herstellerabhängige Serviceleistungen
- **Lizenzmodelle:** Unterschiedliche Lizenzbedingungen je nach Hersteller

Hardware-Anforderungen

Mindest- und empfohlene Spezifikationen für die Nutzung einer Software oder eines Systems:

- **Prozessor (CPU):** Mindesttaktfrequenz, Anzahl der Kerne
- **Arbeitsspeicher (RAM):** Mindestgröße für reibungslosen Betrieb
- **Grafikkarte (GPU):** Anforderungen für grafikintensive Anwendungen
- **Festplattenspeicher:** Mindestfreier Speicherplatz für Installation und Betrieb

Netzwerkanforderungen

Voraussetzungen für die Verbindung mit Netzwerken oder Cloud-Diensten:

- **Bandbreite:** Mindestgeschwindigkeit für Datenübertragungen
- **Latenz:** Anforderungen für Echtzeitanwendungen (z. B. VoIP, Gaming)
- **Sicherheitsrichtlinien:** Firewall-Regeln, VPN, Proxy-Unterstützung

Softwareabhängigkeiten

Bestimmte Anwendungen benötigen zusätzliche Software-Komponenten:

- **Frameworks & Bibliotheken:** .NET, Java, Python, DirectX
- **Datenbanken:** MySQL, PostgreSQL, MS SQL Server
- **Schnittstellen (APIs):** Kompatibilität mit externen Systemen

Sicherheitsanforderungen

Technische Maßnahmen zur Sicherstellung von Datenschutz und Systemsicherheit:

- **Verschlüsselung:** SSL/TLS, AES
- **Authentifizierung:** 2-Faktor-Authentifizierung, Single Sign-On (SSO)
- **Zugriffsrechte:** Rollenbasierte Berechtigungen

7.12 Einhaltung des Budgets

Die Einhaltung des Budgets bedeutet, dass die geplanten finanziellen Mittel eines Projekts nicht überschritten werden und die Kosten innerhalb des vorgegebenen Rahmens bleiben.

Maßnahmen zur Budgetkontrolle:

- **Detaillierte Kostenplanung:** Frühzeitige Kalkulation aller Aufwände (Personal, Material, Software, Lizenzen).
- **Regelmäßige Budgetüberprüfung:** Vergleich von geplanten und tatsächlichen Kosten durch Controlling.

- **Risikomanagement**: Identifikation potenzieller Kostensteigerungen und Definition von Gegenmaßnahmen.
- **Kostensenkungsstrategien**: Optimierung von Ressourcen, Verhandlungen mit Lieferanten, Vermeidung unnötiger Ausgaben.
- **Change Management**: Bewertung von Änderungen auf Budgetauswirkungen und frühzeitige Anpassung der Planung.

7.13 Veränderungsprozesse

Veränderungsprozesse (Change Management) sind geplante und gesteuerte Anpassungen in Unternehmen oder Projekten, um neue Strukturen, Technologien oder Arbeitsweisen erfolgreich zu implementieren.

Motivierte Herangehensweise

Ein erfolgreicher Veränderungsprozess erfordert eine positive und motivierte Haltung:

- **Klare Zielsetzung**: Definition der gewünschten Veränderung und erwarteten Vorteile.
- **Transparenz**: Kommunikation der Notwendigkeit und der Vorteile der Veränderung.
- **Führungskräfte als Vorbilder**: Vorgesetzte sollten den Wandel aktiv unterstützen.
- **Partizipation**: Mitarbeiter frühzeitig einbinden, um Akzeptanz zu fördern.

Identifizierung und Darstellung der Veränderung

Damit eine Veränderung erfolgreich umgesetzt werden kann, müssen folgende Aspekte geklärt werden:

- **Analyse des Ist-Zustands**: Wo steht das Unternehmen aktuell?
- **Definition des Soll-Zustands**: Welche Ziele sollen erreicht werden?
- **Gap-Analyse**: Welche Lücken bestehen zwischen Ist- und Soll-Zustand?
- **Kommunikationsstrategie**: Wie wird der Wandel erklärt und präsentiert?

Einbeziehung der Mitarbeiter

Die aktive Beteiligung der Mitarbeiter ist entscheidend für den Erfolg eines Veränderungsprozesses:

- **Offene Kommunikation:** Sorgen und Fragen der Mitarbeiter ernst nehmen.
- **Feedbackmechanismen:** Regelmäßige Rückmeldungen ermöglichen Anpassungen.
- **Mitarbeitermotivation:** Positive Anreize schaffen, um Akzeptanz zu fördern.
- **Verantwortung übertragen:** Mitarbeiter in Entscheidungsprozesse einbinden.

Mitarbeiterqualifizierung

Neue Prozesse oder Technologien erfordern oft neue Fähigkeiten:

- **Schulungen und Workshops:** Weiterbildungsmöglichkeiten für betroffene Mitarbeiter.
- **Coaching und Mentoring:** Individuelle Unterstützung durch erfahrene Kollegen.
- **E-Learning und Selbststudium:** Flexible Lernformate für eigenständige Weiterbildung.
- **Fortlaufende Qualifizierung:** Langfristige Entwicklung der Mitarbeitenden im Blick behalten.

Promoter, Bremser, Skeptiker und Widerständler

Verschiedene Mitarbeitertypen beeinflussen den Veränderungsprozess:

Typ	Beschreibung	Umgangsstrategie
Promoter	Unterstützen aktiv den Wandel und treiben ihn voran.	Als Change-Agents einsetzen, um andere zu motivieren.
Bremser	Verzögern bewusst oder unbewusst den Prozess.	Ursachen analysieren, gezielt einbinden und Bedenken ausräumen.
Skeptiker	Zweifeln an der Veränderung, sind	Transparenz schaffen, rationale Argumente liefern.

Typ	Beschreibung	Umgangsstrategie
	aber offen für Argumente.	
Widerständler	Lehnen den Wandel aktiv ab und arbeiten dagegen.	Offenes Gespräch suchen, Ängste und Widerstände adressieren.

Ursachen von Widerständen

Widerstand gegen Veränderungen ist normal und kann verschiedene Ursachen haben:

- **Angst vor dem Unbekannten:** Unsicherheit über neue Prozesse oder Technologien.
- **Verlustängste:** Sorge um Arbeitsplatz, Status oder Einfluss.
- **Fehlende Kommunikation:** Unklare Informationen über den Veränderungsprozess.
- **Schlechte Erfahrungen:** Frühere gescheiterte Veränderungen.
- **Zusätzlicher Aufwand:** Angst vor Mehrbelastung ohne erkennbaren Nutzen.

7.14 Leistungsübergabe

Die Leistungsübergabe bezeichnet den formalen Abschluss eines Projekts oder einer Lieferung, bei dem die erbrachte Leistung an den Kunden oder Auftraggeber übergeben wird. Dabei wird geprüft, ob die vereinbarten Anforderungen erfüllt wurden.

Abnahmeprotokoll

Das **Abnahmeprotokoll** dokumentiert die Übergabe der Leistung und enthält:

- **Projekt- oder Liefergegenstand:** Detaillierte Beschreibung der erbrachten Leistung.
- **Prüfkriterien:** Festgelegte Standards und Anforderungen, die erfüllt sein müssen.
- **Mängelliste:** Falls vorhanden, erfasste Abweichungen und Fristen zur Behebung.
- **Unterschriften:** Bestätigung durch Auftraggeber und Auftragnehmer.

> ⚠ Wichtig
>
> Das Abnahmeprotokoll ist ein rechtlich relevantes Dokument und dient als Nachweis für die Erfüllung der vertraglichen Pflichten.

Mängel und Mängelarten

Ein Mangel liegt vor, wenn die erbrachte Leistung nicht den vereinbarten Anforderungen entspricht. Mängelarten:

- **Sachmangel**: Die Leistung weist physische oder funktionale Defekte auf.
- **Rechtsmangel**: Dritte haben Ansprüche an der gelieferten Ware (z. B. Patentrechte).
- **Offene Mängel**: Sofort erkennbare Fehler bei der Abnahme.
- **Versteckte Mängel**: Erst später auftretende oder entdeckte Fehler.

Schlechtleistung

Schlechtleistung liegt vor, wenn eine Leistung zwar erbracht, aber nicht vertragsgemäß ausgeführt wurde. Beispiele:

- **Fehlerhafte Verarbeitung** (z. B. ungenaue Fertigung bei einem Produkt)
- **Nicht ausreichende Qualität** (z. B. instabile Software)
- **Nicht erfüllte Anforderungen** (z. B. falsche Spezifikationen in einer Dienstleistung)

Falschlieferung

Eine **Falschlieferung** bedeutet, dass ein anderes als das bestellte Produkt geliefert wurde. Ursachen können sein:

- **Verwechslung des Produkts** durch Fehler in der Logistik.
- **Fehlkommunikation zwischen Auftraggeber und Auftragnehmer.**
- **Fehlerhafte Artikelnummern oder Bestellprozesse.**

Minderlieferung

Eine **Minderlieferung** tritt auf, wenn die gelieferte Menge geringer ist als bestellt. Dies kann entstehen durch:

- **Fehlberechnungen oder Verpackungsfehler.**
- **Lieferengpässe oder Produktionsprobleme.**
- **Fehlende Kommunikation über Teillieferungen.**

7.15 Leistungserbringung

Die Leistungserbringung umfasst alle Tätigkeiten, die zur Erfüllung eines Auftrags oder Projekts notwendig sind. Dabei wird überprüft, ob die erbrachte Leistung den vereinbarten Anforderungen entspricht.

Soll-Ist-Vergleich

Der **Soll-Ist-Vergleich** dient dazu, die tatsächliche Leistung mit den geplanten Vorgaben zu vergleichen:

- **Soll-Werte**: Geplante Kosten, Zeitaufwand, Qualität und Ressourcen.
- **Ist-Werte**: Tatsächlich aufgewendete Mittel und erzielte Ergebnisse.
- **Abweichungen**: Differenzen zwischen Soll und Ist werden analysiert.

> ⓘ Info
>
> Ein regelmäßiger Soll-Ist-Vergleich hilft, frühzeitig Probleme zu erkennen und gegenzusteuern.

Abweichungsanalyse

Wenn der Soll-Ist-Vergleich Abweichungen zeigt, erfolgt eine **Abweichungsanalyse**:

- **Identifikation der Ursachen**: Warum kam es zu Abweichungen? (z. B. unerwartete Kosten, Verzögerungen, Fehler in der Planung)
- **Bewertung der Auswirkungen**: Welche Konsequenzen haben die Abweichungen für das Gesamtprojekt?
- **Maßnahmen zur Korrektur**: Anpassung von Ressourcen, Budget oder Zeitplan.

Nachkalkulation

Die **Nachkalkulation** dient der Überprüfung der tatsächlichen Kosten nach Projektabschluss:

- **Vergleich von geplanten und realen Kosten** (Material, Personal, Zeitaufwand).
- **Analyse von Kostentreibern und Einsparpotenzialen.**
- **Optimierung der Kalkulation für zukünftige Projekte.**

Generierung von Nachfolgeaufträgen

Nach erfolgreicher Leistungserbringung können weitere Aufträge entstehen:

- **Zufriedene Kunden als Basis für Folgeaufträge** (z. B. Wartungsverträge, Erweiterungen).
- **Analyse des Kundenbedarfs** für zukünftige Dienstleistungen oder Produkte.
- **Proaktive Kundenkommunikation** zur langfristigen Zusammenarbeit.

8. Software-Entwicklung

8.1 Variablen, Datentypen und -strukturen

Variablen

Eine **Variable** ist ein Speicherplatz, der einen bestimmten Wert aufnehmen kann. Variablen haben einen Namen, einen Datentyp und einen Wert.

Datentypen

Ein **Datentyp** beschreibt die Art von Daten, die in einer Variablen gespeichert werden können. Es gibt verschiedene Datentypen, wie z.B.:

- **Ganzzahlen** (int): ganze Zahlen, z.B. 1, 2, 3
- **Fließkommazahlen** (float): Zahlen mit Dezimalstellen, z.B. 3.14, -0.5
- **Zeichen** (char): einzelne Zeichen, z.B. 'a', 'A', '!'
- **Boolesche Werte** (bool): Wahrheitswerte, z.B. true, false
- **Zeichenketten** (string): Folgen von Zeichen, z.B. "Hallo", "Welt"

Datenstrukturen

Eine **Datenstruktur** ist eine Sammlung von Daten, die in einer bestimmten Struktur organisiert sind. Es gibt verschiedene Datenstrukturen, wie z.B.:

- **Arrays**: eine Sammlung von Werten eines bestimmten Datentyps, die durch einen Index zugänglich sind.
- **Listen** (List): eine Sammlung von Werten, die in einer bestimmten Reihenfolge gespeichert sind.
- **Verknüpfte Listen** (Linked List): eine Liste, bei der jeder Eintrag auf den nächsten Eintrag verweist.
- **Stacks**: eine Datenstruktur, die es ermöglicht, Elemente hinzuzufügen und zu entfernen, wobei das letzte hinzugefügte Element zuerst entfernt wird.

- **Queues**: eine Datenstruktur, die es ermöglicht, Elemente hinzuzufügen und zu entfernen, wobei das erste hinzugefügte Element zuerst entfernt wird.

Komplexe Datenstrukturen

Es gibt auch komplexe Datenstrukturen, wie z.B.:

- **Structs**: eine Sammlung von Variablen, die zusammen eine Einheit bilden.
- **Klassen**: eine Sammlung von Variablen und Funktionen, die zusammen eine Einheit bilden.
- **Hash-Tabelle** (Hash Table): eine Datenstruktur, die es ermöglicht, Werte über einen Schlüssel zu speichern und abzurufen.

8.2 Kontrollstrukturen

Folgestruktur

Eine **Folgestruktur** ist die sequenzielle (Schritt für Schritt) Ausführung von Anweisungen in der Reihenfolge, in der sie geschrieben wurden.

Auswahlstruktur

- **Einseitige Auswahl**: If-Struktur (nur eine Bedingung und deren Ausführung)
- **Zweiseitige Auswahl**: If-Else-Struktur (zwei mögliche Ausführungspfade)
- **Mehrstufige Auswahl**: If-Elif-Else-Struktur (mehrere Bedingungen und Pfade)
- **Fallunterscheidung**: Switch/Case oder Match-Statement

Wiederholungsstruktur (Schleifen)

1. Geschlossene Schleifen
    - Feste Anzahl von Durchläufen (zählergesteuert)
    - Beispiel: For-Schleife, foreach-Schleife
2. Offene Schleifen
    - **Kopfgesteuerte Schleifen**: Bedingung wird am Anfang geprüft (while)

- **Fußgesteuerte Schleifen**: Bedingung wird am Ende geprüft (do-while)

8.3 Prozeduren und Funktionen

Einfache Funktionen

Einfache Funktionen haben keine Übergabeparameter und keinen Rückgabewert. Sie führen nur Code aus.

Funktionen mit Rückgabewert

Funktionen mit Rückgabewert haben keine Übergabeparameter, geben aber einen Wert zurück. Beispiel: Getter-Methoden oder Statusabfragen.

Funktionen mit Übergabeparameter

Funktionen mit Übergabeparameter akzeptieren einen oder mehrere Parameter, geben aber keine Werte zurück. Parameter können übergeben werden als:

- **Call-By-Value**: Übergabe einer Wertkopie (Original bleibt unverändert)
- **Call-By-Reference**: Übergabe einer Referenz/Adresse (Original kann verändert werden)

Funktionen mit Rückgabewert und Übergabeparameter

Diese Funktionen kombinieren beides: Sie akzeptieren Parameter und geben einen Wert zurück. Dies ist die flexibelste Form von Funktionen.

8.4 Objektorientierung

Objekt / Instanz

Eine **Instanz** ist ein konkretes Exemplar einer Klasse. Beispiel: "meinAuto" ist eine Instanz der Klasse "Auto" mit individuellen Attributwerten wie Farbe="rot".

Klasse

Eine **Klasse** ist ein Bauplan für Objekte mit gemeinsamer Struktur. Sie definiert, welche Attribute und Methoden alle Objekte dieser Klasse besitzen.

Methoden

Methoden sind Funktionen einer Klasse, die das Verhalten der Objekte definieren. Sie können Attribute verarbeiten und mit anderen Objekten interagieren.

Attribut

Attribute sind die Eigenschaften eines Objekts. Sie speichern die spezifischen Werte (**Attributwerte**), die den Zustand des Objekts beschreiben.

Kapselung

Kapselung bedeutet, dass Daten (Attribute) und Methoden in einer Klasse zusammengefasst und nach außen verborgen werden. Zugriff erfolgt nur über definierte Schnittstellen.

Polymorphie

Polymorphie ermöglicht es, dass eine Methode in verschiedenen Klassen unterschiedlich implementiert werden kann, aber den gleichen Namen behält.

Vererbung

Vererbung erlaubt es, Eigenschaften und Methoden einer existierenden Klasse (Elternklasse/Basisklasse) an neue Klassen (Kindklassen/Abgeleitete Klasse) weiterzugeben.

Public, Private und Protected

- **Public**: Zugriff von überall möglich
- **Private**: Zugriff nur innerhalb der eigenen Klasse
- **Protected**: Zugriff in eigener Klasse und Kindklassen

Assoziation, Aggregation und Komposition

- **Assoziation**: Lose Beziehung zwischen Klassen

- **Aggregation**: "hat ein"-Beziehung, Teile können unabhängig existieren
- **Komposition**: Starke "hat ein"-Beziehung, Teile können nicht unabhängig existieren

8.5 Bibliotheken und Frameworks

Bibliotheken

Eine **Bibliothek** ist eine Sammlung von vordefinierten Funktionen und Klassen, die von einem Programm verwendet werden können. Bibliotheken erleichtern die Entwicklung von Programmen, indem sie wiederkehrende Aufgaben übernehmen und auslagern.

Frameworks

Ein **Framework** ist eine umfassende Sammlung von Bibliotheken, Tools und Konventionen, die die Entwicklung von Programmen erleichtern. Frameworks bieten eine Struktur für die Entwicklung von Anwendungen und erleichtern die Wiederverwendung von Code.

Vorteile

- Erleichtern die Entwicklung von Programmen
- Reduzieren die Entwicklungzeit
- Erhöhen die Wiederverwendbarkeit von Code
- Bieten eine umfassende Funktionalität

8.6 Skriptsprachen

Skriptsprachen sind interpretierte Sprachen, welche zudem meist auf gewisse Sprachelemente wie z.B. die Deklaration von Variablen. Sie werden hauptsächlich für kleine Programme und Algorithmen benutzt.
Ein Beispiel hierfür ist Shellscipt.

Shellscript

Ein Shellskript ist ein Skript, das vom Shell-Interpreter ausgeführt wird. In der ersten Zeile eines Skriptes wird mit `#!/bin/sh` oder ähnlichem deklariert, welcher Interpreter das Skript ausführen soll. Die Dateiendung für Shellskripte ist typischerweise `.sh`.

Ein einfaches Beispiel für ein Shellskript könnte wie folgt aussehen:

```
#!/bin/sh
echo "Hallo Welt!"
```

Dieses Skript gibt einfach den Text "Hallo Welt!" auf der Konsole aus.

Ein weiteres Beispiel könnte ein Skript sein, das eine Liste von Dateien in einem Verzeichnis auflistet:

```
#!/bin/sh
ls -l
```

Dieses Skript führt den Befehl `ls -l` aus, der eine detaillierte Liste der Dateien im aktuellen Verzeichnis anzeigt.

8.7 Pseudocode

Verzweigungen

- **IF-Anweisung:**

  ```
  IF Bedingung THEN
      Anweisung(en)
  END IF
  ```

- **IF-ELSE:**

  ```
  IF Bedingung THEN
      Anweisung(en)
  ELSE
      Anweisung(en)
  END IF
  ```

- **IF-ELSE IF-ELSE:**

```
IF Bedingung1 THEN
    Anweisung(en)
ELSE IF Bedingung2 THEN
    Anweisung(en)
ELSE
    Anweisung(en)
END IF
```

- **SWITCH/CASE:**

```
SWITCH Variable
    CASE Wert1:
        Anweisung(en)
        BREAK
    CASE Wert2:
        Anweisung(en)
        BREAK
    DEFAULT:
        Anweisung(en)
END SWITCH
```

Schleifen

- **WHILE-Schleife:**

```
WHILE Bedingung DO
    Anweisung(en)
END WHILE
```

- **REPEAT-UNTIL-Schleife:**

```
REPEAT
    Anweisung(en)
UNTIL Bedingung
```

- **FOR-Schleife:**

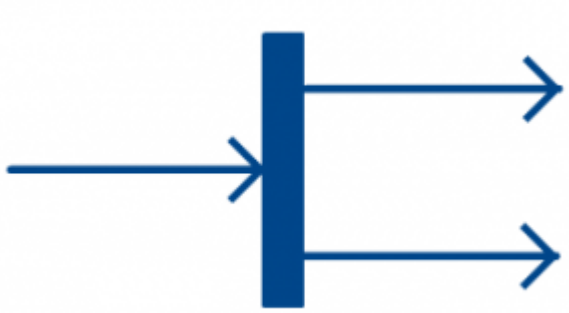
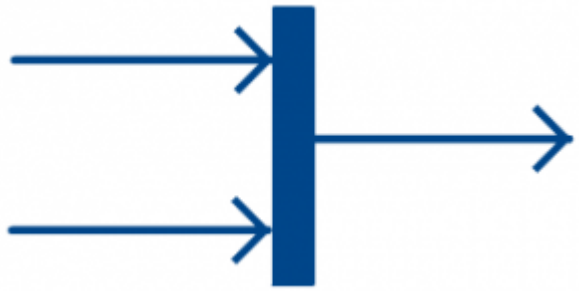
```
FOR Variable = Startwert TO Endwert DO
    Anweisung(en)
END FOR
```

8.8 UML

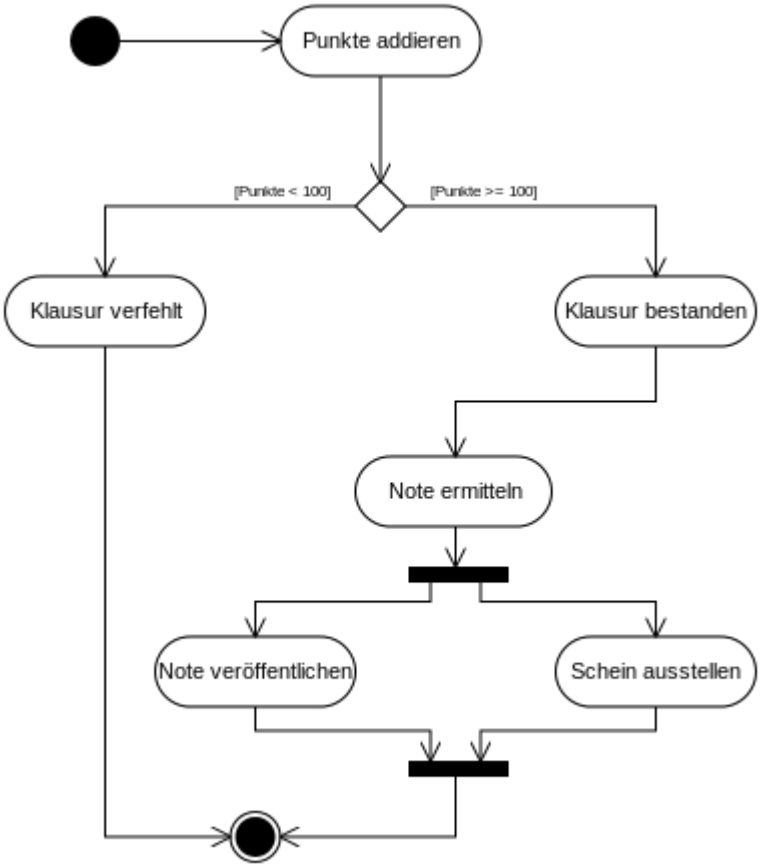
Aktivitätsdiagramm

Ein **Aktivitätsdiagramm** ist eine Art von UML-Diagramm, das den Ablauf von Aktivitäten und Aktionen innerhalb eines Systems veranschaulicht. Es wird verwendet, um den Workflow eines Systems zu modellieren und die Schritte zu identifizieren, die in einem Prozess involviert sind.

Symbol	Name
	Start/Anfangsknoten
	Aktivität/Aktionsstatus
	Aktion
	Kontrollfluss/Kante
	Objektfluss/Steuerkante
	Aktivität Endknoten
	Fluss-Endknoten
	Entscheidungs-Knotenpunkt

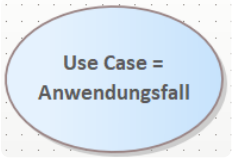
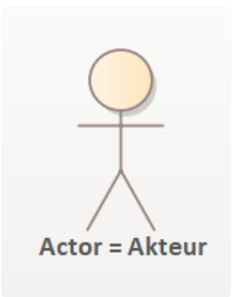
Symbol	Name
	Knoten verschmelzen
	Gabel
	Zusammenführen
	Senden von Signalen
	Signal-Empfang
	Anmerkung/Kommentar

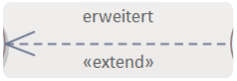
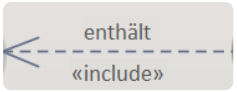
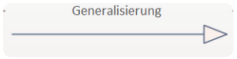
Beispiel:



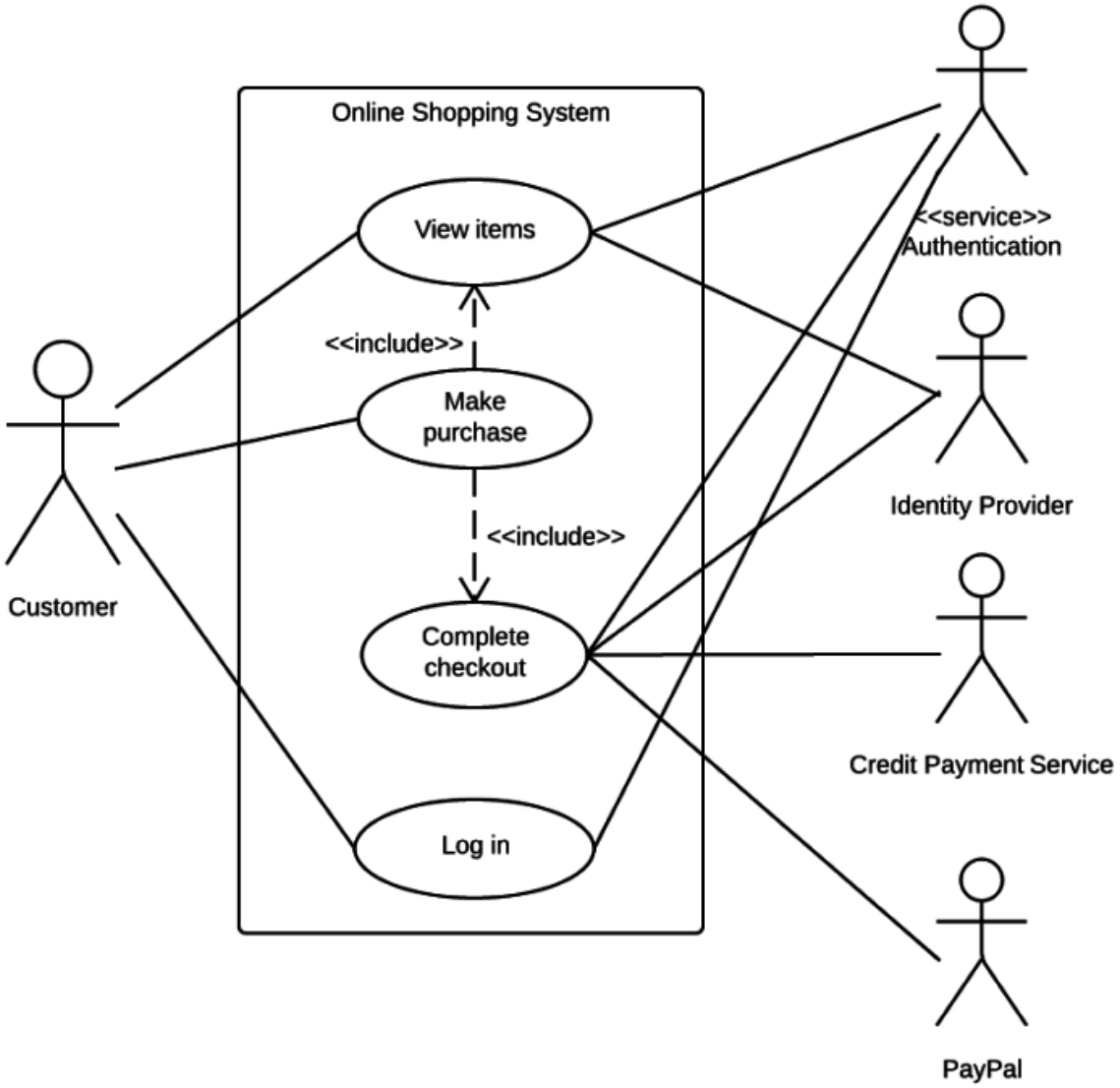
Anwendungsfalldiagramm

Ein **Anwendungsfalldiagramm** ist ein UML-Diagramm, das die Interaktionen zwischen einem System und seinen Benutzern zeigt. Es wird verwendet, um die funktionalen Anforderungen eines Systems zu identifizieren und die Beziehungen zwischen Akteuren und Anwendungsfällen zu modellieren.

Symbol	Bedeutung
	**Anwendungsfall** – Eine Funktion oder ein Prozess, der vom System bereitgestellt wird.
	**Akteur** – Eine Person oder ein System, das einen Anwendungsfall auslöst.

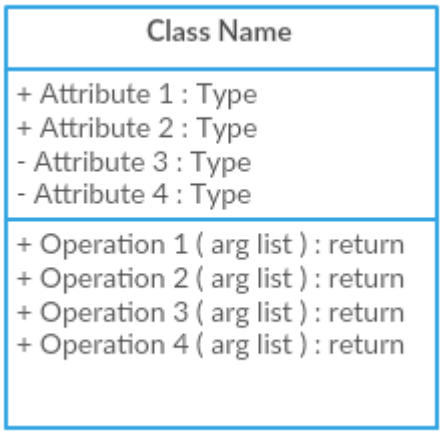
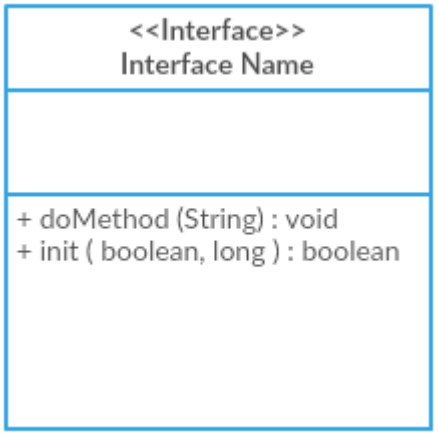
Symbol	Bedeutung
	**Beziehung zwischen Akteur und Anwendungsfall** – Ein Akteur nutzt einen Anwendungsfall, wenn er ihn auslöst.
	**«extend»-Beziehung** – Ein Anwendungsfall erweitert einen anderen unter bestimmten Bedingungen. Die Pfeilspitze zeigt auf den erweiterten Anwendungsfall.
	**«include»-Beziehung** – Ein Anwendungsfall ist fester Bestandteil eines anderen. Die Pfeilspitze zeigt auf den enthaltenen Anwendungsfall.
	**Generalisierung** – Ein Akteur oder Anwendungsfall wird spezialisert. Die Pfeilspitze zeigt auf das übergeordnete Element.
	**Notiz** – Zusatzinformationen zum Modell, verbunden mit einer gestrichelten Linie.

Beispiel:

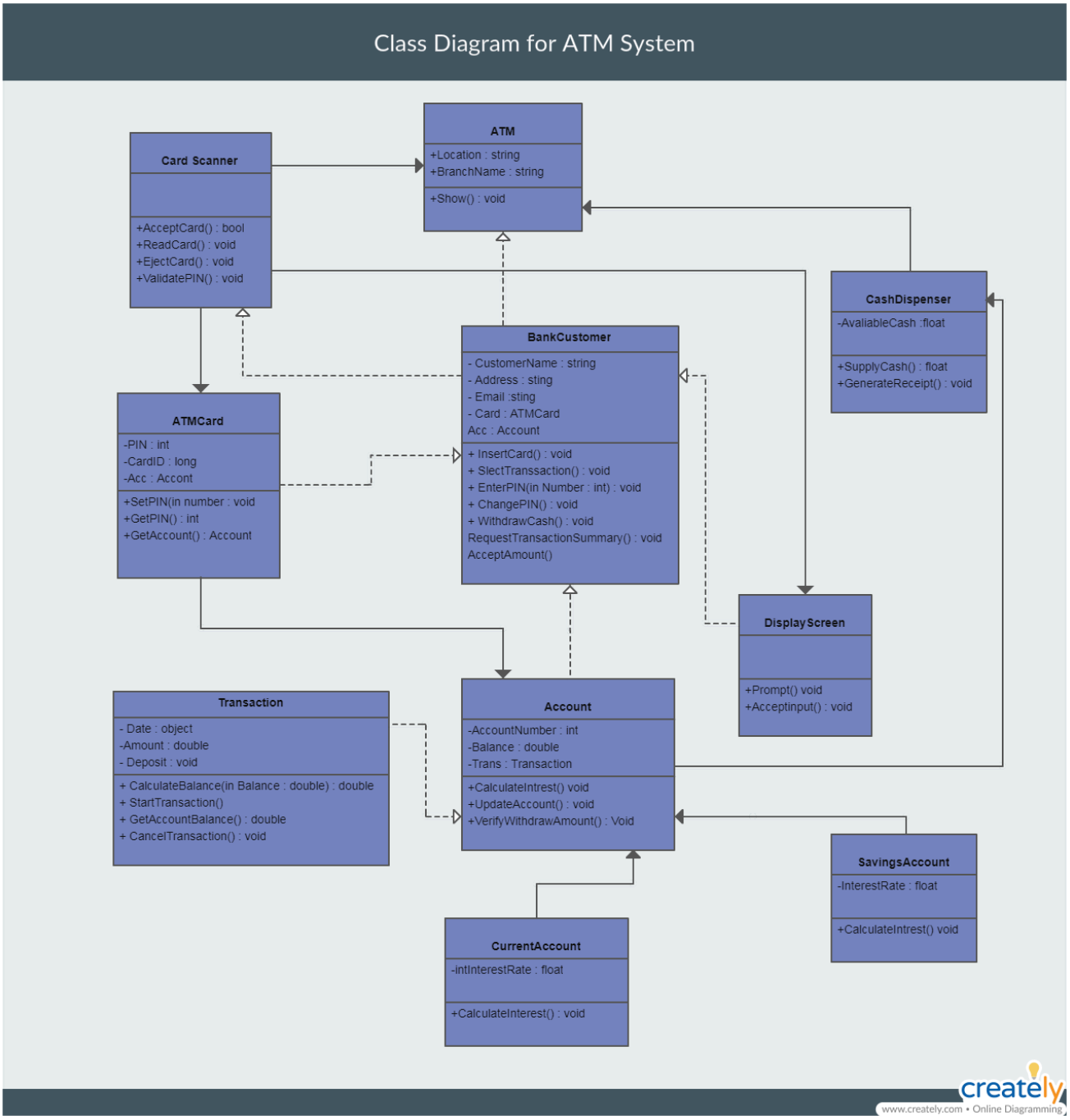


Klassendiagramm

Ein **Klassendiagramm** ist ein UML-Diagramm, das die Struktur eines Systems durch die Darstellung von Klassen, Attributen und Beziehungen zwischen ihnen veranschaulicht. Es wird verwendet, um die statische Struktur eines Systems zu modellieren und die Klassen und Objekte zu identifizieren, die involviert sind.

Symbol	Name
	Klasse
	Schnittstelle / Interface
	Assoziation
	Vererbung
	Implementation
	Abhängigkeit
	Aggregration
	Komposition

Beispiel:



8.9 Tests

White Box Test

- Tester kennt den Quellcode ("durchsichtige Box")
- Test der internen Programmlogik und Abläufe
- Prüfung der Anweisungen, Zweige und Bedingungen
- Geeignet für Unit Tests

Black Box Test

- Tester kennt nur Ein- und Ausgaben ("schwarze Box")

- Test des externen Verhaltens ohne Codekenntnis
- Prüfung von Funktionalität und Spezifikationen
- Geeignet für System- und Abnahmetests

Schreibtisch-Test

Beim **Schreibtisch-Test** geht der Tester den Code oder Algorithmus im Kopf durch, ohne ihn tatsächlich auszuführen, und simuliert damit das Verhalten eines Computers, um Fehler oder Schwachstellen zu finden.

8.10 Bildschirmausgabemasken

Softwareergonomie

- **Ziel:** Benutzerfreundliche und intuitive Oberflächen
- **Prinzipien:**
  - Konsistentes Layout und Navigation
  - Klare, verständliche Symbole und Beschriftungen
  - Minimierung der Komplexität (wenige Klicks zum Ziel)
  - Feedback bei Aktionen (z.B. Ladeanzeigen, Bestätigungsdialoge)

Corporate Identity

- **Ziel:** Einheitlicher Auftritt der Marke
- **Aspekte:**
  - Verwendung von definierten Farben, Schriften und Logos
  - Konsistenz in Stil und Ton der Kommunikation
  - Anpassung an firmenspezifische Richtlinien und Design-Vorlagen

Barrierefreiheit

- **Ziel:** Zugänglichkeit für alle Benutzer, inkl. Menschen mit Einschränkungen
- **Maßnahmen:**
  - Hoher Kontrast und skalierbare Schriftgrößen
  - Alternative Texte für Bilder und Symbole (Screenreader-kompatibel)
  - Tastatur-Navigation und barrierefreie Formulare

- Einhaltung von Standards (z.B. WCAG-Richtlinien)

8.11 Relationale Datenbanken

Eine Relationale Datenbank besteht aus **Tabellen** (Relationen), **Zeilen** (Tupel), **Spalten** (Attribute), **Primärschlüssel** und **Fremdschlüssel**. Sie organisiert Daten in tabellarischer Form, wobei die Tabellen durch definierte Beziehungen miteinander verknüpft sind.

Vorteile: Datenkonsistenz, einfache Abfragen
Nachteile: Planungskomplexität, Leistungsprobleme bei vielen Daten

Entität (Entity)

Eine Entität ist ein eindeutig identifizierbares Objekt/Konzept aus der realen Welt, welches in einer Datenbank gespeichert werden kann.

Entitätsmenge/Entitätstyp

Die Entitätsmenge ist die Gesamtheit aller **gleichen** Entitäten. Gleich bedeutet in dem Fall, dass sie mit den gleichen Attributen beschrieben werden.

Attribut

Ein Attribut ist eine Eigenschaft einer Entität. Das kann zum Beispiel der Name, oder die Farbe eines Objektes sein.

Beziehung (Relationship)

Eine Beziehung ist eine Verknüpfung zwischen zwei oder mehreren Entitäten

Kardinalität

Eine Kardinalität beschreibt wie viele Entitäten eines Types in Beziehung mit Entitäten eines anderen Types stehen. Mögliche Kardinalitäten sind: **1:n**, **m:n**, **1:1**

⚠ WICHTIG!

m:n Kardinalitäten MÜSSEN durch eine eigene Tabelle (Beziehungstabelle) dargestellt werden. Zu dieser Tabelle werden dann **1:n** Beziehungen hergestellt.

Primärschlüssel (Primary Key, PK)

Ein Primärschlüssel ist ein Attribut oder eine Kombination von Attributen einer Entität, welches diese eindeutig identifiziert. Innerhalb einer Tabelle darf kein Primärschlüsselwert mehrfach vorkommen (Eindeutigkeit) und der Primärschlüssel darf nicht NULL sein.

Fremdschlüssel (Foreign Key, FK)

Ein Fremdschlüssel ist ein Attribut oder eine Attributkombination in einer Tabelle, das auf den Primärschlüssel einer anderen (oder derselben) Tabelle verweist. Der Fremdschlüssel stellt damit eine referentielle Integrität zwischen den Tabellen her.

Normalisierung

Die Normalisierung ist ein systematischer Prozess zur Strukturierung von Datenbanken, der Redundanzen vermeidet und Datenanomalien verhindert. Es gibt 3 verschiedene Normalformen (NF)

Referenzielle Integrität

Referentielle Integrität sind Regeln bei Relationalen Datenbanken, die sicherstellen, dass Beziehungen zwischen Tabellen konsistent bleiben. Sie garantiert, dass Fremdschlüssel nur auf wirklich existierende Primärschlüssel verweisen dürfen. Dies wird durch zwei Hauptregeln gewährleistet:

1. Ein Fremdschlüssel muss auf einen existierenden Primärschlüssel verweisen oder NULL sein
2. Ein Primärschlüssel kann nicht gelöscht oder geändert werden, solange noch Fremdschlüssel darauf verweisen

Anomalien

Einfügeanomalie

Tritt auf, wenn neue Daten nicht eingefügt werden können, weil notwendige Informationen fehlen oder Abhängigkeiten dies verhindern.

Änderungsanomalie

Wenn eine Information an mehreren Stellen gespeichert ist und nicht überall konsistent geändert wird, entstehen widersprüchliche Daten.

Löschanomalie

Beim Löschen eines Datensatzes gehen unbeabsichtigt auch andere, noch benötigte Informationen verloren.

Datenredundanz

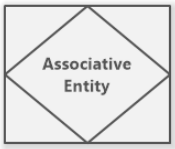
Datenredundanz heißt, wenn identische Datenmehrfach in einer Datenbank gespeichert werden.

Datenkonsistenz

Datenkonsistenz heißt, dass die Daten korrekt, widerspruchsfrei und vollständig sind.

8.12 ERD Chen-Notation

Symbole

Symbol	Name	Bedeutung
	Entität	Ein konkretes oder abstraktes Objekt (z.B. Kunde, Produkt)
	Schwache Entität	Abhängige Entität, die ohne Hauptentität nicht existieren kann
	Verbindungsentität	Entität zur Auflösung einer N:M Beziehung

Symbol	Name	Bedeutung
	Attribut	Merkmal oder Eigenschaft einer Entität
	Schlüsselattribut	Eindeutiger Identifikator (Primärschlüssel)
	Teilschlüssel	Schlüssel einer schwachen Entität
	Mehrwertiges Attribut	Attribut mit mehreren möglichen Werten
	Abgeleitetes Attribut	Berechnetes Attribut aus anderen Attributen
	Beziehung	Verbindung zwischen Entitäten
	Identifizierende Beziehung	Beziehung bei schwachen Entitäten

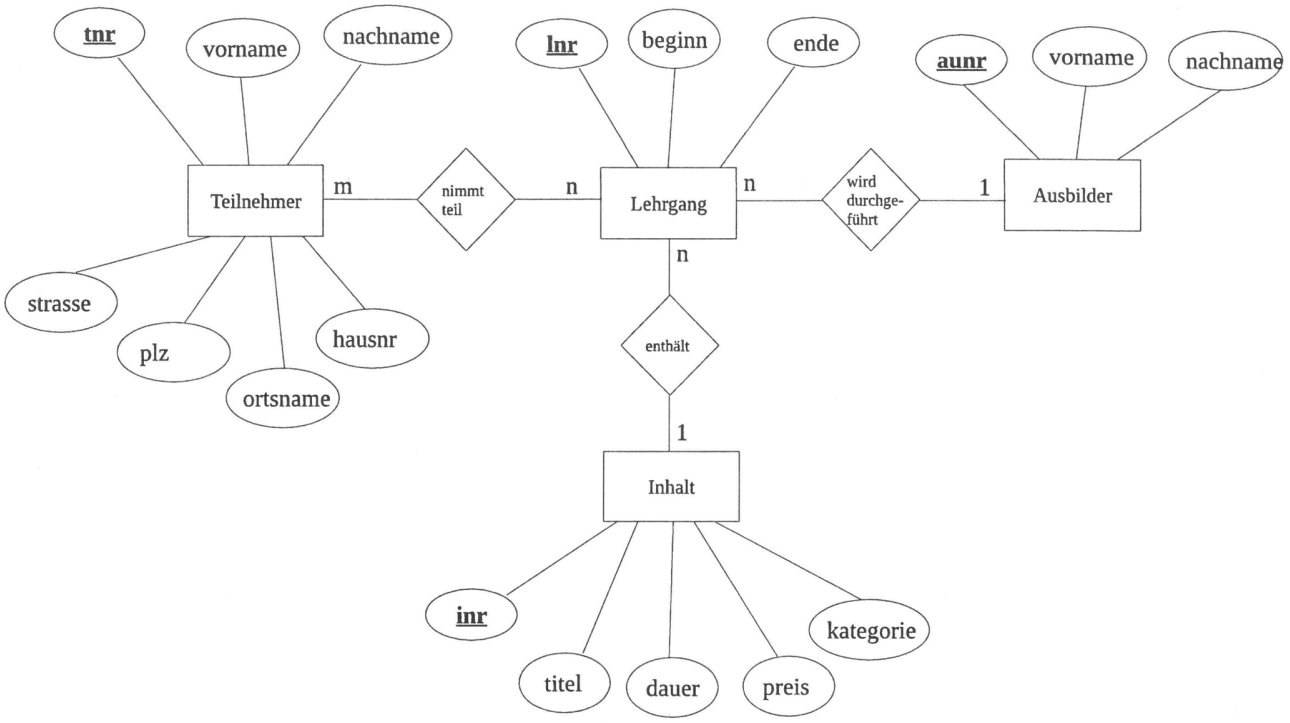
Teilnahmegrad

Symbol	Bedeutung
=	Total (verpflichtend)
–	Partiell (optional)
---	Optional (gestrichelt)

Kardinalitäten

- 1:1 (Eins-zu-Eins)
- 1:N (Eins-zu-Viele)
- N:1 (Viele-zu-Eins)
- N:M (Viele-zu-Viele)

Beispiel


tnr
vorname
nachname
Teilnehmer
m
nimmt teil
n
lnr
beginn
ende
Lehrgang
n
wird durchge-führt
1
aunr
vorname
nachname
Ausbilder
strasse
plz
ortsname
hausnr
n
enthält
1
Inhalt
inr
titel
dauer
preis
kategorie

9. Support

9.1 Kommuniaktion

Kommunikationsarten

- **Verbale Kommunikation**: Direkte Kommunikation über gesprochene oder geschriebene Wörter.
- **Paraverbale Kommunikation**: Tonfall, Lautstärke, Sprechgeschwindigkeit.
- **Nonverbale Kommunikation**: Körpersprache, Gestik und Mimik.

Konfliktgespräch deeskalieren

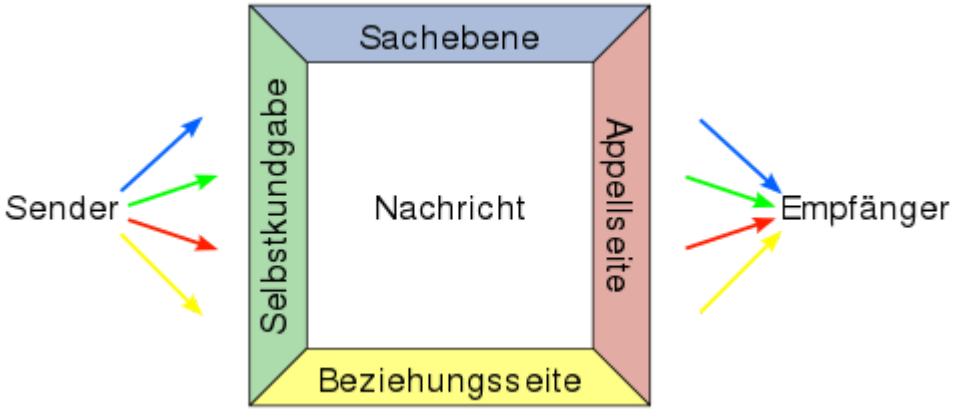
- Ruhige Kommunikation
- Empathie
- Konkrete Hilfen und Lösungsansätze finden
- Schuldzuweisung vermeiden
- Sachlich bleiben
- Trennung von sachlicher und persönlicher Ebene
- Weitere Gespräche anbieten mit dem Ziel, eine Lösung zu finden

4-Ohren-Modell

Das **4-Ohren-Modell** formuliert die Annahme, dass jede kommunikative Nachricht verschiedene Ebenen enthält, die ein differentes Verständnis einer Nachricht ermöglichen. Die Ebenen sind:

- **Appell**: Was ich von dir möchte
- **Beziehung**: Wie ich dich einschätze und zu dir stehe
- **Sachinhalt**: Was ich dir sagen will

- **Selbstoffenbarung:** Was ich von mir erzählen will

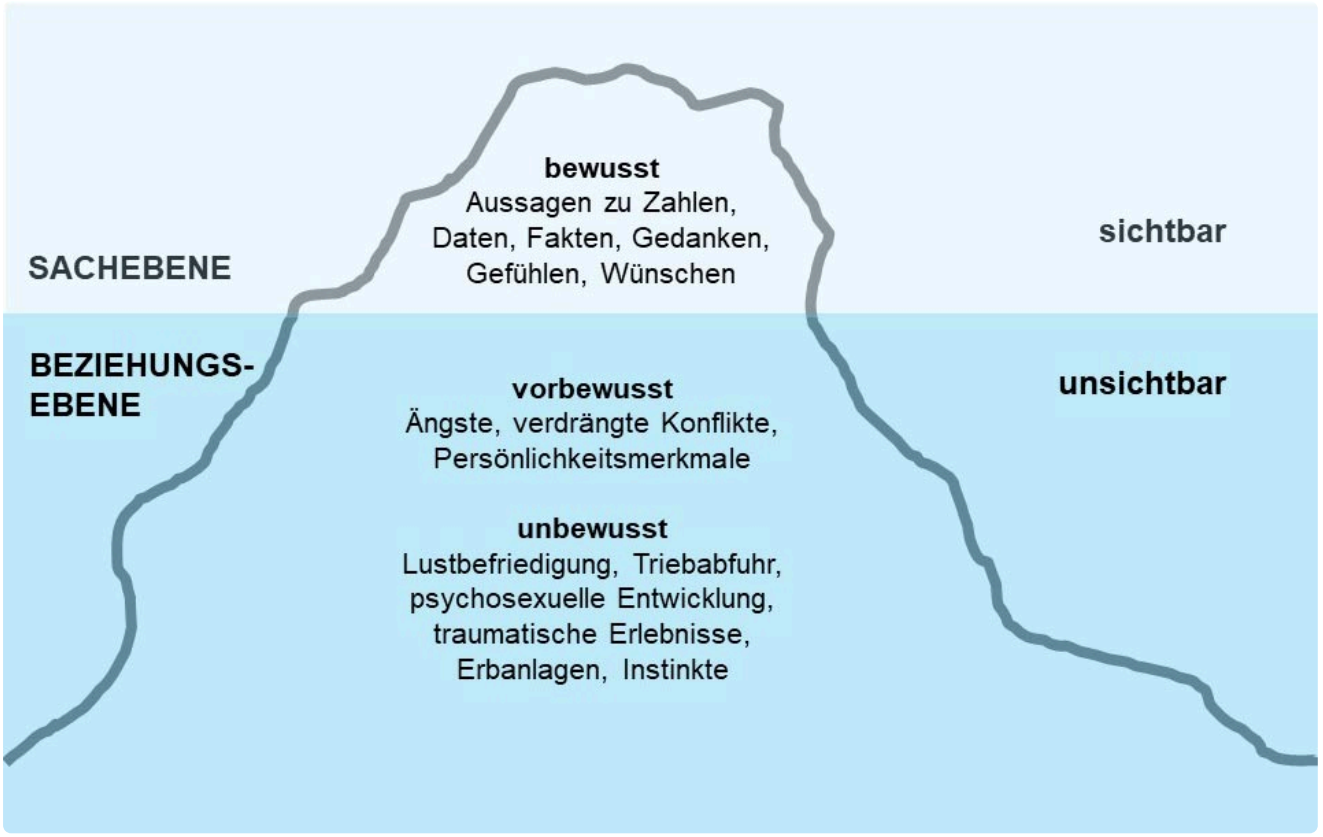


Sender-Empfänger-Modell

Das **Sender-Empfänger-Modell** veranschaulicht, wie Kommunikation funktioniert, indem das Modell den Kommunikationsprozess vereinfacht abbildet. Das Modell besteht aus mehreren Schlüsselkomponenten: dem Sender, der Codierung, dem Kanal, dem Empfänger und zumeist den Störungen während des Übertragungsprozesses.

Eisbergmodell

Das **Eisbergmodell** verdeutlicht, dass es bei der zwischenmenschlichen Kommunikation eine sichtbare Sachebene und eine unsichtare Beziehungsebene gibt.



9.2 Fehlermanagement

Problem Control

Identifikation, Dokumentation und Klassifizierung von Problemen.

Error Control

Überwachung bekannter Fehler, um diese mittels Problemumgehungen zu minimieren.

Proaktives Problem Management

Probleme finden, bevor Störungen auftreten.

Workaround

Ein **Workaround** ist eine temporäre Lösung, die eingesetzt wird, bis eine permanente Lösung gefunden wurde.

9.3 Störungsmanagement

Urgency (Incident Management)

Beschreibt, wie dringend ein **Incident** behoben werden muss.

Impact (Incident Management)

Beschreibt, welche Auswirkungen ein **Incident** auf den Geschäftsbetrieb hat.

9.4 Ticketsystem

Ein Ticket-System ist eine Software zur strukturierten Bearbeitung von Anfragen und Problemen.

- **Issue-Tracking-System**: Verfolgung & Verwaltung von Fehlern
- **Helpdesk-System**: Erfassung von Kundenanfragen

9.5 Support- und Serviceanfragen

First-Level-Support

Erste Anlaufstelle für IT-Probleme und Anfragen von Benutzern/Kunden, meist ohne tiefgehendes IT-Fachwissen.

Second-Level-Support

Bearbeitung komplexerer Probleme, die der First-Level-Support nicht lösen kann.

Third-Level-Support

Spezialisierte Problemlösung für hochkomplexe oder kritische Fehler (z.B. Hersteller-Support oder speziell geschultes Personal).

Kundenanfrage (Service Request)

Ein **Service Request** ist eine Anfrage eines Nutzers nach einer Ressource oder Dienstleistung, z. B. ein neuer Arbeitsplatz für einen neuen Mitarbeiter.

Statusmeldung (Event)

Ein **Event** ist eine automatisierte oder manuelle Meldung über den Zustand eines Systems oder einer Ressource (z. B. Warnung eines Druckers über leeren Toner).

Störungsfall (Incident)

Ein **Incident** ist eine unerwartete Unterbrechung oder ein Ausfall einer Dienstleistung, z. B. eine nicht erreichbare Webseite oder fehlender Zugriff auf Personalakten.

Service Request

Formelle Anfrage eines Kunden an den Kundenservice, bei der der Kunde **aktiv** einen Mitarbeiter kontaktiert.

Self Service

Der Kunde löst das Problem selbst mit bereitgestellten Tools und Anleitungen, wobei die Interaktion mit einem Mitarbeiter **indirekt** erfolgt (z. B. Chatbots, FAQ-Sektionen).

Knowledge Base

Ein Dokument oder Wiki, das mit Anleitungen und Beschreibungen zur Lösungsfindung gefüllt ist – nutzbar durch den Kundenservice oder direkt von Kunden im Rahmen des Self Service.

9.6 Service Level

Service Level

Gibt an, welche Qualitätsstufe ein Produkt/Service haben soll.

Service Level Requirement

Beschreibt die Anforderungen an einen IT-Service aus Kundensicht.

Service Level Agreement (SLA)

Ein Vertrag zwischen **IT-Service** und **Kunden**, in dem das gewünschte/erwartete **Service Level** definiert wird. Enthält u. a.:

- Leistungsbeschreibung
- Verfügbarkeitszeiten
- Reaktionszeiten bei Störungen
- Supportzeiten
- Strafen bei Vertragsbruch

Der Kunde hat eine **Mitwirkungspflicht**; er muss z. B. rechtzeitig auf Probleme aufmerksam machen und dem IT-Service Zugriff auf alle relevanten Räumlichkeiten und Systeme gewähren.

10. Qualitätsmanagement

10.1 QM-Systeme

Qualitätsmanagementsysteme (QMS) sind strukturierte Ansätze zur Sicherstellung der Qualität in Unternehmen. Sie umfassen Prozesse, Richtlinien und Verantwortlichkeiten zur kontinuierlichen Verbesserung der Qualität.

10.2 QS-Normen

Qualitätssicherungsnormen (QS-Normen) definieren Standards zur Gewährleistung der Produkt- und Dienstleistungsqualität. Sie dienen als Leitfaden für Unternehmen zur Erfüllung bestimmter Qualitätsanforderungen.

Wichtige QS-Normen:

- **ISO 9001** – Qualitätsmanagementsysteme
- **ISO 27001** – Informationssicherheitsmanagement

10.3 Zertifizierung

Eine Zertifizierung ist ein formaler Nachweis, dass ein Unternehmen oder Produkt festgelegte Standards erfüllt. Sie wird durch externe Prüfstellen (z. B. TÜV, DEKRA) durchgeführt.

Ablauf einer Zertifizierung:

1. Vorbereitung (Selbstevaluierung, Dokumentation)
2. Audit durch eine Zertifizierungsstelle
3. Ausstellung des Zertifikats bei Erfolg
4. Regelmäßige Überwachungsaudits

Beispiele: ISO-Zertifikate, CE-Kennzeichnung, ITIL-Zertifizierung

10.4 Qualitätsplanung und -ziele

Qualitätsplanung umfasst die Definition von Qualitätszielen und Maßnahmen zur Erreichung dieser Ziele.

Schritte der Qualitätsplanung:

1. **Ist-Zustand ermitteln**
   - Analyse bestehender Prozesse
   - Erfassung von Fehlerquellen
2. **Ziel-Zustand festlegen**
   - Definition von Qualitätsstandards
   - Festlegung messbarer Qualitätsziele

Beispiel für Qualitätsziele:

- Reduzierung der Fehlerquote um 10 %
- Verbesserung der Kundenzufriedenheit auf 90 %

10.5 Qualitätslenkung

Qualitätslenkung umfasst Maßnahmen zur Steuerung und Kontrolle von Prozessen zur Einhaltung der Qualitätsanforderungen.

Methoden der Qualitätslenkung:

- **Präventiv:** Fehlervermeidung durch Standards und Schulungen
- **Korrektiv:** Maßnahmen zur Behebung erkannter Mängel
- **Feedback-basiert:** Kontinuierliche Anpassung durch Kunden- und Mitarbeiterfeedback

Werkzeuge:

- Prozesskontrollen
- Qualitätsprüfungen
- Fehleranalyse

10.6 PDCA

Der PDCA-Zyklus ist eine iterative Methode zur kontinuierlichen Verbesserung von Prozessen.

1. **Plan (Planen)** – Problem analysieren, Maßnahmen festlegen
2. **Do (Umsetzen)** – Maßnahmen in kleinem Rahmen testen
3. **Check (Überprüfen)** – Ergebnisse auswerten, Wirksamkeit prüfen
4. **Act (Handeln)** – Erfolgreiche Maßnahmen standardisieren, Fehler korrigieren

10.7 Testprotokoll

Ein Testprotokoll dokumentiert die Prüfungen und Ergebnisse beim Einrichten eines Arbeitsplatzes.

Typischer Aufbau:

- **Allgemeine Daten** (Datum, Prüfer, Arbeitsplatz-ID)
- **Prüfpunkte** (Hardware, Software, Netzwerkverbindung, Sicherheitsrichtlinien)
- **Ergebnisse** (Bestanden/Nicht bestanden, Abweichungen)
- **Maßnahmen** (Fehlerbehebung, Nachkontrolle)

11. IT-Sicherheit und Datenschutz

11.1 Verfügbarkeit, Vertraulichkeit und Integrität

Schutzziele der IT-Sicherheit sind Verfügbarkeit, Vertraulichkeit und Integrität (CIA-Triade). Diese müssen durch geeignete Maßnahmen sichergestellt werden.

Verfügbarkeit (Availability)

Daten und Systeme müssen stets zugänglich sein.
Maßnahmen:

- Redundanz (z. B. Cluster, Backup-Server)
- Notfallpläne & Disaster Recovery
- DDoS-Schutz & Lastverteilung
- Wartung & Updates zur Fehlervermeidung

Vertraulichkeit (Confidentiality)

Zugriff auf Daten ist nur autorisierten Personen erlaubt.
Maßnahmen:

- Verschlüsselung (z. B. TLS, AES)
- Zugriffskontrollen (RBAC, ACLs)
- Sichere Authentifizierung (2FA, biometrische Verfahren)
- Datensparsamkeit & Rechteverwaltung

Integrität (Integrity)

Daten dürfen nicht unbemerkt verändert oder manipuliert werden.
Maßnahmen:

- Hashing & Prüfsummen (z. B. SHA-256)
- Digitale Signaturen & Zertifikate
- Zugriffskontrollen & Änderungsprotokolle (Logging)
- Manipulationssichere Speicherung (WORM-Prinzip)

11.2 Maßnahmen zur Informationssicherheit

Informationssicherheit umfasst technische, organisatorische und personelle Maßnahmen zum Schutz von Daten und Systemen.

Technische Maßnahmen

Schützen IT-Systeme durch Hard- und Softwarelösungen.

- **Netzwerksicherheit:** Firewalls, Intrusion Detection/Prevention Systems (IDS/IPS)
- **Verschlüsselung:** TLS, AES, VPNs für sichere Kommunikation
- **Zugangskontrollen:** Biometrie, Zwei-Faktor-Authentifizierung (2FA)
- **Sicherheitsupdates:** Regelmäßige Patches und Aktualisierungen
- **Backups & Redundanz:** Regelmäßige Sicherungen, Georedundanz

Organisatorische Maßnahmen

Regeln und Prozesse zur Einhaltung von Sicherheitsrichtlinien.

- **ISMS (Information Security Management System):** Systematische Verwaltung der Informationssicherheit (z. B. nach ISO 27001)
- **Richtlinien & Schulungen:** Sicherheitsrichtlinien für Mitarbeiter, Awareness-Trainings
- **Zugriffsmanagement:** Rollen- und Rechtekonzepte (z. B. Least Privilege Principle)
- **Notfallmanagement:** Disaster Recovery Pläne, Incident Response

Personelle Maßnahmen

Mitarbeiter als Sicherheitsfaktor durch Sensibilisierung und Schulung.

- **Awareness-Programme:** Phishing-Tests, Schulungen zu Social Engineering
- **Verhaltensrichtlinien:** Klare Vorgaben zur Passwortsicherheit, Nutzung externer Geräte
- **Vertraulichkeitsvereinbarungen:** NDA, Geheimhaltungsvereinbarungen

11.3 Technisch-organisatorische Maßnahmen

11.3 IT-Sicherheitsbeauftragter

Der IT-Sicherheitsbeauftragte (ITSB) ist für die Planung, Umsetzung und Überwachung der IT-Sicherheitsmaßnahmen in einem Unternehmen verantwortlich. Er arbeitet an der Risikominimierung und Einhaltung gesetzlicher Vorgaben.

Aufgaben

- Entwicklung und Umsetzung von **Sicherheitsrichtlinien**
- Überwachung der Einhaltung von **IT-Sicherheitsstandards** (z. B. ISO 27001, BSI-Grundschutz)
- Durchführung von **Risikoanalysen und Schwachstellenbewertungen**
- Koordination von **Maßnahmen zur IT-Sicherheit** (Firewalls, Verschlüsselung, Zugriffskontrollen)
- Schulung und Sensibilisierung der Mitarbeiter in **IT-Sicherheitsfragen**
- Zusammenarbeit mit Datenschutzbeauftragten und IT-Abteilungen
- Reaktion auf **Sicherheitsvorfälle und Notfallmanagement**

Rechtliche Grundlagen und Anforderungen

- Kein gesetzlich vorgeschriebener Pflichtposten, aber empfohlen nach **IT-Sicherheitsgesetz (IT-SiG)**
- DSGVO fordert angemessene Schutzmaßnahmen, die durch einen ITSB koordiniert werden können
- In Unternehmen mit **kritischer Infrastruktur (KRITIS)** kann ein ITSB erforderlich sein

Anforderungen

- Fachwissen in IT-Sicherheit, Netzwerksicherheit und Datenschutz
- Kenntnisse in IT-Sicherheitsstandards und Compliance-Anforderungen
- Kommunikationsstärke und Fähigkeit zur Schulung von Mitarbeitern
- Erfahrung in **Risikomanagement und Incident Response**

11.4 Datenschutzbeauftragter

Der Datenschutzbeauftragte (DSB) ist für die Überwachung und Einhaltung des Datenschutzes gemäß **DSGVO (Art. 37-39)** verantwortlich. Er berät das Unternehmen in Datenschutzfragen und stellt sicher, dass personenbezogene Daten rechtskonform verarbeitet werden.

Aufgaben

- Überwachung der Einhaltung der **DSGVO und des BDSG**
- Beratung der Geschäftsleitung und Mitarbeiter zu Datenschutzfragen
- Erstellung und Pflege von **Datenschutzrichtlinien und -konzepten**
- Durchführung von **Datenschutz-Folgenabschätzungen (DPIA)**
- Schulung und Sensibilisierung der Mitarbeiter
- Zusammenarbeit mit der **Aufsichtsbehörde** bei Datenschutzvorfällen
- Kontrolle der **Technisch-organisatorischen Maßnahmen (TOM)**

Wann ist ein Datenschutzbeauftragter erforderlich?

Ein Unternehmen muss einen DSB benennen, wenn:

- Es **mindestens 20 Personen** mit der automatisierten Verarbeitung personenbezogener Daten beschäftigt (§ 38 BDSG).
- Die Kerntätigkeit in der **umfangreichen Verarbeitung sensibler Daten** (z. B. Gesundheitsdaten) besteht.
- Es eine **behördliche oder öffentliche Stelle** ist.

Anforderungen

- Fachkunde im **Datenschutzrecht und IT-Sicherheit**
- Unabhängigkeit und Weisungsfreiheit
- Vermeidung von Interessenkonflikten (z. B. keine gleichzeitige IT-Leitung)

11.5 IT-Sicherheitsrichtlinien

IT-Sicherheitsrichtlinien sind verbindliche Vorgaben eines Unternehmens zur Gewährleistung von **Verfügbarkeit, Vertraulichkeit und Integrität** von IT-Systemen und Daten. Sie definieren Regeln und Maßnahmen zum Schutz vor Cyberangriffen, Datenverlust und unbefugtem Zugriff.

Inhalte einer IT-Sicherheitsrichtlinie

1. **Zugriffs- und Berechtigungskonzepte**
   - Nutzung von **starken Passwörtern** und **Multi-Faktor-Authentifizierung (MFA)**
   - Prinzip der **minimalen Rechtevergabe (Least Privilege Principle)**
   - Regelungen für **Gast- und externe Zugriffe**
2. **Umgang mit Daten und Speichermedien**
   - **Verschlüsselung sensibler Daten** (z. B. AES, TLS)
   - Regelungen zur **Datenklassifizierung** (öffentlich, vertraulich, geheim)
   - Sichere **Datenspeicherung und -löschung**
3. **Netzwerksicherheit**
   - Einsatz von **Firewalls, VPNs und Intrusion Detection Systems (IDS)**
   - Regeln für **WLAN-Nutzung und Remote-Zugriff**
   - Schutzmaßnahmen gegen **DDoS- und Man-in-the-Middle-Angriffe**
4. **Endgerätesicherheit**
   - Verwendung von **Antivirensoftware und Endpoint Protection**
   - Regelmäßige **Updates und Patches** für Betriebssysteme und Anwendungen
   - Verbot oder Einschränkung privater Geräte (BYOD-Richtlinien)
5. **Incident Management & Notfallpläne**
   - Meldewege für **Sicherheitsvorfälle und Datenschutzverletzungen**
   - Definition von **Reaktionsmaßnahmen bei Cyberangriffen**
   - Regelmäßige **Backups und Disaster-Recovery-Tests**
6. **Mitarbeitersensibilisierung & Schulungen**
   - **Schulungen zu IT-Sicherheit & Social Engineering**
   - Richtlinien für **E-Mail-Sicherheit (Phishing-Erkennung)**

- Vorgaben für sicheres Arbeiten im Homeoffice

Ziel der IT-Sicherheitsrichtlinien

- **Schutz der IT-Infrastruktur** vor Angriffen und Bedrohungen
- **Einhaltung gesetzlicher Vorgaben** (DSGVO, IT-Sicherheitsgesetz, ISO 27001)
- **Minimierung von Risiken** durch klare Sicherheitsvorgaben

11.6 Personelle Maßnahmen und Entwicklung des Sicherheitsbewusstseins

---

Personelle Maßnahmen sind organisatorische Vorkehrungen zur Sensibilisierung von Mitarbeitern für IT-Sicherheit. Da menschliches Fehlverhalten eine häufige Ursache für Sicherheitsvorfälle ist, sind Schulungen und klare Verhaltensrichtlinien essenziell.

Schulungen

- **Regelmäßige IT-Sicherheitsschulungen** für alle Mitarbeiter
- **Phishing-Tests** zur Sensibilisierung für Social Engineering
- **Workshops zu sicherem Umgang mit Passwörtern & Authentifizierungsmethoden**
- **E-Learning-Programme** für flexibles Lernen

Klare Verhaltensrichtlinien

- Nutzung von **starken Passwörtern und Multi-Faktor-Authentifizierung (MFA)**
- **Keine Weitergabe von Zugangsdaten** oder sensiblen Informationen
- **Sichere Nutzung von E-Mails & Anhängen** (Phishing-Prävention)
- **Regelungen für mobiles Arbeiten & Homeoffice-Sicherheit**

Zugriffskontrollen & Berechtigungskonzepte

- **Least Privilege Principle:** Mitarbeiter erhalten nur notwendige Rechte
- **Rollenbasierte Zugriffskontrollen (RBAC)** zur Einschränkung sensibler Datenzugriffe

- **Dokumentation & Protokollierung von Zugriffen**

Förderung der Sicherheitskultur im Unternehmen

- **Mitarbeiter zu Botschaftern der IT-Sicherheit machen**
- **Meldesysteme für Sicherheitsvorfälle etablieren** (ohne Angst vor Sanktionen)
- **Belohnung sicherheitsbewussten Verhaltens** (z. B. Gamification-Ansätze)

Ziel der personellen Maßnahmen

- **Reduzierung von Sicherheitsrisiken** durch menschliches Fehlverhalten
- **Bewusstseinsbildung für IT-Sicherheit im Arbeitsalltag**
- **Langfristige Stärkung der Sicherheitskultur** im Unternehmen

11.7 BSI IT-Grundschutz-Kompendium

Das **BSI IT-Grundschutz-Kompendium** ist ein Standardwerk des **Bundesamts für Sicherheit in der Informationstechnik (BSI)** zur Umsetzung von **IT-Sicherheitsmaßnahmen** in Unternehmen und Behörden. Es basiert auf einem systematischen Ansatz zur Identifikation und Minimierung von Risiken.

IT-Grundschutz-Methodik

Der IT-Grundschutz umfasst:

- **Basis-Sicherheitsmaßnahmen** für typische IT-Umgebungen
- **Gefährdungskataloge** zur Identifikation von Risiken
- **Schutzbedarfsermittlung** für IT-Systeme und Daten
- **Umsetzungsplanung von Sicherheitsmaßnahmen**

Schichtenmodell des IT-Grundschutzes

Sicherheitsmaßnahmen werden in fünf Schichten unterteilt:

1. **Übergreifende Aspekte** (Sicherheitsmanagement, Personal, Notfallmanagement)
2. **Prozesse und Organisation** (Betriebs- und Sicherheitskonzepte)
3. **Netze und Kommunikation** (Firewalls, VPNs, Netzwerksicherheit)
4. **IT-Systeme** (Clients, Server, Mobile Geräte)

5. **Anwendungen** (Webanwendungen, Cloud-Dienste, E-Mail-Sicherheit)

IT-Grundschutz-Bausteine

Das Kompendium gliedert Sicherheitsmaßnahmen in **Bausteine**, darunter:

- **ORP**: Organisation und Personal (z. B. Sicherheitsrichtlinien, Schulungen)
- **INF**: Infrastruktur (z. B. Rechenzentren, Verkabelung)
- **NET**: Netzwerke (z. B. WLAN-Sicherheit, Firewalls)
- **SYS**: IT-Systeme (z. B. Server, Clients, Virtualisierung)
- **APP**: Anwendungen (z. B. Webanwendungen, Datenbanken)

Schutzmaßnahmen nach IT-Grundschutz

- **Technische Maßnahmen**: Firewalls, Verschlüsselung, Zugriffskontrollen
- **Organisatorische Maßnahmen**: Sicherheitsrichtlinien, Notfallpläne
- **Personelle Maßnahmen**: Sensibilisierung, Schulungen
- **Physische Maßnahmen**: Zutrittskontrollen, Schutz von Rechenzentren

11.8 Datenschutzgesetze

Die Einhaltung der Datenschutzgesetze, insbesondere der **DSGVO** (Datenschutz-Grundverordnung) und des **BDSG** (Bundesdatenschutzgesetz), muss regelmäßig überprüft werden, um Datenschutzverstöße zu vermeiden und gesetzliche Anforderungen zu erfüllen.

Datenschutz-Grundverordnung (DSGVO)

Die **DSGVO** (EU-weit gültig) regelt den Schutz personenbezogener Daten.

Prüfkriterien:

- **Rechtmäßigkeit der Verarbeitung**: Liegt eine **Rechtsgrundlage** vor (z. B. Einwilligung, Vertrag, berechtigtes Interesse)?
- **Datensparsamkeit**: Werden nur die **notwendigen Daten** erhoben?

- **Transparenz:** Werden Betroffene über die **Datenverarbeitung** informiert (Art. 13, 14 DSGVO)?
- **Betroffenenrechte:** Können Personen **Auskunft, Berichtigung oder Löschung** ihrer Daten verlangen?
- **Technisch-organisatorische Maßnahmen (TOM):** Sind angemessene Sicherheitsmaßnahmen implementiert (z. B. Verschlüsselung, Zugriffskontrollen)?
- **Meldung von Datenschutzverletzungen:** Sind Prozesse zur **Meldepflicht bei Datenschutzpannen** (Art. 33 DSGVO) vorhanden?
- **Datenschutz-Folgenabschätzungen (DPIA):** Wurden für risikobehaftete Verarbeitungen Folgenabschätzungen durchgeführt (Art. 35 DSGVO)?
- **Verarbeitungsverzeichnis:** Existiert eine **Dokumentation aller Datenverarbeitungen** im Unternehmen (Art. 30 DSGVO)?
- **Auftragsverarbeitung:** Sind Verträge mit **externen Dienstleistern** datenschutzkonform (Art. 28 DSGVO)?

Bundesdatenschutzgesetz (BDSG)

Das **BDSG** ergänzt die DSGVO und enthält nationale Datenschutzregelungen für Deutschland.

Prüfkriterien:

- **Bestellung eines Datenschutzbeauftragten:** Ist ein **DSB benannt**, falls erforderlich (§ 38 BDSG)?
- **Besonderer Schutz sensibler Daten:** Werden besonders schützenswerte Daten (z. B. Gesundheitsdaten, biometrische Daten) gemäß **§ 22 BDSG** gesichert?
- **Beschäftigtendatenschutz:** Werden Daten von Mitarbeitern nur im **zulässigen Rahmen** verarbeitet (§ 26 BDSG)?
- **Videoüberwachung:** Werden die **rechtlichen Vorgaben für Kameraüberwachung** eingehalten (§ 4 BDSG)?

11.8 Personenbezogene Daten

Personenbezogene Daten sind gemäß **Art. 4 Nr. 1 DSGVO** alle Informationen, die sich auf eine **identifizierte oder identifizierbare natürliche Person** beziehen.

Beispiele für personenbezogene Daten

Direkte Identifikation:

- Name, Vorname
- Adresse
- Telefonnummer
- E-Mail-Adresse (z. B. max.mustermann@example.com)

Indirekte Identifikation:

- IP-Adresse
- Kundennummer
- Standortdaten
- Cookie-IDs

Besondere Kategorien personenbezogener Daten (Art. 9 DSGVO):

- Gesundheitsdaten
- Biometrische Daten (Fingerabdruck, Gesichtserkennung)
- Politische Meinungen
- Religiöse Überzeugungen

11.10 Rechte der Betroffenen und Konsequenzen der Einwilligung

---

Die **DSGVO** räumt betroffenen Personen umfassende Rechte ein, um die Kontrolle über ihre personenbezogenen Daten zu gewährleisten. Eine Verarbeitung personenbezogener Daten ist nur zulässig, wenn eine **Rechtsgrundlage** vorliegt, darunter die **Einwilligung der betroffenen Person** (Art. 6 Abs. 1 lit. a DSGVO).

Rechte der Betroffenen (Art. 12–22 DSGVO)

Recht auf Information (Art. 13, 14 DSGVO)

- Betroffene müssen über die **Datenverarbeitung, Zwecke und Rechtsgrundlagen** informiert werden.
- Dies erfolgt in der **Datenschutzerklärung** oder durch direkte Mitteilung.

Recht auf Auskunft (Art. 15 DSGVO)

- Betroffene können eine **Kopie ihrer gespeicherten Daten** anfordern.
- Unternehmen müssen zusätzlich angeben, **woher die Daten stammen, an wen sie weitergegeben wurden und wie lange sie gespeichert werden.**

Recht auf Berichtigung (Art. 16 DSGVO)

- Unrichtige oder unvollständige Daten müssen auf Antrag berichtigt werden.

Recht auf Löschung („Recht auf Vergessenwerden“, Art. 17 DSGVO)

- Personen können die **Löschung ihrer Daten** verlangen, wenn
  - die Daten nicht mehr benötigt werden,
  - die Einwilligung widerrufen wurde oder
  - eine unrechtmäßige Verarbeitung vorliegt.
- Ausnahmen bestehen, wenn eine gesetzliche Pflicht zur Aufbewahrung besteht.

Recht auf Einschränkung der Verarbeitung (Art. 18 DSGVO)

- Daten dürfen nur noch gespeichert, aber nicht mehr verarbeitet werden, wenn z. B. die Richtigkeit überprüft wird.

Recht auf Datenübertragbarkeit (Art. 20 DSGVO)

- Betroffene können verlangen, dass ihre Daten in einem **maschinenlesbaren Format** bereitgestellt oder an einen anderen Anbieter übertragen werden.

Widerspruchsrecht (Art. 21 DSGVO)

- Verarbeitung kann **jederzeit aus persönlichen Gründen** untersagt werden, es sei denn, es gibt zwingende schutzwürdige Gründe.

Recht auf keine automatisierte Entscheidung (Art. 22 DSGVO)

- Personen dürfen nicht allein durch **automatisierte Entscheidungen** benachteiligt werden (z. B. Kreditbewertung durch Algorithmen).

Konsequenzen der Einwilligung (Art. 7 DSGVO)

Die Einwilligung ist eine **freiwillige, informierte und unmissverständliche Zustimmung** zur Datenverarbeitung.

Voraussetzungen einer gültigen Einwilligung

- **Freiwilligkeit:** Keine Zwangslage oder Nachteile bei Verweigerung.
- **Informiertheit:** Klare Erklärung zu Zweck, Art und Umfang der Verarbeitung.
- **Eindeutige Handlung:** Opt-in-Verfahren, keine vorausgewählten Checkboxen.
- **Widerrufbarkeit:** Die Einwilligung kann **jederzeit ohne Nachteile widerrufen** werden.

Ungültige Einwilligung

- Wenn die betroffene Person **nicht ausreichend informiert wurde.**
- Wenn sie **nicht aktiv zugestimmt hat** (z. B. durch vorangekreuzte Felder).
- Wenn sie **nicht freiwillig gegeben wurde** (z. B. bei Abhängigkeitsverhältnissen).

Konsequenzen bei fehlender oder unwirksamer Einwilligung

- Die Verarbeitung ist rechtswidrig und kann Bußgelder nach **Art. 83 DSGVO** zur Folge haben.
- Betroffene können **Beschwerde bei der Aufsichtsbehörde einlegen** und **Schadensersatz fordern.**

11.11 Anonymisierung

Anonymisierung ist ein Verfahren zur **unwiderruflichen Entfernung des Personenbezugs** aus Daten. Nach der Anonymisierung können die Daten keiner bestimmten Person mehr zugeordnet werden und unterliegen nicht mehr der **DSGVO.**

Merkmale

- Keine Identifikation der betroffenen Person möglich
- Keine Rückführung der Daten auf die Person

- Dauerhafter Verlust des Personenbezugs

Methoden der Anonymisierung

- **Datenaggregation:** Zusammenfassung von Einzelwerten zu Gruppen (z. B. Durchschnittswerte statt individueller Daten).
- **Maskierung:** Ersetzung bestimmter Merkmale (z. B. Namen durch „XXX“).
- **Rauschen (Noise Addition):** Zufällige Veränderung der Daten, um Identifikation zu verhindern.
- **Generalization:** Reduzierung der Genauigkeit (z. B. „Alter 25“ → „Altersgruppe 20-30“).
- **k-Anonymität:** Daten sind nur dann freigegeben, wenn mindestens **k Personen** dieselben Merkmale teilen.

11.12 Pseudonymisierung

Pseudonymisierung ist ein Datenschutzverfahren, bei dem personenbezogene Daten so verändert werden, dass sie **ohne zusätzliche Informationen** nicht mehr einer bestimmten Person zugeordnet werden können. Die Zuordnung kann jedoch durch Verwendung eines Schlüssels wiederhergestellt werden.

Merkmale

- Der direkte Bezug zur Person wird durch **Ersetzung oder Verschlüsselung** entfernt.
- Die ursprünglichen Daten bleiben erhalten und sind durch einen **separaten Schlüssel rekonstruierbar**.
- Die Daten unterliegen weiterhin der **DSGVO** (Art. 4 Nr. 5 DSGVO), da die Identifikation theoretisch möglich bleibt.

Methoden der Pseudonymisierung

- **Ersatz durch Kennungen:** Zuweisung zufälliger Identifikatoren (z. B. Kundennummer statt Name).
- **Hashing:** Umwandlung von Daten in einen Hash-Wert (z. B. SHA-256), ohne Rückführungsmöglichkeit.
- **Tokenisierung:** Austausch sensibler Daten durch zufällige Token, die separat gespeichert werden.

- **Verschlüsselung:** Speicherung der Originaldaten in verschlüsselter Form mit separatem Schlüssel.

11.13 Schutzbedarfsanalyse

Die **Schutzbedarfsanalyse** bewertet die Sicherheitsanforderungen von **Daten, IT-Systemen und Infrastruktur**, um angemessene Schutzmaßnahmen abzuleiten.

Durchführung

Eine Schutzbedarfsanalyse erfolgt in mehreren Schritten:

1. **Identifikation der Schutzobjekte**
   - Welche **Daten, Anwendungen, IT-Systeme, Räume oder Kommunikationsverbindungen** müssen geschützt werden?
2. **Bewertung des Schutzbedarfs**
   - Welche Auswirkungen hätte ein **Datenverlust, ein Ausfall oder eine Manipulation**?
   - Schutzbedarf wird meist in **niedrig, mittel und hoch** eingeteilt.
3. **Ableitung von Schutzmaßnahmen**
   - Maßnahmen anhand des Schutzbedarfs auswählen (z. B. Verschlüsselung, Firewalls, Zugriffskontrollen).
4. **Dokumentation & Umsetzung**
   - Ergebnisse dokumentieren und **regelmäßig überprüfen** (z. B. durch Audits).

Schutzbedarfsanalyse für verschiedene Bereiche

Anwendungen

- Bewertung der **Kritikalität** einer Software (z. B. ERP-System, Webanwendung).
- Anforderungen an **Datenintegrität, Verfügbarkeit und Vertraulichkeit**.
- Schutz vor **Manipulation, Datenverlust und unbefugtem Zugriff**.

IT-Systeme

- Identifikation kritischer **Server, Datenbanken, Endgeräte**.
- Bewertung von **Ausfallrisiken und Angriffsszenarien**.

- Maßnahmen wie **Backup-Strategien, Firewalls, Zugriffskontrollen.**

Räume und Infrastruktur

- Schutz sensibler Bereiche (z. B. **Rechenzentren, Büros mit vertraulichen Daten**).
- Maßnahmen wie **Zutrittskontrollen, Videoüberwachung, Brandschutz.**

Kommunikationsverbindungen

- Bewertung von **Netzwerkverbindungen, VPNs, Cloud-Diensten.**
- Schutz durch **Ende-zu-Ende-Verschlüsselung, Firewalls, IDS/IPS.**

11.14 Arbeitsplatzbezogenes Sicherheitskonzept

Ein **arbeitsplatzbezogenes Sicherheitskonzept** definiert Maßnahmen zum Schutz von IT-Arbeitsplätzen gegen **technische, organisatorische und menschliche Risiken.**

Schutzbedarfskategorien

Der **Schutzbedarf eines Arbeitsplatzes** richtet sich nach den **Auswirkungen eines Sicherheitsvorfalls:**

- **Normal** → Keine gravierenden Folgen, Standardmaßnahmen ausreichend.
- **Hoch** → Betrifft vertrauliche Daten oder geschäftskritische Systeme, erweiterte Sicherheitsmaßnahmen nötig.
- **Sehr hoch** → Schwerwiegende wirtschaftliche oder rechtliche Konsequenzen, maximale Schutzmaßnahmen erforderlich.

Risiko-Klassifikation

Ein Arbeitsplatz wird anhand möglicher **Bedrohungen und Auswirkungen** klassifiziert:

Risiko	Beispiel	Schutzmaßnahme
Gering	Kein Zugriff auf interne Systeme	Standard-Passwortschutz
Mittel	Zugriff auf Kundendaten	2FA, Verschlüsselung
Hoch	Verarbeitung sensibler Daten	Härtung, Netzwerksegmentierung, Monitoring

11.15 ISMS

Ein **ISMS (Information Security Management System)** ist ein systematischer Ansatz zur **Planung, Umsetzung, Überwachung und Verbesserung der Informationssicherheit** in einer Organisation. Es basiert auf definierten **Richtlinien, Prozessen und technischen Maßnahmen.**

Bestandteile

- **Sicherheitsrichtlinien:** Vorgaben zur IT-Sicherheit und Datenschutz.
- **Risikomanagement:** Identifikation, Bewertung und Behandlung von Sicherheitsrisiken.
- **Technische & organisatorische Maßnahmen:** Firewalls, Verschlüsselung, Schulungen.
- **Überwachung & Audits:** Regelmäßige Prüfungen zur Einhaltung der Sicherheitsstandards.
- **Kontinuierliche Verbesserung:** Anpassung an neue Bedrohungen und Anforderungen.

Relevante Standards

- **ISO 27001:** Internationaler Standard für ISMS-Zertifizierung.
- **BSI IT-Grundschutz:** Deutscher Standard für IT-Sicherheit.

Vorteile

- **Schutz sensibler Daten** vor Cyberangriffen und Datenverlust.
- **Einhaltung gesetzlicher Anforderungen** (z. B. DSGVO, IT-Sicherheitsgesetz).

- **Erhöhung des Sicherheitsbewusstseins** durch klare Prozesse und Schulungen.
- **Verbesserung der Geschäftskontinuität** durch systematisches Risikomanagement.

11.16 Security by Design

Security by Design ist ein Konzept, bei dem IT-Systeme, Software und Prozesse **von Anfang an** mit Sicherheitsmechanismen ausgestattet werden. Es basiert auf der Idee, Sicherheitsrisiken **proaktiv** zu minimieren, anstatt sie nachträglich zu beheben.

Prinzipien

- **Minimaler Zugriff (Least Privilege Principle):** Benutzer und Prozesse erhalten nur die minimal notwendigen Rechte.
- **Standardmäßig sicher (Secure Defaults):** Voreinstellungen sollten **sicher** sein (z. B. starke Passwörter, deaktivierte unsichere Funktionen).
- **Fehlertoleranz (Fail-Secure):** Systeme sollen auch bei Fehlern sicher bleiben, z. B. durch automatische Sperrmechanismen.
- **Trennung von Funktionen (Separation of Duties):** Kritische Aktionen erfordern **mehrere Berechtigungen** oder Bestätigungen.
- **Ende-zu-Ende-Verschlüsselung:** Schutz der Daten während Speicherung und Übertragung.
- **Sichere Entwicklungsmethoden:** Secure Coding-Praktiken (z. B. OWASP Top 10 zur Vermeidung von Schwachstellen).

Umsetzung

- **Bedrohungsanalysen bereits in der Planungsphase durchführen.**
- **Sicherheitsmechanismen in den gesamten Software-Development-Lifecycle (SDLC) integrieren.**
- **Automatisierte Sicherheitstests und regelmäßige Code-Reviews nutzen.**
- **Security Audits und Penetrationstests durchführen**, um Schwachstellen frühzeitig zu erkennen.

Vorteile

- Geringere **Kosten für nachträgliche Sicherheitsmaßnahmen.**

- Schutz vor **Datenlecks, Cyberangriffen und Compliance-Verstößen.**
- Verbesserung des **Vertrauens der Nutzer** durch sichere Systeme.

11.17 Security by Default

Security by Default bedeutet, dass IT-Systeme und Software **standardmäßig mit sicheren Einstellungen ausgeliefert werden.** Nutzer müssen Sicherheitsmechanismen nicht manuell aktivieren, sondern erhalten bereits ab Werk eine **sichere Konfiguration.**

Prinzipien

- **Sichere Voreinstellungen:** Standardmäßig sind **nur sichere Funktionen aktiviert**, unsichere Features sind deaktiviert.
- **Minimaler Zugriff (Least Privilege):** Benutzer und Prozesse erhalten nur die **notwendigsten Rechte.**
- **Deaktivierung unsicherer Funktionen:** Nicht benötigte **Dienste, Ports oder Schnittstellen** sind standardmäßig deaktiviert.
- **Starke Authentifizierung:** Vorgabe sicherer **Passwörter, Multi-Faktor-Authentifizierung (MFA).**
- **Automatische Updates:** Sicherheitsupdates werden standardmäßig aktiviert und regelmäßig installiert.

Umsetzung

- Betriebssysteme liefern standardmäßig **aktivierte Firewalls und Verschlüsselung.**
- Webbrowser blockieren **unsichere Inhalte** und setzen HTTPS voraus.
- Cloud-Dienste aktivieren **Datenverschlüsselung und Zugriffsbeschränkungen.**
- IoT-Geräte setzen **starke Passwörter und verschlüsselte Kommunikation** als Standard.

Vorteile

- **Reduzierung von Sicherheitsrisiken**, da Nutzer nicht manuell Einstellungen anpassen müssen.
- **Bessere Benutzerfreundlichkeit**, da Sicherheit ohne zusätzliches Eingreifen gewährleistet ist.

- **Erfüllung von Compliance-Vorgaben** (z. B. DSGVO, ISO 27001) durch sichere Grundkonfiguration.

11.18 Härtung des Betriebssystems

Die Härtung eines Betriebssystems (OS Hardening) umfasst Maßnahmen zur **Reduzierung von Sicherheitsrisiken** durch das Deaktivieren unnötiger Dienste, das Einschränken von Berechtigungen und die Implementierung von Sicherheitsmechanismen.

Maßnahmen

Minimierung der Angriffsfläche

- **Nicht benötigte Dienste und Ports deaktivieren**
- **Überflüssige Benutzerkonten entfernen oder sperren**
- **Unnötige Software und Komponenten deinstallieren**

Starke Zugriffskontrollen

- **Prinzip der geringsten Rechte (Least Privilege Principle)** umsetzen
- **Passwort-Richtlinien** durchsetzen (z. B. Mindestlänge, Ablaufdatum)
- **Multi-Faktor-Authentifizierung (MFA) aktivieren**

System- und Netzwerksicherheit

- **Host-Firewall aktivieren und konfigurieren**
- **Antiviren- und Endpoint-Schutzlösungen einsetzen**
- **Protokollierung (Logging) und Überwachung aktivieren**

Software- und Patch-Management

- **Regelmäßige Sicherheitsupdates installieren**
- **Automatische Updates aktivieren, wenn möglich**
- **Software nur aus vertrauenswürdigen Quellen installieren**

Daten- und Speicherschutz

- **Festplattenverschlüsselung (z. B. BitLocker, LUKS) aktivieren**
- **USB- und externe Speichermedien einschränken oder deaktivieren**

- **Zugriffsrechte auf sensible Dateien und Verzeichnisse einschränken**

Automatisierte Härtung

- **Security Baselines** (z. B. CIS Benchmarks, BSI IT-Grundschutz) verwenden
- **Härtungsskripte und Tools nutzen** (z. B. Microsoft Security Compliance Toolkit, Lynis für Linux)

Vorteile

- **Schutz vor Cyberangriffen, Malware und Exploits**
- **Minimierung von Sicherheitslücken durch reduzierte Angriffsfläche**
- **Erfüllung von Compliance-Anforderungen (z. B. DSGVO, ISO 27001)**

11.19 Datensicherungsverfahren

Datensicherung (Backup) ist eine zentrale Maßnahme zur **Wiederherstellung von Daten** im Falle von Hardware-Ausfällen, Cyberangriffen oder menschlichen Fehlern.

Backup-Arten

Voll-Backup (Full Backup)

- Es wird eine **komplette Kopie aller Daten** erstellt.
- **Vorteil:** Schnelle Wiederherstellung.
- **Nachteil:** Hoher Speicherbedarf und lange Sicherungsdauer.

Inkrementelles Backup

- Es werden nur **die seit dem letzten Backup geänderten Daten** gesichert.
- **Vorteil:** Schnelle Sicherung, wenig Speicherverbrauch.
- **Nachteil:** Wiederherstellung dauert länger, da mehrere Backup-Stände benötigt werden.

Differenzielles Backup

- Sichert alle **seit dem letzten Voll-Backup geänderten Dateien.**

- **Vorteil:** Schnellere Wiederherstellung als beim inkrementellen Backup.
- **Nachteil:** Benötigt mehr Speicherplatz als inkrementelle Backups.

Spiegelung (Mirroring)

- Echtzeit-Kopie der Daten auf ein zweites Speichermedium.
- **Vorteil:** Sofortige Verfügbarkeit der Daten.
- **Nachteil:** Kein Schutz vor logischen Fehlern oder Malware, da fehlerhafte Daten ebenfalls gespiegelt werden.

Backup-Strategien

3-2-1-Backup-Regel

- **3 Kopien** der Daten (Original + zwei Backups).
- **2 verschiedene Medien** (z. B. externe Festplatte und Cloud).
- **1 Kopie extern** (z. B. Offsite-Backup in einem anderen Rechenzentrum).

Cold, Warm & Hot Backups

- **Cold Backup:** Offline-Sicherung, benötigt manuelle Wiederherstellung.
- **Warm Backup:** Schnell aktivierbare Sicherung mit regelmäßigen Updates.
- **Hot Backup:** Ständig synchronisierte Live-Kopie für sofortige Nutzung.

Air-Gapped Backup

- Datensicherung auf einem **nicht mit dem Netzwerk verbundenen System** als Schutz vor Ransomware.

Speicherorte für Backups

- **Lokale Backups:** Externe Festplatten, NAS-Systeme.
- **Cloud-Backups:** Online-Speicher (z. B. AWS, Microsoft Azure, Google Drive).
- **Bandlaufwerke:** Langlebig, aber langsamer Zugriff.

Wichtige Backup-Anforderungen

- **Regelmäßige Backups** (z. B. täglich, wöchentlich, monatlich).
- **Automatisierung** zur Vermeidung menschlicher Fehler.
- **Verschlüsselung und Zugriffsschutz** für vertrauliche Daten.
- **Regelmäßige Tests der Wiederherstellung (Disaster Recovery Tests)**

11.20 Verschlüsselungstechniken

Verschlüsselung dient dem Schutz von Daten vor unbefugtem Zugriff, indem sie in eine unlesbare Form umgewandelt werden. Es gibt drei Hauptarten: **symmetrische, asymmetrische und hybride Verschlüsselung**.

Symmetrische Verschlüsselung

Bei der **symmetrischen Verschlüsselung** wird für **Ver- und Entschlüsselung derselbe Schlüssel** verwendet.

Beispiele:

- AES (Advanced Encryption Standard)
- DES (Data Encryption Standard, veraltet)

Vorteile:

- Sehr schnell und effizient, da nur ein Schlüssel verwendet wird.
- Gut geeignet für große Datenmengen.

Nachteile:

- Sichere **Schlüsselverteilung** ist problematisch.
- Falls der Schlüssel kompromittiert wird, sind die Daten nicht mehr sicher.

Asymmetrische Verschlüsselung

Bei der **asymmetrischen Verschlüsselung** gibt es **zwei Schlüssel**:

- **Öffentlicher Schlüssel**: Zum Verschlüsseln von Daten.
- **Privater Schlüssel**: Zum Entschlüsseln der Daten.

Beispiele:

- RSA (Rivest-Shamir-Adleman)
- ECC (Elliptic Curve Cryptography)

Vorteile:

- Sichere Kommunikation ohne vorherigen Schlüsselaustausch.
- Ermöglicht digitale Signaturen zur Integritätsprüfung.

Nachteile:

- Langsamer als symmetrische Verschlüsselung.
- Hoher Rechenaufwand.

Hybride Verschlüsselung

Hybride Verschlüsselung kombiniert die Vorteile von **symmetrischer und asymmetrischer Verschlüsselung.**

Funktionsweise:

1. Eine **asymmetrische Methode** (z. B. RSA) wird genutzt, um einen **einmaligen symmetrischen Schlüssel** sicher zu übertragen.
2. Die eigentlichen Daten werden mit einer **schnellen symmetrischen Methode** (z. B. AES) verschlüsselt.

Beispiele:

- TLS (Transport Layer Security, z. B. bei HTTPS)
- PGP (Pretty Good Privacy) für E-Mail-Verschlüsselung

Vorteile:

- Hohe Sicherheit durch asymmetrische Schlüsselverteilung.
- Effiziente Datenverschlüsselung durch symmetrische Verfahren.

Nachteile:

- Komplexer als rein symmetrische oder asymmetrische Verschlüsselung.

11.21 Hashwerte

Ein **Hashwert** ist ein eindeutiger Fingerprint, der durch eine **Hashfunktion** aus Daten berechnet wird. Er dient zur **Integritätsprüfung und Authentifizierung.**

Eigenschaften:

- **Einwegfunktion:** Nicht umkehrbar.
- **Kollisionsresistenz:** Unterschiedliche Daten erzeugen unterschiedliche Hashwerte.
- **Deterministisch:** Gleiche Eingabe → gleicher Hashwert.

Gängige Algorithmen:

- **Sicher:** SHA-256, SHA-3, Argon2 (für Passwörter).
- **Veraltet:** MD5, SHA-1 (anfällig für Kollisionen).

11.22 Zertifikate

Ein **Zertifikat** ist eine digitale Bescheinigung, die die **Echtheit einer Identität** bestätigt. Es wird von einer **Zertifizierungsstelle (CA, Certificate Authority)** ausgestellt und dient vor allem der **verschlüsselten Kommunikation**.

Bestandteile

- Öffentlicher Schlüssel des Inhabers
- Name des Inhabers (z. B. Domainname)
- Name der Zertifizierungsstelle (CA)
- Gültigkeitsdauer
- Digitale Signatur der CA

Arten

- **TLS/SSL-Zertifikate:** Sicherung von Webseiten (HTTPS).
- **Code-Signing-Zertifikate:** Verifizierung von Software.
- **E-Mail-Zertifikate (S/MIME):** Signierung und Verschlüsselung von E-Mails.
- **Client-/Benutzerzertifikate:** Authentifizierung von Nutzern.

Vertrauenshierarchie (Public Key Infrastructure – PKI)

1. **Root-CA** (höchste Vertrauensinstanz)
2. **Intermediate-CA** (stellt Zertifikate für Endnutzer aus)
3. **Endnutzer-Zertifikat** (z. B. für eine Webseite)

Anwendung & Sicherheit

- **Authentifizierung:** Prüft die Identität einer Website oder Person.
- **Verschlüsselung:** Sicherstellung vertraulicher Kommunikation.
- **Digitale Signaturen:** Schutz vor Manipulation.

11.23 Digitale Signaturen

Eine **digitale Signatur** stellt die **Echtheit und Integrität** digitaler Daten sicher. Sie basiert auf **asymmetrischer Kryptografie** und wird zur **Verifizierung von Dokumenten, Software und E-Mails** verwendet.

Funktionsweise

1. **Hash-Wert berechnen** → Der Inhalt wird gehasht (z. B. mit SHA-256).
2. **Verschlüsselung mit privatem Schlüssel** → Der Hash wird mit dem **privaten Schlüssel des Signierenden** verschlüsselt.
3. **Verifikation mit öffentlichem Schlüssel** → Der Empfänger entschlüsselt die Signatur mit dem **öffentlichen Schlüssel** des Absenders.

Eigenschaften

- **Authentizität:** Prüft die Identität des Absenders.
- **Integrität:** Erkennt Manipulationen am Inhalt.
- **Nichtabstreitbarkeit:** Der Absender kann die Unterschrift nicht leugnen.

Anwendungen

- **E-Mail-Signaturen (S/MIME, PGP)**
- **Signierte Software (Code-Signing)**
- **Digitale Dokumente (PDF-Signaturen, X.509-Zertifikate)**
- **Blockchain & Kryptowährungen**

11.24 Authentifizierung

Authentifizierung ist der Prozess der **Überprüfung einer Identität** durch einen oder mehrere Nachweise.

Zwei-Faktor-Authentifizierung (2FA)

Bei der **2FA** werden **zwei verschiedene Faktoren** kombiniert:

1. **Wissen** (z. B. Passwort, PIN)
2. **Besitz** (z. B. Smartphone, Token)
3. **Biometrie** (z. B. Fingerabdruck, Gesichtserkennung)

Beispiele:

- SMS-TAN oder Authenticator-App zusätzlich zum Passwort
- Smartcard oder Hardware-Token

Vorteile:

- Erhöht die Sicherheit, da **ein einzelner Faktor nicht ausreicht.**
- Schutz gegen **Phishing und Passwortdiebstahl.**

Passwort-Policy

Regeln zur **sicheren Vergabe und Verwaltung** von Passwörtern.

Best Practices:

- **Mindestlänge:** Mindestens 12–16 Zeichen.
- **Komplexität:** Kombination aus Groß- und Kleinbuchstaben, Zahlen, Sonderzeichen.
- **Kein Wiederverwenden** alter Passwörter.
- **Kein Teilen oder Speichern in unsicheren Orten.**
- **Passwort-Manager nutzen**, um sichere Passwörter zu generieren.

11.25 Personal Firewall

Eine **Personal Firewall** ist eine Software, die den **Netzwerkverkehr auf einem einzelnen Gerät** überwacht und unerwünschte Verbindungen blockiert. Sie schützt vor **Cyberangriffen, Schadsoftware und unautorisierten Zugriffen.**

Funktionen

- **Paketfilterung:** Blockiert unerwünschten ein- und ausgehenden Datenverkehr.

- **Anwendungssteuerung:** Bestimmt, welche Programme auf das Netzwerk zugreifen dürfen.
- **Intrusion Detection/Prevention:** Erkennt und verhindert Angriffe in Echtzeit.

Vorteile

- Schutz vor **Hackerangriffen und Schadsoftware.**
- Kontrolle über **Netzwerkaktivitäten installierter Programme.**
- Ergänzt **Antivirensoftware** für besseren Schutz.