Inhaltsverzeichnis

1 Datenbanken

1.1 Prinzipien

ACID ACID steht für die vier englischen Einzelbegriffe Atomicity, Consistency, Isolation und Durability und ist eine gängige Abkürzung der Informationstechnik. Im Deutschen lauten die vier Begriffe Atomarität, Konsistenz, Isolation und Dauerhaftigkeit. Oft wird im deutschsprachigen Raum das Akronym AKID verwendet. Das ACID-Prinzip stellt Regeln auf, wie mit Transaktionen in Datenbankmanagementsystemen zu verfahren ist, um für verlässliche, konsistente Daten und Systeme zu sorgen.

Eine Transaktion besteht aus einer Folge verschiedener Vorgänge, die diese ACID-Regeln einzuhalten haben. Geprägt wurde das Akronym ACID bereits im Jahr 1983 von den beiden Informatikern Andreas Reuter und Theo Härder. In Normen wie ISO/IEC 10.026-1:1992 oder ISO/IEC 10.026-1:1998 Abschnitt 4 ist das ACID-Prinzip beschrieben. Zur Umsetzung des ACID-Prinzips kommen Transaktions-Manager und Logging-Mechanismen zum Einsatz.

Die vier ACID-Grundprinzipien

Das ACID-Konzept besteht aus vier einzelnen Grundprinzipien. Diese Grundprinzipien lauten:

- Atomicity oder Atomarität: Ausführung aller oder keiner Informationsteile einer Transaktion
- Consistency oder Konsistenz: Transaktionen erzeugen einen gültigen Zustand oder fallen in den alten Zustand zurück
- Isolation oder Abgrenzung: Transaktionen verschiedener Anwender oder Prozesse bleiben voneinander isoliert
- Durability oder Dauerhaftigkeit: Nach einer erfolgreichen Transaktion bleiben die Daten dauerhaft gespeichert

IHK Belegsatz [19]

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Spaltenname <DATENTYP>, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname **ADD COLUMN** Spaltenname Datentyp **DROP COLUMN** Spaltenname Datentyp **ADD FOREIGN KEY**(Spaltenname) **REFERENCES** Tabellenname(Primärschlüssel)	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * | Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT ... **FROM ...** **(SELECT ...** **FROM ...** **WHERE ...) AS** tbl **WHERE ...**	Unterabfrage (subquery), die in eine äußere Abfrage eingebettet ist. Das Ergebnis der Unterabfrage wird wie eine Tabelle - hier mit Namen 'tbl' - behandelt.

SELECT **DISTINCT**	Eliminiert Redundanzen, die in einer Tabelle auftreten können. Werte werden jeweils nur einmal angezeigt.
JOIN / **INNER JOIN**	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / **LEFT OUTER JOIN**	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / **RIGHT OUTER JOIN**	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery) **WHERE NOT EXISTS** (subquery)	Die Bedingung EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NO EXISTS negiert die Bedingung.
WHERE ... IN (subquery) **WHERE NOT ... IN** (subquery)	Der Wert des Datenfelds ist in der ausgewählten Menge vorhanden. Der Wert des Datenfelds ist in der ausgewählten Menge nicht vorhanden.
GROUP BY Spaltenname1 [, Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [, Spaltenname2, ...] **ASC** | **DESC**	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname **SET**	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellenname [(spalte1, spalte2, ...)] **VALUES** (Wert Spalte 1 [, Wert Spalte 2, ...]) oder **SELECT ... FROM ... WHERE**	Fügt Datensätze in die genannte Tabelle ein, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind
Berechtigungen kontrollieren	
CREATE Benutzer | Rolle **IDENTIFIED BY** 'Passwort'	Erzeugt einen neuen Benutzer oder eine neue Rolle mit einem Passwort

GRANT Recht | Rolle **ON** *.* | Datenbank.* | Datenbank.Objekt **TO** Benutzer | Rolle [WITH GRANT OPTION]	Weist einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt Weist einem Benutzer eine Rolle zu
REVOKE Recht | Rolle **ON** *.* | Datenbank.* | Datenbank.Objekt **FROM** Benutzer | Rolle	Entzieht einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt Entzieht einem Benutzer eine Rolle
Aggregatfunktionen	
AVG (Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT (Spaltenname | *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM (Spaltenname | Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN (Spaltenname | Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname | Formel)	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT (Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von links.
RIGHT (Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von rechts.
CURRENT	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT(time,[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE (Wert)	Wandelt einen Wert in ein Datum um
DAY (Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH (Datum)	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY (Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
YEAR (Datum)	Liefert das Jahr aus dem angegebenen Datum
DATEADD (Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgefrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF (Datumsteil, Anfangsdatum, Enddatum) Datumsteile: **DAY, MONTH, YEAR**	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)

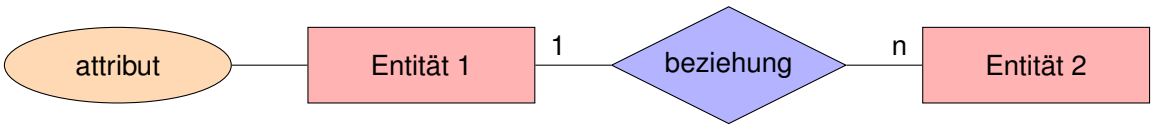
Operatoren	
AND	Logisches UND
LIKE	Überprüfung von Text auf Gleichheit wenn Platzhalter (’regular expressions’) eingesetzt werden.
NOT	Logische Negation
OR	Logisches ODER
IS NULL	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, <>	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeigen

1.2 Datenbankmodelle und -modellierung

1.2.1 Relationale und nicht-relationale Datenbanken

Eine nicht relationale Datenbank ist eine Datenbank, die nicht das tabellarische Schema mit Zeilen und Spalten verwendet, das in den meisten herkömmlichen Datenbanksystemen zum Einsatz kommt. Nicht relationale Daten verwenden stattdessen ein Speichermodell, das für die spezifischen Anforderungen des gespeicherten Datentyps optimiert ist. So können die Daten beispielsweise als einfache Schlüssel-Wert-Paare, als JSON-Dokumente oder als Diagramm mit Edges und Scheitelpunkten gespeichert werden. [24]

1.2.2 ERD (Entity-Relationship-Modell)



Kardinalitäten:

Entitäten besitzen unterschiedliche Kardinalitäten, also die Anzahl zuordenbarer Objekte einer anderen Entität. Es gibt die Ausprägungen 1:1 (eins zu eins), 1:n (eins zu mehreren) und n:m (mehrere zu mehreren). [10]

1.2.3 NoSQL

NoSQL-Datenbanken wurden speziell für bestimmte Datenmodelle entwickelt und speichern Daten in flexiblen Schemata, die sich leicht für moderne Anwendungen skalieren lassen. NoSQL-Datenbanken sind für ihre einfache Entwicklung, Funktionalität und Skalierbarkeit weithin bekannt. [2]

Flexibilität NoSQL-Datenbanken bieten in der Regel flexible Schemata, die eine schnellere und iterativere Entwicklung ermöglichen. Das flexible Datenmodell macht NoSQL-Datenbanken ideal für halbstrukturierte und unstrukturierte Daten.

Skalierbarkeit NoSQL-Datenbanken sind in der Regel so konzipiert, dass sie durch die Verwendung von verteilten Hardware-Clustern skaliert werden können, im Gegensatz zu einer Skalierung durch das Hinzufügen teurer und robuster Server. Einige Cloud-Anbieter übernehmen diese Vorgänge im Hintergrund als vollständig verwaltete Dienstleistung.

Hohe Leistung NoSQL-Datenbanken sind für bestimmte Datenmodelle und Zugriffsmuster optimiert. Diese ermöglichen eine höhere Leistung, als wenn Sie versuchen würden, ähnliche Funktionen mit relationalen Datenbanken zu erreichen.

Hochfunktionell NoSQL-Datenbanken bieten hochfunktionelle APIs und Datentypen, die speziell für ihre jeweiligen Datenmodelle entwickelt wurden.

1.2.4 Welche Arten von NoSQL-Datenbanken gibt es?

Schlüsselwertdatenbanken Eine Schlüsselwertdatenbank speichert Daten als eine Sammlung von Schlüsselwertpaaren, in denen ein Schlüssel als eindeutiger Identifikator dient. Schlüssel und Werte können alles sein, von einfachen Objekten bis hin zu komplexen zusammengesetzten Objekten. Anwendungsfälle wie Gaming, Werbung und IoT eignen sich besonders gut für das Schlüssel-Werte-Datenspeicherungsmodell.

Dokumentdatenbanken Dokumentdatenbanken verfügen über das gleiche Dokumentmodellformat, das Entwickler in ihrem Anwendungscode verwenden. Sie speichern Daten als JSON-Objekte, die flexibel, halbstrukturiert und hierarchisch aufgebaut sind. Aufgrund des flexiblen, semi-strukturierten und hierarchischen Aufbaus der Dokumente und Dokumentdatenbanken können diese entsprechend den Anforderungen der Anwendungen weiterentwickelt werden. Das Dokumentdatenbankmodell eignet sich gut für Kataloge, Benutzerprofile und Content-Management-Systeme, bei denen jedes Dokument einzigartig ist und sich im Laufe der Zeit weiterentwickelt.

Graphdatenbanken Der Zweck einer Graphdatenbank besteht darin, das Entwickeln und Ausführen von Anwendungen zu vereinfachen, die mit hochgradig verbundenen Datensätzen arbeiten. Sie verwenden Knoten zur Speicherung von Dateneinheiten und Edges zur Speicherung von Beziehungen zwischen Einheiten. Ein Edge hat immer einen Startknoten, einen Endknoten, einen Typ und eine Richtung. Er kann Eltern-Kind-Beziehungen, Aktionen, Besitzverhältnisse und Ähnliches beschreiben. Die Anzahl und Art der Beziehungen in einem Knoten ist nicht beschränkt. Sie können eine Graphdatenbank verwenden, um Anwendungen zu erstellen und auszuführen, die mit stark verbundenen Datensätzen arbeiten. Typische Anwendungsfälle für eine Graphdatenbank sind Social Networking, Empfehlungsmodule, Betrugserkennung und Wissensdiagramme.

In-Memory-Datenbanken Während andere nicht-relationale Datenbanken Daten auf Festplatten oder SSDs speichern, sind In-Memory-Datenspeicher so konzipiert, dass kein Zugriff auf Festplatten erforderlich ist. Sie eignen sich ideal für Anwendungen, die Reaktionszeiten im Mikrosekundenbereich erfordern oder große Verkehrsspitzen aufweisen. Sie können sie in Gaming- und Ad-Tech-Anwendungen für Features wie Bestenlisten, Sitzungsspeicher und Echtzeitanalysen verwenden.

Suchmaschinendatenbank Eine Suchmaschinendatenbank ist eine Art nichtrelationaler Datenbank, die sich der Suche nach Dateninhalten widmet, z. B. nach Anwendungsausgabeprotokollen, die von Entwicklern zur Problembehandlung verwendet werden. Sie verwendet Indizes, um ähnliche Merkmale unter den Daten zu kategorisieren und die Suchfunktion zu vereinfachen. Suchmaschinendatenbanken sind für die Sortierung unstrukturierter Daten wie Images und Videos optimiert.

1.3 Normalisierung

Quelle: DatabaseCamp [8]

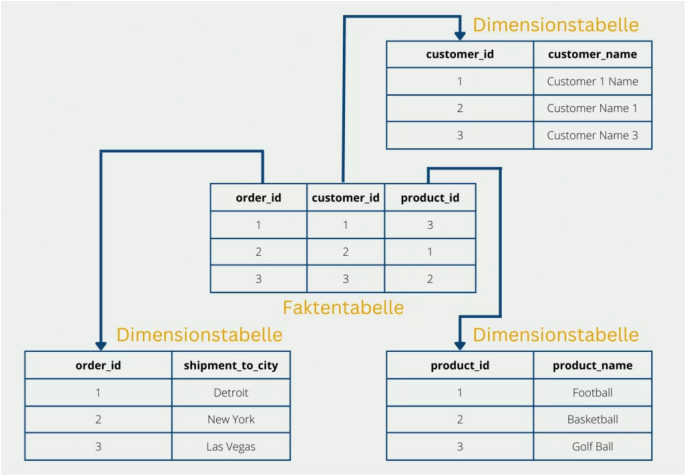
1. Normalform Eine Relation liegt in der ersten Normalform vor, wenn alle Attributwerte atomar vorliegen.

Das bedeutet, dass jedes Datenfeld lediglich einen Wert enthalten darf. Außerdem sollte sichergestellt sein, dass jede Spalte nur Werte desselben Datentyps (Numerisch, Text, etc.) enthält. Folgende Beispiele müssten entsprechend verändert werden, damit eine Datenbank in der 1. Normalform vorhanden ist:

- Adresse: “Hauptstraße 1, 12345 Berlin”
  - Straße: “Hauptstraße”
  - Hausnummer: “1”
  - PLZ: “12345”
  - Ort: “Berlin”
- Rechnungsbetrag: “128,45 €”
  - Betrag: “128,45”
  - Währung: “€”

2. Normalform Eine Relation liegt in der zweiten Normalform vor, wenn sie in der ersten Normalform vorliegt und alle Nichtschlüsselattribute voll funktional vom gesamten Primärschlüssel abhängig sind.

Der Primärschlüssel bezeichnet ein Attribut, das zur eindeutigen Identifikation einer Datenbankzeile verwendet werden kann. Dazu zählen beispielsweise die Rechnungsnummer zur Identifikation einer Rechnung oder die Ausweisnummer zur Identifikation einer Person.



Konkret bedeutet dies in der Anwendung, dass alle Merkmale ausgelagert werden müssen, die nicht ausschließlich vom Primärschlüssel abhängig sind. In der Praxis führt dies dann oft zu einem sogenannten Sternschema.

3. Normalform Eine Relation liegt in der dritten Normalform vor, wenn sie in der ersten und zweiten Normalform vorliegt und keine transitiven Abhängigkeiten bestehen.

Eine transitive Abhängigkeit liegt vor, wenn ein Attribut, welches kein Primärschlüssel ist, nicht nur von diesem abhängt, sondern auch von anderen Attributen.

Wenn wir in unserem Beispiel eine Tabelle haben, in der die Rechnungsnummer, die Produktnummer und der Preis gegeben ist, haben wir höchstwahrscheinlich eine transitive Abhängigkeit. Der Preis des Produktes hängt nämlich nicht wirklich von der Rechnungsnummer ab, sondern vielmehr von der Produktnummer, da für jedes Produkt ein fester Preis definiert ist.

Diese Abhängigkeit kann man auflösen, indem man die Produkte in eine neue Tabelle auslagert und somit das Attribut Preis aus der ursprünglichen Tabelle rausfällt.

1.3.1 Anomalien

Quelle: DatabaseCamp [9]

Anomalien in Datenbanken treten bei einer nicht existierenden oder fehlerhaften Normalisierung auf. Es existieren drei Arten von Datenbank-Anomalien, die Einfüge-Anomalie, die Änderungs-Anomalie und die Lösch-Anomalie.

In der Datenbanktentwicklung ist die Dritte Normalform oft ausreichend, um die perfekte Balance aus Redundanz, Performance und Flexibilität für eine Datenbank zu gewährleisten. Sie eliminiert auch die meisten Anomalien in einer Datenbank, aber nicht alle.

Einfügeanomalie Bei einem fehlerhaften oder inkorrekten Datenbankdesign kann es bei der Einfüge-Anomalie passieren, dass Daten gar nicht in die Datenbank übernommen werden, wenn zum Beispiel der Primärschlüssel keinen Wert erhalten hat, oder eine unvollständigen Eingabe von Daten zu Inkonsistenzen führt.

Änderungsanomalie Bei der Änderungs-Anomalie, auch Update-Anomalie genannt, werden gleiche Attribute eines Datensatzes in einer Transaktion nicht automatisch geändert. So entsteht eine Inkonsistenz der Daten.

Löschanomalie Bei einer Löschanomalie kann es passieren, dass ein Benutzer einer Datenbank aktiv Informationen löschen will und damit indirekt, aufgrund des fehlerhaften Datenbankdesigns, andere zusammenhängende Informationen parallel mitlöscht.

1.4 SQL

Alle SQL-Komponenten für Abfragen siehe *1 Datenbanken auf Seite 1*.

1.4.1 Projektion vs. Selektion

Quelle: Tino Hempel [30]

Selektion Bei der Selektion werden Zeilen aus einer Tabelle ausgewählt, die bestimmten Eigenschaften genügen.

Aus der Tabelle Schüler sollen alle Zeilen selektiert werden, in denen der Name MMüllerßteht. Die Selektion hat also die Form: $S_{Name} =_{'Mueller'} (Schueler)$

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

$S_{Name} =_{'Mueller'} (Schueler)$

SNr	Vorname	Name
12	Paul	Müller
308	Olaf	Müller

Projektion Bei der Projektion werden Spalten aus einer Tabelle ausgewählt, die bestimmten Eigenschaften genügen.

Aus der Tabelle Schüler sollen alle Spalten mit dem Attribut 'Name' projiziert werden. Die Projektion hat also die Form: $P_{Name}(Schueler)$

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

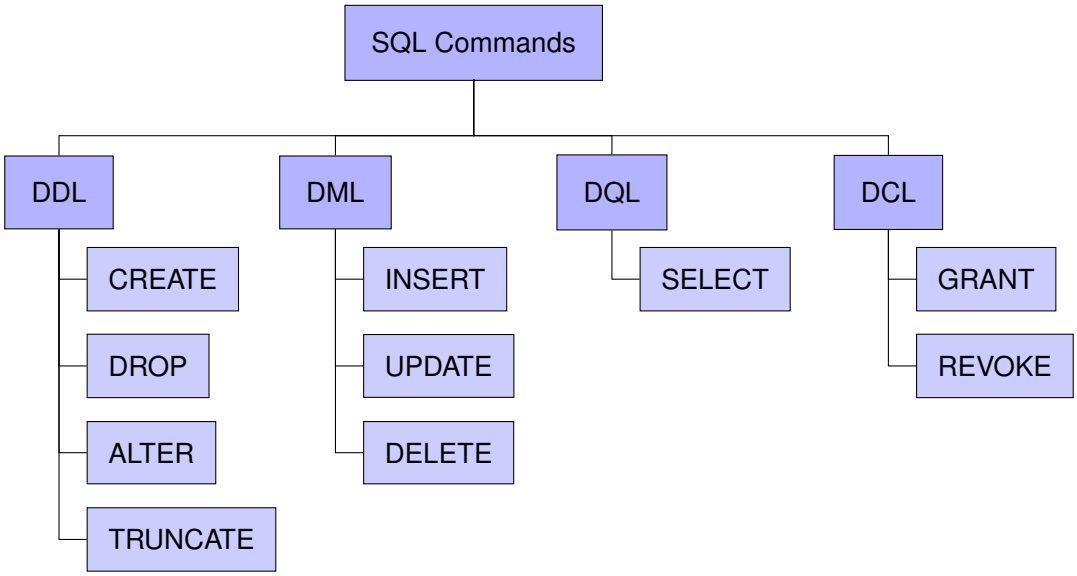
$P_{Name}(Schueler)$

Name
Müller
Schmidt
Lehmann
Müller
Paulsen

1.4.2 DDL, DML & DCL

Quelle: GeeksforGeeks [15]

1. DDL - Data Definition Language
2. DML - Data Manipulation Language
3. DQL - Data Query Language
4. DCL - Data Control Language



1.4.3 CRUD

Quelle: sqlshack.com [27]

C refers to CREATE aka add, insert. In this operation, it is expected to insert a new record using the SQL insert statement. SQL uses INSERT INTO statement to create new records within the table.

```
INSERT INTO <tablename> (column1,column2,...)
VALUES(value1,value2,...),( value1,value2,...), (value1,value2,...) ...
```

```
INSERT INTO dbo.Demo
(id, name)
VALUES
(2, 'Jayaram'),
(3, 'Pravitha');
```

R refers to SELECT (data retrieval) operation. The word 'read' retrieves data or record-set from a listed table(s). SQL uses the SELECT command to retrieve the data.

```
SELECT * FROM <TableName>;
```

U refers to Update operation. Using the Update keyword, SQL brings a change to an existing record(s) of the table. When performing an update, you'll need to define the target table and the columns that need to update along with the associated values, and you may also need to know which rows need to be updated. In general, you want to limit the number of rows in order to avoid lock escalation and concurrency issues.

```
UPDATE <TableName>
SET Column1=Value1, Column2=Value2,...
WHERE <Expression>
```

D refers to removing a record from a table. SQL uses the SQL DELETE command to delete the record(s) from the table.

```
DELETE FROM <TableName>
WHERE <Expression>
```

1.4.4 Subqueries

Fehlt

1.4.5 Aggregatfunktionen

Fehlt

1.4.6 Mengenoperationen (Schnitt-, Vereinigungs- und Differenzmenge)

Quelle: Glossar hs-augsburg [17]

Mengenoperatoren verbinden zwei Abfragen zu einem Resultat.

Beispieltabelle:

Tabelle student

matrikel_nr	name	vorlesung
911574	Meier	Java
676676	Schulz	Datenbanken

Tabelle lehrender

matrikel_nr	name	vorlesung
878999	Kowa	Datenbanken
665544	Müller	XML

UNION bildet die Vereinigung zweier Relationen indem Zeilen der ersten Menge oder des ersten Operanden mit allen Zeilen der zweiten Menge zusammengefasst werden. Zeilen, die in der Ergebnismenge zweimal vorkommen, werden zu einer einzigen Zeile zusammengefasst. Die Datentypen der Spalten müssen kompatibel sein, d.h. es muss entweder ein impliziter Cast (z.B. int auf double) möglich sein, oder wenn dies nicht möglich ist, muß ein expliziter Cast erfolgen. - dies bezieht sich auch auf die Anordnung der Spalten in der Abfrage.

```
1 SELECT name  FROM student
2 UNION
3 SELECT name FROM lehrender
```

Ergebnis:

name
Meier
Schulz
Kowa
Müller

UNION ALL vereinigt alle Zeilen der ersten Menge oder des ersten Operanden mit allen Zeilen der zweiten Menge. Im Unterschied zu UNION werden auch die Duplikate ausgegeben.

INTERSECT überprüft die Zeilen der beiden Eingangsmengen und gibt nur jene Zeilen aus, die in beiden Eingangsmengen vorkommen. Die Durchschnittsmenge wird aus den zwei Relationen gebildete. Auch hier werden vor dem Erstellen der Ergebnismenge die redundanten Zeilen ausgeschaltet.

```
1 SELECT vorlesung FROM student
2 INTERSECT
3 SELECT vorlesung FROM lehrender
```

Ergebnis:

vorlesung
Datenbanken

MINUS gibt die Zeilen aus, die in der ersten Menge, NICHT aber in der zweiten Menge enthalten sind. Zeilen, die in der ersten Menge zweimal vorkommen, werden auf Redundanz überprüft und komprimiert, bevor der Vergleich mit der zweiten Menge beginnt.

```
1 SELECT vorlesung FROM student
2 MINUS
3 SELECT vorlesung FROM lehrender
```

Ergebnis:

vorlesung
Java

1.4.7 SQL Injection

Quelle: kaspersky [22]

Eine SQL-Injection, manchmal abgekürzt als SQLi, ist eine Art von Sicherheitslücke, bei der ein Angreifer einen Teil des SQL-Codes verwendet, um eine Datenbank zu manipulieren und Zugriff auf potenziell wertvolle Informationen zu erhalten. Dies ist eine der häufigsten und bedrohlichsten Angriffsarten, da sie potenziell gegen jede Webanwendung oder Webseite eingesetzt werden kann, die eine SQL-basierte Datenbank verwendet (was bei den meisten der Fall ist).

Arten der SQL-Injection

In-band SQLi Diese Art von SQLi-Angriff ist für Angreifer sehr einfach, da sie denselben Kommunikationskanal verwenden, um Angriffe zu starten und Ergebnisse zu sammeln. Diese Art von SQLi-Angriff hat zwei Untervarianten:

- **Fehlerbasierte SQLi:** Die Datenbank erzeugt aufgrund der Aktionen des Angreifers eine Fehlermeldung. Anhand der von diesen Fehlermeldungen generierten Daten sammelt der Angreifer Informationen über die Datenbankinfrastruktur.
- **Union-basierte SQLi:** Der Angreifer verwendet den UNION-SQL-Operator, um die gewünschten Daten zu erhalten, indem er mehrere Select-Anweisungen in einer einzigen HTTP-Antwort zusammenfasst.

Inferentielle SQLi (auch bekannt als Blind SQL Injection) Bei dieser Art von SQLi nutzen Angreifer die Antwort- und Verhaltensmuster des Servers nach dem Senden von Daten-Nutzdaten, um mehr über seine Struktur zu erfahren. Die Daten werden nicht von der Datenbank der Webseite an den Angreifer übertragen, so dass der Angreifer die Informationen über den In-Band-Angriff nicht sieht (daher der Begriff 'blinde SQLi'). Inferentielle SQLi kann in zwei Untertypen unterteilt werden:

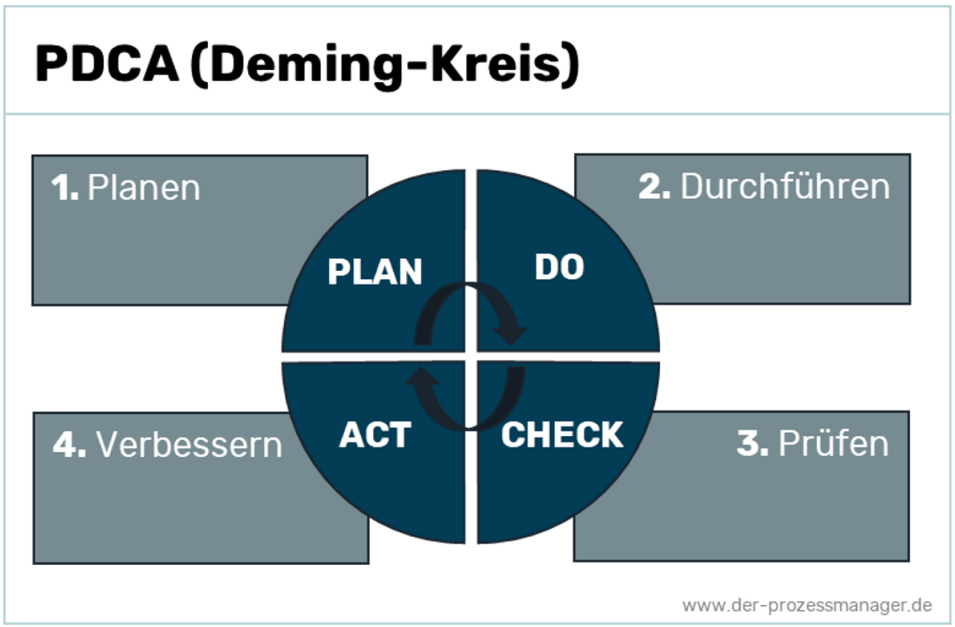
- **Zeitbasierte SQLi:** Angreifer senden eine SQL-Abfrage an die Datenbank und lassen die Datenbank einige Sekunden warten, bevor sie die Abfrage als wahr oder falsch beantwortet.
- **Boolean SQLi:** Angreifer senden eine SQL-Abfrage an die Datenbank und lassen die Anwendung daraufhin entweder ein wahres oder ein falsches Ergebnis erzeugen.

2 Qualitätssicherung

2.1 PDCA-Zyklus

Quelle: der-prozessmanager.de [11]

Der PDCA-Zyklus (auch Deming-Kreis, Deming-Zyklus oder PDCA Kreislauf) bezeichnet ein grundlegendes Konzept im kontinuierlichen Verbesserungsprozess. Es dient der Weiterentwicklung von Produkten und Dienstleistungen sowie bei der Fehler-Ursache-Analyse. Der PDCA-Kreis besteht aus den vier sich wiederholenden Phasen: Plan-Do-Check-Act (dt. Planen – Umsetzen – Überprüfen – Handeln).



Vorteile des PDCA-Kreises

Der wesentliche Vorteil der PDCA-Methode ist wohl die einfache Anwendbarkeit. Das Vorgehen hinsichtlich der spezifischen Aufgaben und Problemstellungen kann nahezu uneingeschränkt angepasst werden. Mit den Schritten Plan, Do, Check und Act bleibt dennoch ein solides Gerüst bestehen.

Weitere Vorteile:

- Benötigt wenig Anleitung auf Grund des einfachen Aufbaus
- Die kreisförmige Konzeption ermöglicht ständige Verbesserung
- Durch den iterativen Ansatz lässt der PDCA-Zyklus Kontrolle und Analyse zu

Nachteile des PDCA-Kreises

Der große Vorteil des Demingkreises ist gleichzeitig auch ein wesentlicher Nachteil: Schnelle Problemlösungen lassen sich mit Hilfe des PDCA-Zyklus nicht umsetzen.

Nachteile auf einen Blick:

- Unklare Definition der einzelnen Schritte kann zu falschem Einsatz führen
- Verbesserungen im Unternehmen müssen langfristig gedacht sein
- Eher ein reaktiver Ansatz, statt proaktiv

2.2 Incident Management

Quelle: it-processmaps.com [28]

Incident Management verwaltet alle Incidents über ihren gesamten Lebenszyklus. Das primäre Ziel dieses ITIL-Prozesses besteht darin, einen IT Service für den Anwender so schnell wie möglich wieder herzustellen.

ITIL unterscheidet zwischen Incidents (Service-Unterbrechungen) und Service Requests (d. h. Anfragen von Anwendern, die keine Service-Unterbrechungen betreffen, wie z.B. das Zurücksetzen eines Passworts). Service-Unterbrechungen werden vom Incident-Management-Prozess behandelt, während Service-Anfragen vom Request Fulfilment bearbeitet werden.

Der Incident-Management-Prozess kann auf verschiedenen Wegen angestoßen werden: Ein Anwender, Kunde oder Supplier kann eine Störung melden, technisches Personal kann einen (drohenden oder tatsächlichen) Ausfall feststellen, oder ein Incident kann automatisch von einem Event-Monitoring-System ausgelöst werden.

Alle Incidents sollten in Incident Records festgehalten werden, so dass ihr Status verfolgt und ihr vollständiger Verlauf dokumentiert werden kann. Die initiale Kategorisierung und Priorisierung der Incidents ist ein wichtiger Schritt zur Bestimmung, wie mit dem Incident verfahren wird und wieviel Zeit für dessen Lösung verfügbar ist.

Falls möglich, sollten Incidents mit anderen Incidents, Problems und Known Errors verknüpft werden.

2.3 Service Level Agreement (SLA), Servicelevel 1-3

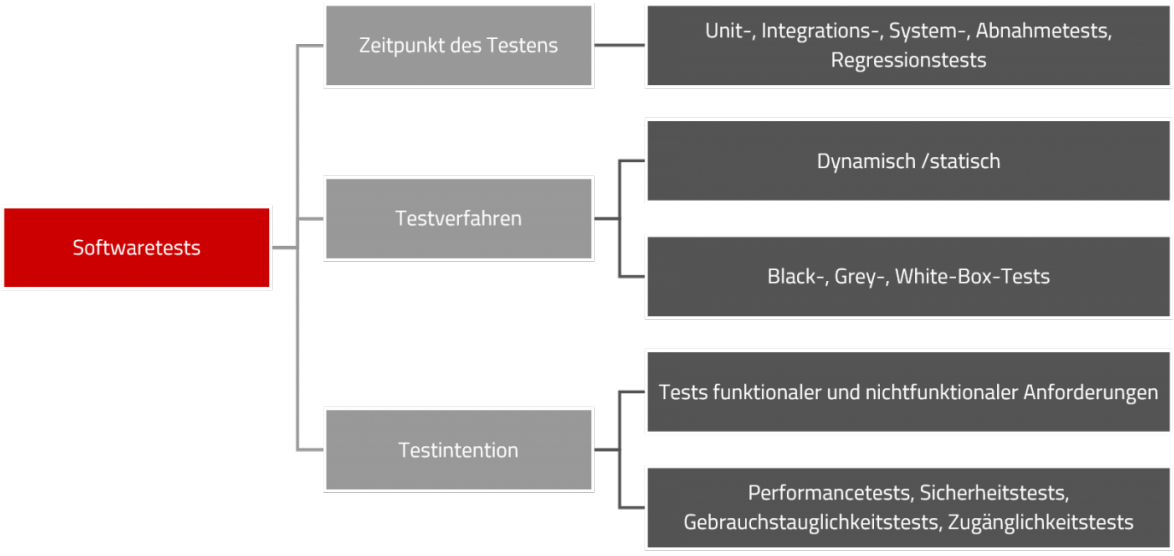
Quelle: Amazon [1]

Ein Service Level Agreement (SLA) ist ein Vertrag mit einem Outsourcing- und Technologieanbieter, in dem das Serviceniveau festgelegt ist, das ein Anbieter dem Kunden zu liefern verspricht. Sie gibt Aufschluss über Kennzahlen wie Betriebszeit, Lieferzeit, Reaktionszeit und Lösungszeit. In einem SLA ist auch festgelegt, was zu tun ist, wenn die Anforderungen nicht erfüllt werden, z. B. zusätzliche Unterstützung oder Preisnachlässe. SLAs werden in der Regel zwischen einem Kunden und einem Service-Anbieter vereinbart, obwohl auch Geschäftseinheiten innerhalb desselben Unternehmens untereinander SLAs abschließen können.

2.4 Testen

2.4.1 Klassifizierung von Testverfahren

- Wer testet?
  - Mensch (manuell) vs. Maschine (automatisch)
  - Entwickler vs. Benutzer
- Was wird getestet?
  - Komponente (Unit-Test/Funktionstest/Klassentest) vs. Integration vs. System (End-to-End)
  - Testpyramide
- Wie wird getestet?
  - Bottom-Up vs. Top-Down
  - statisch (Kompilierzeit) vs. dynamisch (Laufzeit)
  - ohne Kenntnis des Codes (Blackbox) vs. mit Kenntnis des Codes (Whitebox)
  - explorativ
  - Schreibtischtest/Review
- Wann wird getestet?
  - Vor vs. nach der Entwicklung
  - Abnahmetest
- Warum wird getestet?
  - Regressionstest
  - Lasttest/Belastungstest
  - Smoketest



Quelle: redbots [29]

2.5 Versionsverwaltung

Quelle: Atlassian [3]

git commands

git add	Verschiebt Änderungen aus dem Arbeitsverzeichnis in die Staging-Umgebung. Auf diese Weise kannst du einen Snapshot vorbereiten, bevor du an den offiziellen Verlauf committest.
git branch	Dieser Befehl ist dein Allzwecktool zur Branch-Administration. Damit kannst du isolierte Entwicklungsumgebungen innerhalb eines einzigen Repositorys erstellen.
git checkout	Neben dem Auschecken alter Commits und alter Dateiüberarbeitungen kannst du mit 'git checkout' auch zwischen bestehenden Branches navigieren. In Kombination mit den grundlegenden Git-Befehlen kann dadurch in einer bestimmten Entwicklungslinie gearbeitet werden.
git clone	Erstellt eine Kopie eines bestehenden Git-Repositorys. Klonen ist für Entwickler die gängigste Art, eine Arbeitskopie eines zentralen Repositorys zu erhalten.
git commit	Committet den Snapshot aus der Staging-Umgebung in den Projektverlauf. Zusammen mit 'git add' bildet er den grundlegenden Workflow für alle Git-Benutzer.
git fetch	Mit 'git fetch' wird ein Branch von einem anderen Repository zusammen mit allen zugehörigen Commits und Dateien heruntergeladen. Dabei wird jedoch nichts in dein lokales Repository integriert. Auf diese Weise hast du die Möglichkeit, Änderungen vor dem Merge in dein Projekt noch zu überprüfen.
git init	Initialisiert ein neues Git-Repository. Wenn du für ein Projekt eine Versionskontrolle einrichten möchtest, ist dies der erste Befehl, den du kennen musst.
git log	Damit kannst du ältere Überarbeitungen eines Projekts ansehen. Der Befehl bietet mehrere Formatierungsoptionen zur Anzeige committeter Snapshots.
git merge	Eine leistungsstarke Option zur Integration von Änderungen von voneinander abweichenden Branches. Nach dem Forken des Projektverlaufs mit 'git branch', kann diese mit 'git merge' wieder zusammengeführt werden.
git pull	Pulls sind die automatisierte Version von git fetch. Dabei wird ein Branch von einem Remote-Repository heruntergeladen und dann direkt in den aktuellen Branch gemergt. Dies ist das Git-Äquivalent von svn update.

git push	'git push' ist das Gegenteil von 'git fetch' (mit ein paar Einschränkungen). Du kannst mit diesem Befehl einen lokalen Branch in ein anderes Repository verschieben, was eine bequeme Methode zur Veröffentlichung von Beiträgen ist. Dies ist wie 'svn commit', aber hierbei wird eine Reihe von Commits statt eines einzigen Changesets gesendet.
git rebase	Mit Rebasing kannst du Branches verschieben, um unnötige Merge-Commits zu vermeiden. Der daraus resultierende lineare Verlauf ist oft leichter zu verstehen und zu durchsuchen.
git revert	Macht einen committeten Snapshot rückgängig. Wenn du einen fehlerhaften Commit entdeckst, kannst du ihn mit 'git revert' sicher und einfach von der Codebasis entfernen.
git status	Gibt den Status des Arbeitsverzeichnisses und den Status des Snapshots in der Staging-Umgebung zurück. Diesen Befehl solltest du zusammen mit 'git add' und 'git commit' ausführen, um genau zu sehen, was im nächsten Snapshot enthalten sein wird.

3 IT-Sicherheit

3.1 Datensicherheit

3.1.1 Vertraulichkeit, Integrität, Verfügbarkeit (C.I.A Prinzip)

Quelle: Enginsight [13]

Vertraulichkeit Ein System liefert Vertraulichkeit, wenn niemand unautorisiert Informationen gewinnen kann.

Integrität Ein System gewährleistet die Integrität, wenn es nicht möglich ist, zu schützende Daten unautorisiert und unbemerkt zu verändern.

Verfügbarkeit Ein System gewährt Verfügbarkeit, wenn authentifizierte und autorisierte Subjekte in der Wahrnehmung ihrer Berechtigungen nicht unautorisiert beeinträchtigt werden können.

3.1.2 USV (Unterbrechungsfreie Stromversorgung)

Quelle: hagel-it [16]

Was ist eine USV?

Viele Geräte, wie Server und Router müssen hoch verfügbar sein, um z.B. das Internet am Laufen zu halten. Nicht nur Hard- und Software Probleme, sondern auch die Stromversorgung bildet eine Sicherheitslücke. Daher werden meist sensible IT-Systeme mit einer USV (unterbrechungsfreie Stromversorgung) ausgestattet. Denn im Falle eines Netzausfalls möchte vermieden werden, dass beispielsweise dem Server plötzlich der Strom fehlt und ausgeschaltet wird. Viel mehr möchte man erreichen, dass der Server in diesem Fall sauber und ordentlich herunterfährt.

3.1.3 Firewall

Quelle: cisco [7]

Eine Firewall ist eine Netzwerksicherheitsvorrichtung, die eingehenden und ausgehenden Netzwerkverkehr überwacht und auf Grundlage einer Reihe von definierten Sicherheitsregeln entscheidet, ob bestimmter Datenverkehr zugelassen oder blockiert wird.

Firewalls bilden bereits seit über 25 Jahren die erste Verteidigungslinie beim Schutz von Netzwerken. Sie fungieren als Barriere zwischen geschützten und kontrollierten Bereichen des internen, vertrauenswürdigen Netzwerks und nicht vertrauenswürdigen, äußeren Netzwerken wie dem Internet.

Eine Firewall kann Hardware, Software, Software-as-a-Service (SaaS), eine Public Cloud oder eine Private Cloud (virtuell) sein.

3.1.4 Schutzbedarfskategorien

Quelle: BSI [5]

Beispiel Für das Beispielunternehmen, die RECPLAST GmbH, wurde bezüglich der Schadensszenarien „finanzielle Auswirkungen“ und „Beeinträchtigung der Aufgabenerfüllung“ folgendes festgelegt:

Normaler Schutzbedarf:

- „Der mögliche finanzielle Schaden ist kleiner als 50.000 Euro.“
- „Die Abläufe bei RECPLAST werden allenfalls unerheblich beeinträchtigt. Ausfallzeiten von mehr als 24 Stunden können hingenommen werden.“

Hoher Schutzbedarf:

- „Der mögliche finanzielle Schaden liegt zwischen 50.000 und 500.000 Euro.“
- „Die Abläufe bei RECPLAST werden erheblich beeinträchtigt. Ausfallzeiten dürfen maximal 24 Stunden betragen.“

Sehr hoher Schutzbedarf:

- „Der mögliche finanzielle Schaden liegt über 500.000 Euro.“
- „Die Abläufe bei RECPLAST werden so stark beeinträchtigt, dass Ausfallzeiten, die über zwei Stunden hinausgehen, nicht toleriert werden können.“

3.1.5 Begriffe zu Hacking

Quelle: Buch zu Hacking von mitp [14]

3.1.6 Hacker

Scriptkiddies Sie haben wenig Grundwissen und versuchen, mithilfe von Tools in fremde Systeme einzudringen. Dabei sind diese Tools meist sehr einfach über eine Oberfläche zu bedienen. Die Motivation ist meistens Spaß und die Absichten sind oft krimineller Natur. Oftmals möchten Scriptkiddies mit ihren Aktionen Unruhe stiften. Die Angriffe sind meist ohne System und Strategie. Viele Hacker starten ihre Karriere als Scriptkiddie, nutzen die Tools zunächst mit wenig Erfahrung, lernen aus dem Probieren, entwickeln sich weiter und finden dadurch einen Einstieg in die Szene.

Black Hats Diese Gattung Hacker beschreibt am ehesten die Hacker, die man aus den Medien kennt. Hier redet man von Hackern mit bösen Absichten. Sie haben sehr gute Kenntnisse und greifen bewusst und strukturiert Unternehmen, Organisationen oder Einzelpersonen an, um diesen Schaden zuzufügen. Die Ziele der Black Hats sind vielfältig und reichen vom einfachen Zerstören von Daten bis hin zum Diebstahl von wertvollen Informationen, wie Kontodaten oder Unternehmensgeheimnissen. In manchen Fälle reicht es den Black Hats auch, wenn sie erfolgreich die Server ihres Opfers lahmlegen und damit Sabotage verüben.

White Hats Einen *White Hat Hacker* nennt man oft auch einen *Ethical Hacker*. Er nutzt das Wissen und die Tools eines Hackers, um zu verstehen, wie Black Hats bei ihren Angriffen vorgehen. Im Gegensatz zum Black Hat will der White Hat jedoch die betreffenden Systeme letztlich vor Angriffen besser schützen und testet daher die Schwachstellen aktiv aus. Damit hat ein White Hat Hacker grundsätzlich keine bösen Absichten, im Gegenteil, er unterstützt die Security-Verantwortlichen der jeweiligen Organisation. White Hat Hacker oder Ethical Hacker versuchen im Anschluss an ihre Hacking-Tätigkeit herauszufinden, welche Sicherheitslücken es gibt und geben eine Anleitung dazu, diese möglichst effizient zu schließen.

3.1.7 Phishing

Beim Phishing versucht der Phisher in der Regel, an sensible bzw. persönliche Daten des Opfers zu gelangen. Dies sind meistens entweder Login-Informationen oder Kreditkartendaten - im einfachsten Fall einfach eine valide E-Mail-Adresse. Dies wird durch gefakte E-Mails, Webseiten oder Kurznachrichten erreicht.

3.1.8 Spoofing

Quelle: Kaspersky [21]

Spoofing ist ein weit gefasster Begriff, der Aktivitäten beschreibt, bei denen Cyberkriminelle sich als vertrauenswürdige Benutzer oder Geräte „tarnen“, um Benutzer zu Handlungen zu bewegen, die dem Hacker zugutekommen und dem Benutzer Schaden zufügen.

Mail-Spoofing Mail-Spoofing zählt zu den am weitesten verbreiteten Angriffen und zielt oftmals darauf ab, dass persönliche Daten preisgegeben oder Finanztransaktionen durchgeführt werden. Die E-Mails, mit denen diese Angriffe gestartet werden, stammen scheinbar von vertrauenswürdigen Absendern wie Kunden, Kollegen oder Vorgesetzten. In Wirklichkeit kommen sie aber von Cyberkriminellen, die sich bewusst tarnen, um Ihr Vertrauen und Ihre Unterstützung zu erlangen und Sie so zu bestimmten Handlungen zu bewegen. Dabei kann es beispielsweise um eine Geldüberweisung oder die Erlaubnis für den Zugriff auf ein System gehen.

IP-Spoofing Während sich Betrüger beim Mail-Spoofing auf einzelne Benutzer konzentrieren, ist IP-Spoofing in erster Linie auf Netzwerke ausgerichtet. Beim IP-Spoofing versucht ein Angreifer, durch das Senden von Nachrichten von einer gefälschten oder „gespooften“ IP-Adresse unbefugten Zugang zu einem System zu erlangen. Es soll dabei der Anschein erweckt werden, die Nachricht stamme aus einer vertrauenswürdigen Quelle, wie zum Beispiel von einer Adresse aus dem eigenen internen Computernetzwerk.

3.1.9 Session Hijacking

Mit »Session Hijacking« ist die Übernahme einer Session zwischen zwei Systemen durch einen Angreifer gemeint. Session Hijacking ist eine perfide Angriffsform, da sich der Angreifer sozusagen »ins gemachte Nest« setzt. Dabei können diverse Schwachstellen auf unterschiedlichen Ebenen der Kommunikation ausgenutzt werden.

3.1.10 Man-in-the-Middle (MITM)

Bei einer MITM-Attacke platziert sich der Angreifer so, dass er die relevante Kommunikation zwischen Person 1 und Person 2 über ein vomn ihm kontrolliertes System leitet und somit den gesamten Traffic mitlesen und ggf. sogar manipulieren kann. Dies geschieht für Person 1 und Person 2 komplett transparent. Das bedeutet, dass Person 1 glaubt, mit Person 2 direkt zu kommunizieren, und umgekehrt. Tatsächlich aber gibt sich der Angreifer gegenüber Person 1 als Person 2 aus und gegenüber Person 2 tut er so, als wäre er Person 1.

3.1.11 SQL-Injection

1.4.7 SQL Injection auf Seite 13

3.1.12 DoS, DDoS

DoS *Denial-of-Service*

DDoS *Distributed-Denial-of-Service*

Unter einem *Denial-of-Service-Angriff* (DoS) verstehen wir jede Art von Angriff, die das Ziel verfolgt, die Verfügbarkeit eines Computersystems oder Netzwerkdienstes zu reduzieren oder komplett zu verhindern. Diese Nichtverfügbarkeit kann auf ganz verschiedenen Wegen erreicht werden. Letztlich geht es aber immer darum, dass eine Komponente ihren Dienst nicht so verrichten kann wie vorgesehen, da deren Funktionalität willentlich und vorsätzlich angegriffen wurde.

Dabei wird das Opfer-System in der Regel mit Dienstanfragen oder Datenpaketen überhäuft, um dessen Ressourcen aufzubrauchen und eine Überlastung zu erreichen. Das Ziel eines DoS-Angriffs ist rein destruktiv.

In den meisten Fällen basieren DoS-Angriffe schlicht auf einer massiven Überlastung der Opfer-Systeme. Nach dem Motto »Viel hilft viel« werden heutzutage häufig konzentrierte Angriffe von Hunderten oder Tausenden Systemen aus dem Internet durchgeführt. Dies nennen wir *Distributed-Denial-of-Service* (DDoS-Attacke).

3.1.13 Viren

Der oder das Computervirus (beides ist mittlerweile statthaft) ist klassischerweise ein Programm, das sich selbst verbreitet, indem es sich in ein anderes programm, den Wirt, einschleust. Im Gegensatz zu Trojanern oder Würmern benötigen Viren also einen Wirt und können sich nicht selbstständig fortpflanzen.

In dem Moment, in dem das Wirtsprogramm ausgeführt wird, kann auch der Virus aktiv werden und seine Payload ausführen. Zur Replikation führt der Virus immer einen Prozess aus, der es ihm ermöglicht, sich an einen neuen Wirt zu hängen, um sich weiterzuverbreiten.

3.1.14 Würmer

Würmer können sich - im Gegensatz zu Viren - selbstständig und ohne Wirt vermehren. Sie verbreiten sich über Netzwerke oder Wechselmedien, wie USB-Sticks. Dadurch können sie sich höchst effektiv replizieren und verbreiten.

Ebenso wie Trojaner und Viren können auch Würmer einen Schadcode ausführen.

3.1.15 Trojaner

Die Trojaner werden als harmloses Programm getrarnt, enthalten aber Schadcode, der im Hintergrund ohne Wissen des Anwenders ausgeführt wird. Trojaner sind die flexibelste und die am weitesten verbreitete Form von Malware. Im Grunde genommen ist ein Trojaner nur eine Tarnung für eine beliebige weitere Malware-Funktion wie z.B. Installation von:

- Backdoors
- Keylogger
- Ransomware
- Sniffer
- Bots für ein Botnet
- und so weiter

3.1.16 Sniffer, Spyware und Keylogger

Zahlreiche Schadprogramme versuchen, den Anwender auszuspionieren, und senden die Daten anschließend an vordefinierte Zielsysteme im Internet. Die Spionagetätigkeiten können diverse Aspekte und Komponenten umfassen. Angefangen von Daten wie Browser-Verlauf und anderen Zugriffsverläufen (*Most Recently Used*, MRU) über das Mitschneiden des Netzwerk-Traffics oder Aktivieren von Mikrofon und Webcam bis hin zum Protokollieren jedes einzelnen Tastenanschlags mittels Keylogger kann ein Angreifer umfassende Daten über das Opfer sammeln.

3.1.17 Botnetze

Ein *Botnet* besteht aus zahlreichen *Bots*, also einem Stückchen Software, das sich im Opfer-System eingenistet hat und bereit ist, vom Command & Control-Server aus dem Internet Befehle zu empfangen. Bots könenn diverse Funktionen erfüllen - angefangen vom Spamversand über Spyware-Funktionen bis hin zu gezielten und konzentrierten *Distributed-Denial-of-Service-Angriffen*.

3.1.18 Scareware

Eine perfide Art, Malware auf ein Opfer-System zu bringen, ist die *Scareware*. Hierbei täuscht ein Popup einer Website oder eine als kostenloses Antiviren-Programm verteilte Software dem Benutzer den Fund zahlreicher, gefährlicher Vireninfektionen vor, die über ein zu installierendes, ggf. kostenpflichtiges Programm beseitigt werden können.

3.1.19 Kryptotrojaner & Ransomware

Kryptografie soll die Sicherheit erhöhen. Dank ausgeklügelter, komplexer mathematischer Verfahren und Algorithmen gelingt dies in der Regel sehr effektiv und für die »bösen Jungs« ist es teilweise sehr schwierig bis unmöglich, kryptografisch gesicherte Daten und Datenströme zu knacken bzw. zu entschlüsseln. Umso perfider wird es, wenn diese Technologien gegen uns verwendet werden. So geschehen bei den sogenannten *Kryptotrojanern* bzw. der gefürchteten *Ransomware*.

Hinter Kryptotrojanern steckt sehr viel kriminelle Energie, denn die Angreifer wollen in fast allen Fällen Geld ergaunern. Die Schädlinge verschlüsseln persönliche Daten auf dem System des Opfers so, dass sie für den Eigentümer unbrauchbar sind. Es gibt auch Varianten, die den Zugriff auf das komplette System blockieren. Erst gegen Bezahlung eines Lösegeldes (engl. *Ransom*) werden die Dateien entschlüsselt und damit wieder verfügbar gemacht. Nachdem das Opfer über eine anonyme Zahlungsmethode, wie zum Beispiel die Kryptowährung *Bitcoin*, das Lösegeld bezahlt hat, ist natürlich nicht gewährleistet, dass die Daten tatsächlich wieder freigegeben werden!

3.1.20 Backdoor

Eine »Backdoor«, oder deutsch: »Hintertür«, bezeichnet ganz allgemein einen Zugang zu einem System unter Umgehung der Zugriffsschutzmaßnahmen. Der Zweck einer Backdoor besteht in der Regel darin, einem nicht authentisierten und autorisierten Benutzer einen jederzeit verfügbaren Zugriff zum Zielsystem bereitzustellen. Dies impliziert, dass dieser Zugang den autorisierten Benutzern meistens nicht bekannt ist bzw. sein soll.

3.1.21 Exploit

Ein *Exploit* ist ein Prozess, mit dem Sie eine Schwachstelle ausnutzen können. Doch was bedeutet »ausnutzen« eigentlich? Das hängt ganz von der Schwachstelle und vom Exploit ab. In unserem Fall geht es darum, eine *Remote-Shell* bereitzustellen - egal ob Bind- oder Reverse-Shell. Diese Shell ist die *Payload* des Exploits. Durch die Schwachstelle wird z.B. die Ausführung beliebiger Kommandos möglich. Der Exploit setzt dies technisch um. Die Payload stellt dann genau diese Befehle dar, mit denen der Angreifer Zugriff auf das System erlangt.

3.1.22 Rootkit

Eine sehr effektive Möglichkeit, Malware und unerwünschte Prozesse zu verstecken und zu tarnen, besteht darin, sie für den Anwender und das System unsichtbar zu machen. Der Begriff »Rootkit« setzt sich aus dem Linux/UNIX-Administator *root* und dem Wort »Kit« zusammen und bedeutet wörtlich übersetzt ungefähr »Administrator-Werkzeugkasten«. Damit wird zum Ausdruck gebracht, dass es sich eigentlich um eine Toolsammlung handelt.

3.1.23 Verbreitung von Viren/Würmern/Trojanern

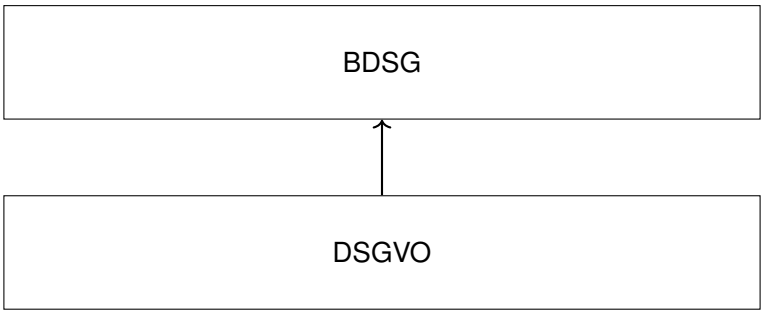
- *Websites:* Eines der wichtigsten Einfallstore für Malware sind kompromittierte Websites. Besucht das Opfer die Website, gibt es verschiedene Möglichkeiten, die der Angreifer ausnutzen kann.
  - Via *Drive-by-Downloads* wird Software ohne Wissen und Zutun des Benutzers automatisch heruntergeladen und installiert. Hierzu werden Browser-Schwachstellen ausgenutzt, da regulär HTML-Code oder Browser-Skriptsprachen kein Zugriff auf Bereiche außerhalb der Browser-Umgebung gestattet wird. Die Voraussetzung hierfür ist also ein ungepatchter Browser oder eine Zero-Day-Schwachstelle.
  - Noch direkter ist die Bereitstellung von Trojanern im Rahmen der Software-Angebote der Website. So gelang es Angreifern zum Beispiel, die beliebte und äußerst nützliche Software *CCleaner* in der Version 5.33 mit Malware zu verseuchen, sodass aus dem Programm ein Trojaner wurde.

- *Instant Messenger:* Fast jeder Messenger bietet die Möglichkeit der Dateiübertragung. Wird das Opfer dazu verführt, ein Programm entgegenzunehmen und zu installieren, hat der Angreifer sein Ziel erreicht.
- *Wechselmedien (USB-Sticks):* Ein Klassiker ist auch die Verteilung von Malware über USB-Sticks, die das Opfer anschließt. Mittlerweile ist diese Gefahr etwas entschäft, da die Autorun-Funktion bei neueren Windows-Versionen standardmäßig deaktiviert ist. Dennoch bleibt das Risiko, dass der Benutzer eine spannend klingende Programmdatei ausführt und sich damit Malware auf den Computer lädt.
- *E-Mail-Anhänge:* Ebenfalls eine der klassischen Methoden, um Malware zu verbreiten. Insbesondere Makroviren über Word- und Excel-Dateien lassen sich über diesen Weg hervorragend verteilen. Auch diverse weitere Dateitypen sind gefährdet und möglicherweise infizierbar. Seit einiger Zeit werden viele manipulierte PDF-Dateien mit angeblichen Lebensläufen verschickt. Die E-Mails sind als Blindbewerbung getarnt.
- *Links in E-Mails und Websites:* Manchmal ist es nicht notwendig, die Malware direkt zu übergeben. Auch Hyperlinks dienen dazu, den ahnungslosen Benutzer auf kompromittierte Websites zu leiten oder direkt einen Download zu starten.
- *File Sharing:* Ein beliebter Weg zur Verteilung von Malware besteht im Austausch von Dateien und Programmen via File Sharing. Fake-Programme und diverse andere trojanerbasierende Tools mit vielversprechenden Inhalten und Funktionen verleiten zum Download. Das betrifft z.B. Spiele, Bildschirmschoner, »Crackz und Warez« und so weiter.
- *Windows-Freigaben:* Die Windows-Netzwerkumgebung bietet einfachen Zugriff auf freigegebene Ordner anderer Windows-Computer im Netzwerk. Gelingt es dem Angreifer, hier zentrale Programmdateien durch infizierte Varianten auszutauschen, kann er unter Umständen mit einem Zug diverse Systeme infizieren, wenn das betreffende Programm regelmäßig von Benutzern auf dem lokalen Computer ausgeführt wird.
- *Software-Bugs:* Enthält eine Software Schwachstellen - allen voran Browser und E-Mail-Clients -, kann die Malware auch über diesen Weg auf den Computer des Opfers gelangen. Die oben erwähnten Drive-by-Downloads basieren auf Schwachstellen im Browser, wie bereits geschrieben. Aber auch andere programme, wie z.B. Flash-Player oder PDF-Reader, können für derartige Angriffe anfällig sein.
- *Fehlkonfiguration:* Ein sehr wichtiger Punkt ist die Konfiguration einer Software. Unter bestimmten Bedingungen ist eine Software entweder per Default oder durch den Administrator unsicher konfiguriert und damit anfällig für entsprechende Angriffe, die die Installation von Malware nach sich ziehen.

4 Datenschutz

DSGVO Datenschutzgrundverordnung

BDSG Bundesdatenschutzgesetz



Quelle: b-quadrat [4]

Ein entscheidender Unterschied zwischen DSGVO und BDSG liegt in ihrem Geltungsbereich. Die DSGVO ist eine Verordnung der Europäischen Union und gilt daher unmittelbar in allen Mitgliedsstaaten. Sie betrifft Unternehmen, die personenbezogene Daten von EU-Bürgern verarbeiten, unabhängig von ihrem Sitz. Dadurch wird ein einheitlicher Datenschutzstandard in der gesamten EU gewährleistet.

Das BDSG, dagegen, ist spezifisch auf Deutschland ausgerichtet. Es ergänzt die DSGVO um nationale Bestimmungen und legt besondere Anforderungen an die Datenverarbeitung in Deutschland fest. Es berücksichtigt dabei kulturelle und rechtliche Besonderheiten, die in Deutschland gelten.

4.1 Grundsätze des Datenschutzes

Quelle: IHK Rhein-Neckar [18]

Rechtmäßigkeit Die Verarbeitung personenbezogener Daten ist rechtmäßig, wenn eine Einwilligung des Betroffenen (Art. 6 Abs. 1 S. 1 lit. a) DSGVO) oder eine gesetzliche Erlaubnis (Art. 6 Abs. 1 S. 1 lit. b) bis f) DSGVO) vorliegt.

Transparenz Es muss für den Betroffenen immer ersichtlich sein, für welchen Zweck seine personenbezogenen Daten verarbeitet werden. Eine „heimliche“ Verarbeitung ist unzulässig.

Zweckbindung Bei der Verarbeitung personenbezogener Daten muss der Zweck der Verarbeitung vor der Erhebung der Daten festgelegt werden. Stellen Sie sich also immer die Frage, wofür konkret Sie die von Ihnen erhobenen Daten verarbeiten wollen. Eine nachträgliche Änderung des Zwecks ist grundsätzlich nicht zulässig. Sie müssen also stets gewährleisten, dass keine zweckentfremdete Verarbeitung der von Ihnen erhobenen Daten stattfindet.

Ferner ist der Zweck maßgebend für die Speicherdauer. Wenn er entfällt, sind Sie verpflichtet, die personenbezogenen Daten zu löschen (Ausnahme: gesetzliche Aufbewahrungspflichten, siehe unter Speicherbegrenzung).

Datenminimierung/-sparsamkeit Bei der Datenverarbeitung dürfen nur so viele personenbezogene Daten gesammelt werden, wie für den jeweiligen Verarbeitungszweck unbedingt notwendig sind. Es gilt der Grundsatz: „So viele Daten wie nötig, so wenige Daten wie möglich." Dadurch soll der Betroffene vor einer übermäßigen Preisgabe personenbezogener Daten geschützt werden.

Richtigkeit Personenbezogene Daten müssen richtig und aktuell sein sowie aus zuverlässigen Quellen stammen. Unrichtige oder veraltete Daten müssen unmittelbar gelöscht oder korrigiert werden.

Speicherbegrenzung (Löschung/Sperrung) Werden personenbezogene Daten nicht mehr benötigt, müssen sie gelöscht werden, es sei denn, der Löschung stehen gesetzliche Aufbewahrungspflichten (insbesondere im Handels- und Steuerrecht) entgegen. Solange die Aufbewahrungsfrist läuft, werden die Daten zwar nicht gelöscht, aber für eine weitere Nutzung durch den Verantwortlichen gesperrt.

Integrität und Vertraulichkeit Personenbezogene Daten müssen sicher und vertraulich behandelt werden. Insbesondere dürfen Unbefugte keinen Zugang zu ihnen haben und weder die Daten noch die Geräte, mit denen diese verarbeitet werden, benutzen können.

Rechenschaftspflicht (Dokumentation) Ihr Unternehmen muss gegenüber Aufsichtsbehörden nachweisen können, alle Vorgaben der DSGVO einzuhalten. Aus diesem Grund müssen Sie die von Ihnen getroffenen rechtlichen, technischen und organisatorische Maßnahmen zur Sicherstellung des Datenschutzes genauestens dokumentieren. Dokumentation heißt, dass Sie entsprechende Dokumente, Belege und sonstige Materialien in schriftlicher oder elektronischer Form systematisch aufbewahren und archivieren, damit Sie diese im Ernstfall unverzüglich griffbereit haben. Zu diesen Dokumentationspflichten gehört beispielsweise auch das Führen eines Verarbeitungsverzeichnisses gemäß Art. 30 DSGVO.

4.2 Betroffenenrechte

Quelle: BfDI [6]

Das Recht auf Auskunft (Art. 15 DSGVO) Mit dem Auskunftsrecht garantiert Ihnen Art. 15 der Datenschutz-Grundverordnung (DSGVO) ein bedeutsames Betroffenenrecht. Danach können Sie als betroffene Person von dem für die Datenverarbeitung Verantwortlichen Auskunft darüber verlangen, welche Daten dort über Sie gespeichert sind bzw. verarbeitet werden.

Das Recht auf Berichtigung (Art. 16 DSGVO) Die DSGVO gewährt allen Bürgern, deren personenbezogene Daten verarbeitet werden, eine Vielzahl an Rechten, wie das Recht auf Berichtigung.

Das Recht auf Löschung / 'Recht auf Vergessenwerden' (Art. 17 DSGVO) Nach der DSGVO haben Sie das Recht auf Löschung Ihrer personenbezogenen Daten sowie auf 'Vergessenwerden'.

Das Recht auf Einschränkung der Verarbeitung (Art. 18 DSGVO) Ein weiteres Betroffenenrecht, das Ihnen die DSGVO an die Hand gibt, ist das Recht auf Einschränkung der Verarbeitung.

Das Recht auf Widerspruch (Art. 21 DSGVO) Art. 21 Abs. 1 Datenschutz-Grundverordnung (DSGVO) gewährt Ihnen das Recht, aus Gründen, die sich aus Ihrer besonderen Situation ergeben, ausnahmsweise auch gegen eine an sich rechtmäßige Datenverarbeitung Widerspruch einzulegen. Einer an sich rechtmäßigen Datenverarbeitung zur Direktwerbung können Sie nach Art. 21 Abs. 2 DSGVO sogar ohne jede Begründung widersprechen.

4.3 Persönlichkeitsrechte

Quelle: dr-datenschutz [12]

4.3.1 Selbstbestimmung

Selbstbestimmung umfasst beispielsweise das Recht

- zur eigenen Namenswahl,
- auf informationelle Selbstbestimmung,
- auf Kenntnis der eigenen Abstammung,
- auf sexuelle Selbstbestimmung und
- das Grundrecht auf Gewährleistung der Vertraulichkeit und Integrität informationstechnischer Systeme.

4.3.2 Selbstbewahrung

Art. 13 GG, der die räumliche Privatheit aufrechterhalten möchte, wird durch das Recht auf Selbstbewahrung ergänzt, das den privaten Lebensbereich schützt. Hier besteht ein Recht darauf, sich von anderen abschirmen und zurückziehen zu können. Geschützt werden zum Beispiel Tagebuchaufzeichnungen, Krankenakten und ähnlich sensible Bereiche.

4.3.3 Selbstdarstellung

Selbstdarstellung bedeutet, dass jeder selbst darüber bestimmen kann, wie er sich in der Öffentlichkeit darstellt. Geradezu als legendär zu bezeichnen sind die Caroline von Monaco – Fälle. Caroline Prinzessin von Hannover ging mehrmals gerichtlich gegen in Zeitschriften veröffentlichte Fotos ihres Privatlebens vor – mit gemischtem Erfolg.

Die Selbstdarstellung als Teil des APR betrifft unter anderem das Recht

- am eigenen Bild,
- am eigenen Wort,
- auf Schutz der Vertraulichkeit des Gesprächs und
- auf Schutz der persönlichen Ehre.

Es besteht aber natürlich kein Recht darauf, nur so dargestellt zu werden, wie man das selbst gerne möchte. Wäre auch zu schön gewesen!

5 Netzwerktechnik

5.1 ISO/OSI Modell

ISO/OSI Modell Das ISO/OSI-Modell (International Organization for Standardization/Open Systems Interconnection Model) ist ein Referenzmodell, das die Grundlagen für die Kommunikation in Rechnernetzen definiert. Es unterteilt den Kommunikationsprozess in sieben Schichten, wobei jede Schicht bestimmte Funktionen und Dienste für die Datenübertragung bereitstellt. Diese Schichten sind:

1. **Physikalische Schicht (Physical Layer):** Diese Schicht ist für die physische Übertragung von Datenbits über das Kommunikationsmedium verantwortlich. Sie beschäftigt sich mit Aspekten wie Signalisierung, Übertragungsraten und elektrischen Eigenschaften der Verbindungen.
   Protokolle: Token Ring
2. **Sicherungsschicht (Data Link Layer):** Die Sicherungsschicht ist für die fehlerfreie Übertragung von Datenrahmen zwischen benachbarten Netzwerkknoten verantwortlich. Sie befasst sich mit Themen wie Rahmenbildung, Adressierung, Fehlererkennung und -korrektur sowie Zugriffskontrolle.
   Protokolle: MAC, Ethernet, Token Ring
3. **Netzwerkschicht (Network Layer):** Diese Schicht ist für die Weiterleitung von Datenpaketen von einem Quell- zum Zielknoten über ein Netzwerk verantwortlich. Sie bietet Funktionen wie Routing, Adressierung und Vermittlung von Daten zwischen verschiedenen Subnetzen.
   Protokolle: IP, IPsec, ICMP

4. **Transportschicht (Transport Layer):** Die Transportschicht ist für die zuverlässige Übertragung von Daten zwischen Endpunkten (z.B. Hosts) im Netzwerk verantwortlich. Sie stellt sicher, dass Datenpakete korrekt und in der richtigen Reihenfolge übertragen werden und bietet Mechanismen zur Fehlerbehebung und Flusskontrolle.
   Protokolle: TCP, UDP

5. **Sitzungsschicht (Session Layer):** Die Sitzungsschicht ist für die Verwaltung von Sitzungen oder Verbindungen zwischen Anwendungen auf verschiedenen Netzwerkknoten verantwortlich. Sie ermöglicht die Einrichtung, Aufrechterhaltung und Beendigung von Kommunikationssitzungen sowie die Synchronisierung und Wiederherstellung von Datenübertragungen.
   Protokolle: DHCP, DNS, FTP, HTTP, HTTPS, LDAP, SMTP

6. **Darstellungsschicht (Presentation Layer):** Diese Schicht ist für die Umwandlung von Daten in ein für die Übertragung geeignetes Format verantwortlich und umgekehrt. Sie kümmert sich um Aspekte wie Datenkompression, Verschlüsselung, Formatierung und Kodierung, um sicherzustellen, dass Daten zwischen verschiedenen Systemen verstanden werden können.
   Protokolle: DHCP, DNS, FTP, HTTP, HTTPS, LDAP, SMTP

7. **Anwendungsschicht (Application Layer):** Die Anwendungsschicht ist die oberste Schicht des ISO/OSI-Modells und stellt Dienste und Schnittstellen bereit, die Anwendungen den Zugriff auf das Netzwerk ermöglichen. Sie umfasst Protokolle und Dienste für Anwendungen wie E-Mail, Dateiübertragung, Remotezugriff und Webbrowser.
   Protokolle: DHCP, DNS, FTP, HTTP, HTTPS, LDAP, SMTP

5.2 Addressierung

IPv4/IPv6 IPv4 (Internet Protocol Version 4) und IPv6 (Internet Protocol Version 6) sind Protokolle, die zur Adressierung von Geräten in einem Netzwerk verwendet werden. IPv4 verwendet 32-Bit-Adressen und ist das am häufigsten verwendete Protokoll im Internet. Es hat jedoch eine begrenzte Anzahl von verfügbaren Adressen, was mit dem Wachstum des Internets zu Engpässen geführt hat. IPv6 hingegen verwendet 128-Bit-Adressen und wurde entwickelt, um dieses Problem zu lösen und zukünftiges Wachstum zu ermöglichen.

MAC MAC (Media Access Control) bezieht sich auf die physische Adresse eines Netzwerkgeräts, die auch als Hardwareadresse bekannt ist. Diese Adresse wird in der Netzwerkkarte eines Geräts eingebettet und dient zur eindeutigen Identifizierung innerhalb eines lokalen Netzwerks. MAC-Adressen sind in der Regel in Form von sechs Doppelpunkten getrennten Hexadezimalzahlen dargestellt und werden von Netzwerkprotokollen wie Ethernet verwendet.

ARP ARP (Address Resolution Protocol) ist ein Netzwerkprotokoll, das verwendet wird, um die IP-Adresse eines Netzwerkgeräts in die entsprechende MAC-Adresse umzuwandeln. Wenn ein Gerät in einem lokalen Netzwerk Daten an ein anderes Gerät senden möchte, benötigt es die MAC-Adresse des Zielgeräts. ARP ermöglicht es, diese Zuordnung von IP-Adressen zu MAC-Adressen dynamisch zu ermitteln und zu speichern, indem es ARP-Anfragen sendet und ARP-Antworten empfängt. So kann effizienter Datenverkehr im Netzwerk stattfinden.

5.3 Routing, Switching

Routing Routing bezieht sich auf den Prozess der Weiterleitung von Datenpaketen zwischen verschiedenen Netzwerken oder Subnetzen, um den besten Weg für den Datenverkehr zu finden. Dabei werden Routing-Algorithmen verwendet, die basierend auf verschiedenen Kriterien wie Kosten, Latenzzeit oder Bandbreite entscheiden, welcher Pfad für die Übertragung von Daten am besten geeignet ist. Router sind Geräte, die für das Routing zuständig sind und Datenpakete entsprechend weiterleiten.

Switching Switching ist der Prozess des Weiterleitens von Datenpaketen innerhalb eines lokalen Netzwerks. Im Gegensatz zu Routern, die Daten zwischen verschiedenen Netzwerken weiterleiten, arbeiten Switches auf der Ebene des lokalen Netzwerks und verbinden verschiedene Geräte innerhalb desselben Netzwerks miteinander. Switches verwenden MAC-Adressen, um Datenpakete an das richtige Zielgerät innerhalb des Netzwerks zu senden, was zu einer effizienten Datenübertragung und -kommunikation in lokalen Netzwerken führt.

5.4 DNS, DHCP

DNS DNS (Domain Name System) ist ein hierarchisches und verteiltes System zur Auflösung von Domainnamen in IP-Adressen und umgekehrt. Es ermöglicht die Verwendung von leicht zu merkenden Domainnamen, wie z.B. ’example.com’, anstelle von IP-Adressen, wie z.B. ’192.0.2.1’, um auf Websites, Server und andere Ressourcen im Internet zuzugreifen. DNS funktioniert, indem es Anfragen von Clients entgegennimmt und diese Anfragen an DNS-Server weiterleitet, die die entsprechenden IP-Adressen zurückgeben.

DHCP DHCP (Dynamic Host Configuration Protocol) ist ein Netzwerkprotokoll, das automatisch IP-Adressen, Subnetzmasken, Standard-Gateways und andere Netzwerkkonfigurationen an Geräte in einem Netzwerk verteilt. DHCP ermöglicht es, dass Geräte, die sich in einem Netzwerk anmelden, automatisch eine IP-Adresse und andere Netzwerkkonfigurationen erhalten, ohne dass der Netzwerkadministrator manuell jeden Client konfigurieren muss. Dies erleichtert die Verwaltung von IP-Adressen in einem Netzwerk und reduziert potenzielle Konflikte.

5.5 TCP/UDP

TCP TCP (Transmission Control Protocol) ist ein zuverlässiges, verbindungsorientiertes Protokoll, das für die Übertragung von Daten über Netzwerke verwendet wird. Es gewährleistet die zuverlässige und geordnete Zustellung von Datenpaketen, indem es Bestätigungen für den Erhalt von Datenpaketen verwendet und bei Bedarf fehlende Pakete erneut sendet. TCP wird häufig für Anwendungen verwendet, bei denen eine fehlerfreie und vollständige Übertragung von Daten erforderlich ist, wie z.B. beim Abrufen von Webseiten, dem Senden von E-Mails oder dem Herunterladen von Dateien.

UDP UDP (User Datagram Protocol) ist ein unzuverlässiges, verbindungsloses Protokoll, das für die Übertragung von Daten über Netzwerke verwendet wird. Im Gegensatz zu TCP bietet UDP keine Garantie für die zuverlässige Zustellung von Daten oder die Beibehaltung der Reihenfolge. UDP wird häufig für Anwendungen verwendet, bei denen eine niedrige Latenz und eine schnelle Datenübertragung wichtiger sind als die Zuverlässigkeit, wie z.B. bei Voice-over-IP (VoIP), Videostreaming und Online-Spielen.

5.6 HTTPS, TLS/SSL, IPsec

HTTPS HTTPS (Hypertext Transfer Protocol Secure) ist ein Protokoll zur sicheren Übertragung von Daten über das Internet. Es basiert auf dem HTTP-Protokoll, verwendet jedoch zusätzliche Verschlüsselungsschichten, um die Vertraulichkeit und Integrität der übertragenen Daten zu gewährleisten. HTTPS wird häufig für die sichere Übertragung von sensiblen Daten wie Login-Informationen, Kreditkartendaten und persönlichen Informationen auf Websites verwendet. Es verwendet SSL (Secure Sockets Layer) oder TLS (Transport Layer Security) zur Verschlüsselung der Datenübertragung.

TLS/SSL TLS (Transport Layer Security) und sein Vorgänger SSL (Secure Sockets Layer) sind Verschlüsselungsprotokolle, die zur Sicherung der Kommunikation über das Internet verwendet werden. Sie bieten Sicherheitsebenen, indem sie Verschlüsselung, Authentifizierung und Integritätsschutz für die übertragenen Daten bieten. TLS wird häufig in Kombination mit anderen Protokollen wie HTTPS (sicheres HTTP), SMTPS (sicheres SMTP) und FTPS (sicheres FTP) eingesetzt, um eine sichere Kommunikation zwischen Clients und Servern zu gewährleisten.

IPsec IPsec (Internet Protocol Security) ist ein Protokollsuite, die zur Sicherung der Kommunikation auf Netzwerkebene verwendet wird. Es bietet Vertraulichkeit, Integrität und Authentizität für die übertragenen IP-Pakete, indem es Verschlüsselung und Authentifizierung verwendet. IPsec wird häufig in Virtual Private Networks (VPNs) eingesetzt, um eine sichere Kommunikation über unsichere Netzwerke wie das Internet zu ermöglichen. Es kann in zwei Modi betrieben werden: Transportmodus, der nur die Nutzdaten verschlüsselt, und Tunnelmodus, der das gesamte IP-Paket einschließlich Header verschlüsselt.

Hash Ein Hash ist eine kryptografische Funktion, die eine Eingabe (Nachricht, Daten) in eine feste Länge von Daten umwandelt, die als Hash-Wert bezeichnet wird. Der Hash-Wert ist im Allgemeinen eine eindeutige, nicht umkehrbare Darstellung der Eingabe. Hash-Funktionen werden häufig verwendet, um die Integrität von Daten zu überprüfen, da eine geringfügige Änderung an den Eingabedaten zu einem völlig anderen Hash-Wert führt. Sie werden auch in Passwort-Hashes, digitalen Signaturen und anderen kryptografischen Anwendungen verwendet.

Signatur Eine Signatur ist ein kryptografisches Verfahren, das verwendet wird, um die Authentizität, Integrität und Nichtabstreitbarkeit von Daten zu gewährleisten. Sie wird normalerweise mit einem privaten Schlüssel erstellt und kann mit einem öffentlichen Schlüssel überprüft werden. Durch Signieren von Daten mit einem privaten Schlüssel können Sender ihre Identität bestätigen und sicherstellen, dass die Daten nicht manipuliert wurden. Die Empfänger können dann die Signatur mit dem öffentlichen Schlüssel des Senders überprüfen, um die Echtheit der Daten zu bestätigen.

Zertifikate Ein Zertifikat ist eine digitale Bescheinigung, die die Identität einer Entität (Person, Organisation, Website) authentifiziert und von einer vertrauenswürdigen Zertifizierungsstelle (Certificate Authority, CA) signiert ist. Zertifikate enthalten öffentliche Schlüssel und andere Informationen über die identifizierte Entität. Sie werden häufig in der SSL/TLS-Verschlüsselung verwendet, um die Identität von Websites zu bestätigen und die sichere Kommunikation zu gewährleisten.

Certificate Authority Eine Certificate Authority (CA) ist eine vertrauenswürdige Organisation, die digitale Zertifikate ausstellt und signiert, um die Identität von Entitäten im Internet zu authentifizieren. CAs sind für die Verifizierung der Identität von Antragstellern, die Ausstellung von Zertifikaten und die Verwaltung von Zertifikatswiderrufung verantwortlich. Zu den bekannten CAs gehören Unternehmen wie Let's Encrypt, VeriSign und Comodo, die von den gängigen Webbrowsern und Betriebssystemen als vertrauenswürdig eingestuft werden.

5.7 Verschlüsselung

Symmetrische Verschlüsselung Die symmetrische Verschlüsselung ist ein Verfahren, bei dem derselbe Schlüssel zum Verschlüsseln und Entschlüsseln von Daten verwendet wird. Sowohl der Sender als auch der Empfänger müssen denselben geheimen Schlüssel kennen, um die Kommunikation zu verschlüsseln und zu entschlüsseln. Symmetrische Verschlüsselungsalgorithmen wie AES (Advanced Encryption Standard) sind schnell und effizient, werden jedoch oft durch das Schlüsselverwaltungsproblem eingeschränkt, da der Schlüssel sicher zwischen den Parteien ausgetauscht werden muss.

Asymmetrische Verschlüsselung Die asymmetrische Verschlüsselung, auch Public-Key-Verschlüsselung genannt, verwendet zwei unterschiedliche Schlüssel: einen öffentlichen Schlüssel und einen privaten Schlüssel. Der öffentliche Schlüssel wird zum Verschlüsseln von Daten verwendet und kann frei verteilt werden, während der private Schlüssel zum Entschlüsseln der Daten verwendet wird und geheim

gehalten werden muss. Dieses Verfahren ermöglicht sichere Kommunikation, da der private Schlüssel nicht offenbart wird. Asymmetrische Verschlüsselung wird häufig für digitale Signaturen, sichere E-Mail-Kommunikation und die Erstellung von SSL/TLS-Zertifikaten verwendet.

RADIUS RADIUS (Remote Authentication Dial-In User Service) ist ein Netzwerkprotokoll, das zur Authentifizierung, Autorisierung und Buchhaltung von Benutzern in einem Netzwerk verwendet wird, insbesondere in drahtlosen Netzwerken und virtuellen privaten Netzwerken (VPNs). RADIUS ermöglicht es Netzwerkgeräten wie WLAN-Zugriffspunkten, Switches und VPN-Konzentratoren, Benutzerinformationen über ein Netzwerk zu überprüfen und zu autorisieren, bevor sie den Netzwerkzugriff gewähren. Es bietet auch Funktionen zur Buchführung und Protokollierung von Benutzeraktivitäten im Netzwerk.

Pre-shared key Ein Pre-shared key (PSK) ist ein gemeinsamer geheimer Schlüssel, der zwischen zwei oder mehr Parteien vor der Kommunikation vereinbart wird. PSKs werden häufig in verschiedenen Sicherheitsprotokollen und -mechanismen verwendet, um die Authentizität und Vertraulichkeit der übertragenen Daten zu gewährleisten. Beispiele für den Einsatz von PSKs sind in WLAN-Netzwerken, VPN-Verbindungen und drahtlosen Sensornetzwerken, wo sie zur Authentifizierung und Verschlüsselung des Datenverkehrs verwendet werden. Im Gegensatz zu öffentlichen Schlüsselverfahren erfordern PSKs keine Infrastruktur für Zertifikate oder öffentliche Schlüsselverwaltung und sind daher oft einfacher einzurichten und zu verwalten.

5.8 Netzwerktypen

LAN Ein Local Area Network (LAN) ist ein Netzwerk, das Geräte in einem begrenzten geografischen Bereich wie einem Gebäude, einer Wohnung oder einem Campus miteinander verbindet. LANs ermöglichen den Austausch von Daten und Ressourcen wie Dateien, Druckern und Anwendungen zwischen den verbundenen Geräten. Sie werden häufig in Unternehmen, Bildungseinrichtungen und Privathaushalten eingesetzt und verwenden in der Regel Ethernet- oder WLAN-Technologien für die Verbindung der Geräte.

WAN Ein Wide Area Network (WAN) ist ein Netzwerk, das Geräte über große geografische Entfernungen miteinander verbindet, oft über öffentliche Telekommunikationsnetze wie das Internet oder private dedizierte Leitungen. WANs ermöglichen die Kommunikation zwischen entfernten Standorten, Niederlassungen und Rechenzentren und unterstützen verschiedene Dienste wie E-Mail, Webzugriff, VoIP und Videokonferenzen. WANs können sowohl öffentliche als auch private Verbindungen verwenden und erfordern in der Regel spezielle Netzwerkgeräte wie Router und Switches.

MAN Ein Metropolitan Area Network (MAN) ist ein Netzwerk, das mehrere LANs in einem geografisch begrenzten städtischen Gebiet miteinander verbindet. MANs bieten eine höhere Bandbreite und Reichweite als LANs und ermöglichen die Kommunikation zwischen verschiedenen Standorten innerhalb einer

Stadt oder einer städtischen Region. Sie werden häufig von Unternehmen, Regierungsbehörden und Serviceanbietern eingesetzt, um standortübergreifende Dienste wie VoIP, Datenübertragung und Videoüberwachung bereitzustellen.

GAN Ein Global Area Network (GAN) ist ein Netzwerk, das mehrere WANs miteinander verbindet und eine weltweite Kommunikation ermöglicht. GANs nutzen verschiedene Technologien wie Satellitenverbindungen, Unterseekabel und drahtlose Netzwerke, um eine nahtlose Konnektivität über große geografische Entfernungen hinweg zu gewährleisten. Sie werden häufig von multinationalen Unternehmen, Telekommunikationsunternehmen und Regierungsbehörden eingesetzt, um weltweite Kommunikationsdienste anzubieten und globale Geschäftsaktivitäten zu unterstützen.

SAN Ein Storage Area Network (SAN) ist ein dediziertes Netzwerk, das die Verbindung von Speichergeräten wie Festplatten, Solid-State-Laufwerken (SSDs) und Speicherarrays mit Servern ermöglicht. SANs werden häufig in Rechenzentren und Unternehmensumgebungen eingesetzt, um eine zentrale und hochverfügbare Speicherlösung bereitzustellen. Sie ermöglichen es mehreren Servern, gleichzeitig auf gemeinsame Speicherressourcen zuzugreifen, was eine flexible und skalierbare Speicherinfrastruktur bietet. SANs verwenden oft spezielle Netzwerkprotokolle wie Fibre Channel oder iSCSI zur Datenübertragung zwischen den Servern und Speichergeräten.

PAN Die Abkürzung 'PAN' steht für 'Personal Area Network' (Persönliches Netzwerk). Ein PAN ist ein Netzwerk, das sich in einem begrenzten Bereich um eine Person herum erstreckt und verschiedene Geräte miteinander verbindet, die sich in unmittelbarer Nähe befinden. Typische Geräte in einem PAN sind Computer, Smartphones, Tablets, Drucker, Headsets und andere persönliche elektronische Geräte.

5.9 Strukturierte Verkabelung

Primäre/Sekundäre/Tertiäre Verkabelung Bei der strukturierten Verkabelung werden die Kabeltypen je nach ihrer Verwendung in primäre, sekundäre und tertiäre Verkabelung unterteilt.

- Die primäre Verkabelung umfasst die Hauptverkabelung, die das gesamte Gebäude oder die gesamte Anlage abdeckt. Sie verbindet die Telekommunikationsräume, Verteiler und Etagenverteiler miteinander.

- Die sekundäre Verkabelung umfasst die Verkabelung innerhalb einzelner Etagen oder Abteilungen. Sie verbindet die Verteiler mit den Anschlussdosen und sorgt für die horizontale Verbindung zu den Endgeräten.

- Die tertiäre Verkabelung umfasst die Verkabelung innerhalb eines Raumes oder einer Arbeitsplatzgruppe. Sie umfasst die Anschlussdosen, Patchfelder und Patchkabel, die die Endgeräte mit dem Netzwerk verbinden.

Kabeltypen

Twisted Pair Twisted Pair ist ein Kabeltyp, der aus mehreren Kupferdrahtpaaren besteht, die miteinander verdrillt sind. Diese Verdrillung reduziert elektromagnetische Störungen und verbessert die Signalqualität. Twisted-Pair-Kabel werden häufig für die Verkabelung von Ethernet-Netzwerken, Telefonleitungen und anderen Datenübertragungsanwendungen verwendet. Es gibt verschiedene Kategorien von Twisted-Pair-Kabeln, wie z.B. Cat5e, Cat6 und Cat6a, die unterschiedliche Bandbreiten und Übertragungsgeschwindigkeiten bieten.

LWL LWL (Lichtwellenleiter) oder Glasfaserkabel bestehen aus dünnen Glas- oder Kunststofffasern, die Lichtsignale zur Übertragung von Daten verwenden. Sie bieten eine hohe Bandbreite, geringe Dämpfung und Immunität gegen elektromagnetische Interferenzen, was sie ideal für die Übertragung großer Datenmengen über große Entfernungen macht. LWL-Kabel werden häufig in Backbone-Netzwerken, Telekommunikationsnetzen und Rechenzentren eingesetzt, wo hohe Übertragungsraten und Zuverlässigkeit erforderlich sind.

5.10 VLAN

VLAN Ein Virtual Local Area Network (VLAN) ist ein logisches Netzwerk, das aus Geräten in verschiedenen physischen Netzwerken besteht, die auf der Grundlage von Portnummern, MAC-Adressen, Protokollen oder anderen Kriterien gruppiert sind. VLANs ermöglichen es, ein physisches Netzwerk in mehrere virtuelle Netzwerke zu unterteilen, wodurch die Sicherheit, Skalierbarkeit und Verwaltbarkeit verbessert werden. Sie können verwendet werden, um Benutzer in verschiedenen Abteilungen, virtuelle Maschinen, Gäste und andere Gruppen zu isolieren und den Datenverkehr zwischen den VLANs zu steuern.

5.11 Sicherheitskonzepte und -risiken: WEB, WPA

WEB Wired Equivalent Privacy (WEP) ist ein veraltetes Sicherheitsprotokoll, das zur Verschlüsselung von Daten in drahtlosen Netzwerken verwendet wurde. WEP verwendet symmetrische Verschlüsselung mit einem gemeinsamen Schlüssel, um die Vertraulichkeit des Datenverkehrs zu gewährleisten. Es war jedoch anfällig für verschiedene Sicherheitslücken und Schwachstellen, die es Angreifern ermöglichten, den WEP-Schlüssel zu knacken und auf den Netzwerkverkehr zuzugreifen. Aufgrund seiner Schwächen wurde WEP durch sicherere Protokolle wie WPA und WPA2 ersetzt.

WPA Wi-Fi Protected Access (WPA) ist ein Sicherheitsprotokoll, das entwickelt wurde, um die Sicherheit von drahtlosen Netzwerken zu verbessern und die Schwächen von WEP zu überwinden. WPA verwendet verbesserte Verschlüsselungsalgorithmen wie TKIP (Temporal Key Integrity Protocol) und AES (Advanced Encryption Standard), um die Vertraulichkeit und Integrität des Datenverkehrs zu gewährleisten. Es bietet auch Funktionen wie Pre-shared Key (PSK), 802.1X-Authentifizierung und dynamische Schlüsselwechsel, um die Sicherheit des drahtlosen Netzwerks weiter zu erhöhen.

5.12 Netzwerktopologien

Netzwerktopologien Eine Netzwerktopologie beschreibt die physische oder logische Anordnung von Geräten, Kabeln und anderen Komponenten in einem Netzwerk. Es gibt verschiedene Arten von Netzwerktopologien, die je nach den Anforderungen und den spezifischen Einsatzszenarien eines Netzwerks eingesetzt werden können.

- Die Bus-Topologie besteht aus einem einzigen Kommunikationskanal, der von allen Geräten im Netzwerk gemeinsam genutzt wird. Alle Geräte sind direkt mit diesem Bus verbunden, was zu einer einfachen und kostengünstigen Verkabelung führt. Ein Ausfall eines Geräts kann jedoch das gesamte Netzwerk beeinträchtigen.
- Die Stern-Topologie besteht aus einem zentralen Knoten, der alle anderen Geräte im Netzwerk verbindet. Alle Datenverbindungen laufen über diesen zentralen Knoten, was eine einfache Erweiterung und Verwaltung ermöglicht. Ein Ausfall des zentralen Knotens kann jedoch das gesamte Netzwerk lahmlegen.
- Die Ring-Topologie besteht aus einer Reihe von Geräten, die in einer geschlossenen Schleife verbunden sind. Daten werden in einer Richtung von einem Gerät zum nächsten übertragen, bis sie ihr Ziel erreichen. Diese Topologie bietet eine gleichmäßige Lastverteilung und hohe Zuverlässigkeit, ist jedoch anfällig für Ausfälle, da der Ausfall eines einzigen Geräts den gesamten Ring unterbrechen kann.
- Die Stern-basierte Ring-Topologie kombiniert die Vorteile der Stern- und Ring-Topologien, indem sie eine Sternstruktur mit einem Ringverbindungsnetzwerk innerhalb des Sterns verwendet. Dies bietet die Vorteile einer einfachen Erweiterung und Verwaltung sowie einer gleichmäßigen Lastverteilung und hohen Zuverlässigkeit.
- Die Mesh-Topologie besteht aus einer Reihe von Geräten, die direkt miteinander verbunden sind und mehrere Pfade zur Kommunikation bieten. Diese Topologie bietet eine hohe Redundanz und Ausfallsicherheit, ist jedoch aufgrund der großen Anzahl von Verbindungen zwischen den Geräten teuer und schwer zu verwalten.

5.13 Netzwerkplan

Netzwerkplan Ein Netzwerkplan ist eine schematische Darstellung eines Netzwerks, die die physische und logische Struktur sowie die Konnektivität der verschiedenen Netzwerkkomponenten zeigt. Ein Netzwerkplan kann verschiedene Elemente enthalten, wie z.B. Router, Switches, Server, Arbeitsstationen, Verbindungslinien, IP-Adressen und VLANs. Er dient als Referenzdokument für Netzwerkadministratoren und IT-Techniker, um das Netzwerkdesign zu visualisieren, Änderungen zu planen und Probleme zu diagnostizieren.

5.14 VPN

Funktionsweise von VPN Ein Virtual Private Network (VPN) ist eine Technologie, die es Benutzern ermöglicht, eine sichere Verbindung zu einem privaten Netzwerk über ein öffentliches Netzwerk wie das Internet herzustellen. VPNs verwenden Verschlüsselung und andere Sicherheitsmechanismen, um den Datenverkehr zwischen dem Benutzer und dem privaten Netzwerk zu schützen und die Privatsphäre und Sicherheit zu gewährleisten. VPNs können verwendet werden, um sichere Remote-Zugriffe auf Unternehmensnetzwerke, den Schutz der Privatsphäre bei der Internetnutzung und die Umgehung geografischer Einschränkungen zu ermöglichen.

VPN-Modelle Es gibt verschiedene VPN-Modelle, die je nach den Anforderungen und dem Einsatzzweck eines Netzwerks eingesetzt werden können:

- Remote Access VPN: Ermöglicht autorisierten Benutzern den sicheren Zugriff auf das Unternehmensnetzwerk von entfernten Standorten aus über das Internet.
- Site-to-Site VPN: Verbindet zwei oder mehr physische Standorte oder Netzwerke miteinander und ermöglicht den sicheren Austausch von Daten über das Internet.
- Extranet VPN: Ermöglicht autorisierten externen Benutzern oder Partnern den sicheren Zugriff auf bestimmte Ressourcen oder Dienste im Unternehmensnetzwerk.
- Intranet VPN: Bietet sicheren Zugriff auf interne Ressourcen und Dienste innerhalb eines Unternehmensnetzwerks für autorisierte Benutzer, unabhängig von ihrem Standort.

Tunneling Tunneling ist ein Prozess, bei dem Datenpakete in einem anderen Netzwerkprotokoll eingekapselt werden, um sie sicher über ein unsicheres Netzwerk zu übertragen. Bei der VPN-Kommunikation wird Tunneling verwendet, um den privaten Datenverkehr über das öffentliche Internet zu transportieren, ohne dass Dritte den Inhalt der Daten sehen oder manipulieren können. Die Daten werden verschlüsselt und in Pakete eines anderen Protokolls eingekapselt, bevor sie über das Internet gesendet werden. Am Zielort werden die Pakete wieder entkapselt und entschlüsselt, um die ursprünglichen Daten wiederherzustellen.

5.15 Serverarten

Mailserver Ein Mailserver ist ein spezieller Server, der E-Mails empfängt, speichert, weiterleitet und zustellt. Er fungiert als zentrale Komponente eines E-Mail-Systems und ermöglicht es Benutzern, E-Mails zu senden und zu empfangen. Mailserver verwenden verschiedene Protokolle wie SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol Version 3) und IMAP (Internet Message Access Protocol) für die Kommunikation zwischen den Clients und dem Server.

Webserver Ein Webserver ist ein Server, der Webinhalte wie Webseiten, Bilder, Videos und andere Dateien über das World Wide Web bereitstellt. Er empfängt HTTP-Anfragen von Webbrowsern und sendet die angeforderten Dateien als HTTP-Antworten zurück. Webserver hosten Websites und Webanwendungen und verwenden häufig Serversoftware wie Apache, Nginx, Microsoft IIS und andere.

Groupware Groupware bezeichnet Softwareanwendungen, die die Zusammenarbeit und Kommunikation in einer Gruppe oder Organisation erleichtern. Ein Groupware-Server ermöglicht es Benutzern, gemeinsam an Projekten zu arbeiten, Termine zu planen, Dokumente zu teilen, E-Mails zu senden und andere kollaborative Aktivitäten durchzuführen. Beispiele für Groupware-Server sind Microsoft Exchange Server, IBM Notes/Domino und Open-Xchange.

Datenbanken Ein Datenbankserver ist ein spezieller Server, der die Speicherung, Verwaltung und Abfrage von Daten in einer Datenbank ermöglicht. Er fungiert als zentrale Schnittstelle für Benutzer und Anwendungen, um auf die Datenbank zuzugreifen und mit ihr zu interagieren. Datenbankserver verwenden SQL (Structured Query Language) oder andere Abfragesprachen, um Daten zu manipulieren und abzurufen. SQL-Datenbanken, wie MySQL, PostgreSQL, Oracle Database und Microsoft SQL Server, folgen einem relationalen Datenbankmodell und speichern Daten in Tabellen, die durch Beziehungen miteinander verbunden sind. Im Gegensatz dazu verwenden NoSQL-Datenbanken, wie MongoDB, Couchbase und Cassandra, ein nicht-relationales Datenbankmodell, das keine festen Tabellenstrukturen erfordert. NoSQL-Datenbanken sind flexibler und skalierbarer für die Speicherung unstrukturierter Daten, wie beispielsweise in Big-Data-Anwendungen, aber SQL-Datenbanken sind oft besser geeignet für komplexe Abfragen und Transaktionen in traditionellen Unternehmensanwendungen.

Proxy Ein Proxyserver ist ein Server, der als Vermittler zwischen Clientgeräten und anderen Servern im Internet fungiert. Er empfängt Anfragen von Clientgeräten wie Webbrowsern und leitet sie an andere Server weiter. Proxyserver können verschiedene Funktionen erfüllen, darunter die Verbesserung der Sicherheit und Privatsphäre, die Optimierung der Netzwerk- und Webbeschleunigung sowie die Filterung und Blockierung von unerwünschtem Datenverkehr. Sie werden häufig in Unternehmensnetzwerken, Schulen und öffentlichen WLANs eingesetzt.

5.16 Sicherstellung des Betriebs

USV Eine USV (Unterbrechungsfreie Stromversorgung) ist eine elektrotechnische Vorrichtung, die dazu dient, elektronische Geräte vor Stromausfällen und Spannungsschwankungen zu schützen. Sie besteht aus einem Akku, der bei einem Stromausfall automatisch einspringt und den angeschlossenen Geräten kontinuierlich Strom liefert, bis die Stromversorgung wiederhergestellt ist oder die Geräte sicher heruntergefahren werden können. USVs sind besonders wichtig für Server und andere kritische IT-Systeme, um Datenverluste und Betriebsunterbrechungen zu vermeiden.

RAID RAID (Redundant Array of Independent Disks) ist eine hardwaretechnische Methode zur Erhöhung der Datensicherheit und -verfügbarkeit durch die redundante Speicherung von Daten über mehrere Festplatten. Es gibt verschiedene RAID-Level, die unterschiedliche Methoden der Datenspiegelung, Parität und Striping verwenden, um die Leistung, Zuverlässigkeit und Kapazität der Speicherlösung zu optimieren. RAID ermöglicht es, Daten auch bei Ausfall einer Festplatte weiterhin verfügbar zu halten und die Ausfalltoleranz des Systems zu verbessern.

Back-ups Back-ups sind eine softwaretechnische Maßnahme zur Sicherung und Wiederherstellung von Daten im Falle von Datenverlust, Hardwarefehlern, menschlichen Fehlern oder Katastrophen. Durch regelmäßige Back-ups werden Kopien der Daten erstellt und an einem sicheren Ort gespeichert, um im Bedarfsfall die Wiederherstellung der Daten zu ermöglichen. Back-ups können auf lokalen Speichermedien wie Festplatten oder Bandlaufwerken sowie in der Cloud gespeichert werden. Sie sind ein wichtiger Bestandteil der Datensicherungsstrategie eines Unternehmens und dienen dem Schutz vor Datenverlust und Geschäftsunterbrechungen.

5.17 Firewall

Firewall Eine Firewall ist eine Sicherheitsvorrichtung, die dazu dient, ein Netzwerk vor unerwünschtem Zugriff, Datenverkehr und potenziell schädlichen Angriffen aus dem Internet oder anderen Netzwerken zu schützen. Sie überwacht den Datenverkehr zwischen einem internen Netzwerk und externen Netzwerken und filtert oder blockiert Datenpakete basierend auf vordefinierten Sicherheitsregeln. Firewalls können auf Netzwerkgeräten wie Routern, Switches oder dedizierten Firewall-Appliances implementiert werden und bieten Funktionen wie Paketfilterung, Zustandsinspektion, Anwendungsfilterung und Virtual Private Network (VPN)-Unterstützung.

5.18 Portsecurity, Port-Forwarding

Portsecurity Portsecurity ist eine Sicherheitsfunktion, die dazu dient, unautorisierten Zugriff auf Netzwerkgeräte wie Switches zu verhindern, indem sie den Zugriff auf bestimmte Netzwerkports basierend auf der MAC-Adresse des angeschlossenen Geräts beschränkt. Durch die Konfiguration von Portsecurity können Administratoren die Anzahl der Geräte begrenzen, die an einen Switchport angeschlossen werden können, und unbekannte oder nicht autorisierte Geräte blockieren oder alarmieren, die versuchen, auf das Netzwerk zuzugreifen.

Port-Forwarding Port-Forwarding ist eine Netzwerkkonfigurationstechnik, die verwendet wird, um den Datenverkehr von einem bestimmten Port auf einem Netzwerkgerät an einen anderen Port auf einem anderen Gerät in einem lokalen Netzwerk oder im Internet weiterzuleiten. Es ermöglicht die Weiterleitung eingehender Netzwerkanfragen von einem externen Netzwerk an bestimmte Dienste oder Anwendungen, die

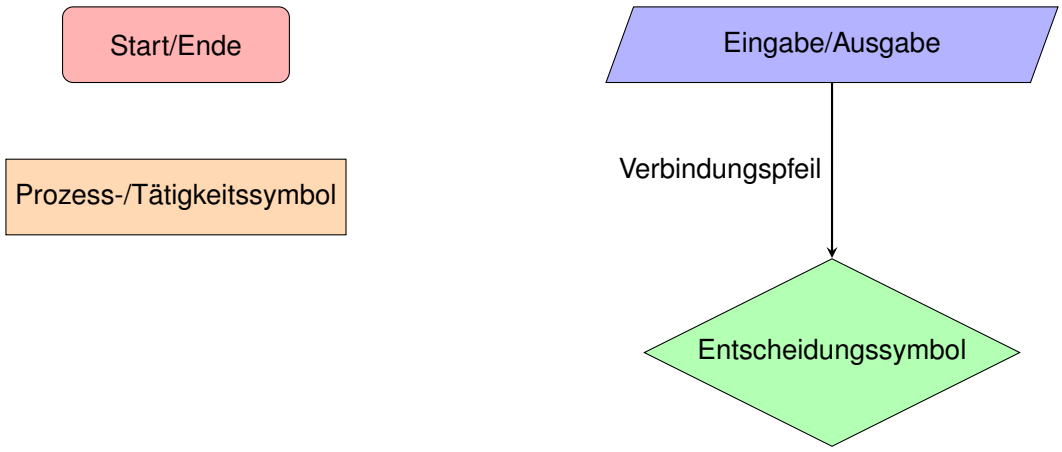
auf internen Servern oder Endgeräten ausgeführt werden. Port-Forwarding wird häufig in Heimnetzwerken, Unternehmen und Rechenzentren eingesetzt, um den Zugriff auf interne Ressourcen wie Webserver, FTP-Server, Spiele-Server und Remote-Desktop-Verbindungen zu ermöglichen.

6 Softwareentwicklung

6.1 Algorithmen

- Abbildung der Kontrollstrukturen mittels Struktogramm, PAP oder Pseudocode als didaktisches Hilfsmittel
- grundlegende Algorithmen kennen, eigene Algorithmen auch programmiersprachenfrei formulieren und zur Lösung von Problemen, z.B. in einem IT-System bzw. einer Softwareanwendung einsetzen
- Entwickeln und Darstellen von Programmlogiken unabhängig von der Programmiersprache, z.B. mithilfe von Struktogrammen nach Nassi-Shneidermann sowie Strukturdiagrammen und Verhaltensdiagrammen aus der UML
- Rekursion: Funktionsweise, Vor-/Nachteile
- Algorithmen implementieren/durchspielen
  - Mittelwert
  - doppelte Einträge in einem Array finden/löschen
  - Dateibäume rekursiv kopieren
  - (Zinses-)Zinsberechnung
  - Planen eines regelmäßigen Backups
  - Ablauf einer Benutzerauthentifizierung an einer Website
  - Abbuchen von einem Konto
  - Lineare Suche
  - Binäre Suche
  - Bubble Sort

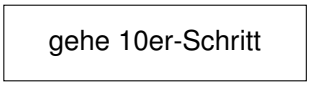
6.1.1 Flussdiagramm



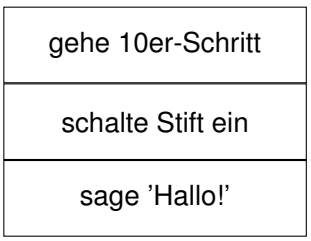
6.1.2 Struktogramm (Nassi-Shneiderman-Diagramm)

Quelle: Lehrerfortbildung-bw.de [23]

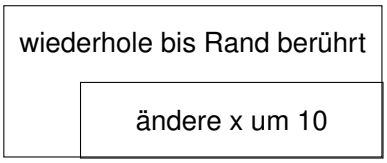
Anweisung



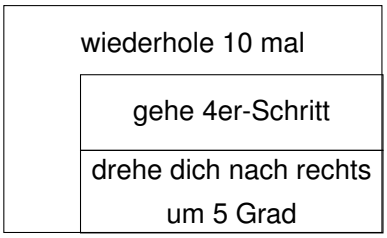
Sequenz



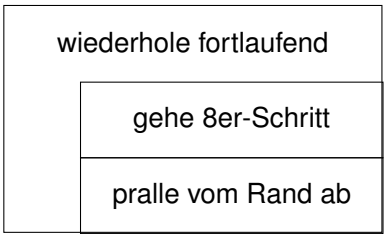
Schleife mit Bedingung



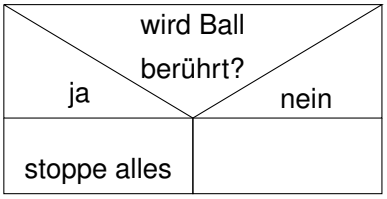
Schleife mit Zähler



Endlosschleife



Verzweigung mit Alternative



6.2 Schnittstellen, APIs, Datenaustausch

- Datenaustauschformate: CSV, XML, JSON
- XML
  - Wohlgeformtheit, Validität
  - DTD, Schema, RelaxNG, Schematron
  - XSLT, XSL-FO
- JSON
  - Syntax, Vor-/Nachteile, Einsatzgebiete
- REST
  - Adressierbarkeit, Zustandslosigkeit, einheitliche Schnittstelle (uniform interface), Ressource vs. Repräsentation
- Webservices
  - SOAP

6.3 Objektorientierung

In der Objektorientierung geht es darum, die reale Welt im Programm abzubilden. Um dies zu erreichen, werden verschiedene Konzepte gebündelt, die man zusammen unter der **objektorientierten Programmierung** kennt.

6.3.1 Objekte und Klassen

Um Informationen zu speichern und zu manipulieren, benötigen wir etwas, was mit Daten arbeiten kann. Etwas, was Daten halten und verändern kann.

Klasse Eine Klasse ist ein Bauplan, in dem definiert wird, was unsere Entität **ist** und was sie **kann**.

Hierfür sprechen wir von **Attributen** und **Methoden**.

Attribute sind Container, die Daten über und / oder für die Entität speichern. Methoden hingegen sind vordefinierte Programmabläufe, die eine Entität ausführen kann, die Daten manipulieren.

Beispiel: Als Klasse könnte man **Auto** definieren. Dieses Auto besitzt die Attribute:

- Marke
- Baujahr
- Motortyp

und die Methoden:

- fahren()
- tanken()

Objekt Objekte sind die zur Laufzeit **instanziierten** Klassen. Das bedeutet, dass auf Basis der vorher definierten Klasse ein Objekt erstellt wird. Die Attribute des Objekts macht es einzigartig. Das Objekt kann die Informationen in seinen Attributen verwenden und ihre Methoden ausführen, um Daten zu manipulieren.

6.3.2 Vererbung

1. Vererbung

- Prinzipien der OOP
  - Begriffe der OOP erläutern: Attribut, Nachricht/Methodenaufruf, Persistenz, Schnittstelle/API/Interface, Polymorphie, Vererbung
  - Bestandteile von Klassen
  - Unterschied Klasse/Objekt
  - Unterschied Klasse/Interface
  - Erklärung Klassenbibliothek vs. Framework
  - Klassenbeziehungen: Assoziation, Aggregation, Komposition, Spezialisierung, Generalisierung
- Unterschied statische/nicht-statische Methoden und Attribute
- Datenstrukturen (Baum, Array)

- funktionale Aspekte in modernen Sprachen: Lambda-Ausdrücke, Functional Interfaces, Map/Filter/Reduce, deklarativ vs. imperativ

6.4 Programmiersprachen

- Programmierparadigmen: unstrukturiert, strukturiert, prozedural, funktional, objektorientiert, logisch
- imperativ vs. deklarativ
- synchrone vs. asynchrone Programmierung
- Herausforderungen paralleler Programmierung
- Eigenschaften funktionaler Programmierung
  - Pattern Matching

6.5 Allgemeines Fehlerhandling bei Programmen

Exception Exceptions sind Ereignisse, die während der Programmausführung auftreten und den normalen Ablauf unterbrechen. Sie signalisieren, dass ein Fehler oder eine unerwartete Situation aufgetreten ist, die nicht von der aktuellen Programmausführung behandelt werden kann. Exceptions können verschiedene Ursachen haben, wie z.B. ungültige Eingaben, nicht verfügbare Ressourcen oder interne Fehler. In vielen Programmiersprachen können Exceptions mit try-catch-Blöcken behandelt werden, wodurch der Programmfluss kontrolliert und Fehlerbehandlungsroutinen ausgeführt werden können.

Return/Exit Codes Return- oder Exit-Codes sind numerische Werte, die von einem Programm zurückgegeben werden, um den Status der Programmausführung anzuzeigen. Sie werden häufig verwendet, um anzuzeigen, ob ein Programm erfolgreich ausgeführt wurde oder ob ein Fehler aufgetreten ist. Üblicherweise wird ein Wert ungleich Null verwendet, um anzuzeigen, dass ein Fehler aufgetreten ist, während der Wert Null normalerweise für eine erfolgreiche Ausführung steht. Diese Codes können von aufrufenden Programmen oder Skripten verwendet werden, um auf den Erfolg oder das Scheitern der Ausführung zu reagieren und entsprechende Maßnahmen zu ergreifen.

6.6 Rechnerarchitektur

CPU Die CPU (Central Processing Unit) ist das Herzstück eines Computers und ist für die Ausführung von Programmen und die Verarbeitung von Daten verantwortlich. Sie führt die Befehle aus, die von der Software geladen werden, und koordiniert die Operationen anderer Hardwarekomponenten. Die CPU besteht aus verschiedenen Einheiten, darunter die Rechen- und Steuerungseinheiten, sowie Register und Caches, die zur temporären Speicherung von Daten und Befehlen verwendet werden.

BUS Der Bus ist ein System aus elektrischen Leitungen oder Verbindungen, das die Kommunikation zwischen den verschiedenen Komponenten eines Computers ermöglicht. Er dient als Datenübertragungsweg für die Übertragung von Daten, Befehlen und Steuersignalen zwischen der CPU, dem Arbeitsspeicher, den Peripheriegeräten und anderen Komponenten des Computers. Der Bus kann intern, um die Kommunikation innerhalb des Computers zu ermöglichen, oder extern, um die Kommunikation mit externen Geräten wie Festplatten und Druckern zu ermöglichen, sein.

Speicher und deren Adressierung Der Speicher eines Computers besteht aus verschiedenen Typen von Speichermedien, darunter der Hauptspeicher (RAM), der Cache-Speicher und der Massenspeicher (Festplatte, SSD). Der Hauptspeicher dient zur temporären Speicherung von Daten und Programmen, die während der Ausführung vom Prozessor benötigt werden. Die Adressierung des Speichers erfolgt über ein Adressbus-System, das es der CPU ermöglicht, auf bestimmte Speicheradressen zuzugreifen und Daten zu lesen oder zu schreiben. Die Größe des Adressbusses bestimmt die maximale Speicherkapazität, die von der CPU adressiert werden kann, während die Breite des Datenbusses die Anzahl der Datenbits angibt, die gleichzeitig übertragen werden können.

6.7 Lizenzen

Open Source Open-Source-Lizenzen sind Lizenzvereinbarungen, die es Entwicklern erlauben, den Quellcode einer Software frei einzusehen, zu modifizieren und weiterzuverbreiten. Sie fördern die Zusammenarbeit, Transparenz und den freien Austausch von Software und ermöglichen es Entwicklern, auf bereits existierende Codebasen aufzubauen und diese weiterzuentwickeln. Beispiele für Open-Source-Lizenzen sind die GPL, Apache License, MIT License und BSD License.

Proprietär Proprietäre Lizenzen sind Lizenzvereinbarungen, die es den Rechteinhabern erlauben, die Verwendung, Verbreitung und Modifikation ihrer Software einzuschränken und zu kontrollieren. Sie bieten in der Regel eingeschränkten oder keinen Zugriff auf den Quellcode und erfordern eine Lizenzgebühr oder andere Einschränkungen für die Nutzung der Software. Proprietäre Software wird häufig von Unternehmen entwickelt und vertrieben und umfasst Produkte wie Microsoft Windows, Adobe Photoshop und Oracle Database.

EULA Eine Endbenutzer-Lizenzvereinbarung (EULA) ist eine rechtliche Vereinbarung zwischen dem Hersteller oder Anbieter von Software und dem Endbenutzer, die die Bedingungen für die Nutzung der Software festlegt. Sie regelt die Rechte und Pflichten der Benutzer in Bezug auf die Software, einschließlich der Lizenzbedingungen, Nutzungseinschränkungen, Garantieausschlüsse und Haftungsbeschränkungen. Benutzer müssen der EULA zustimmen, bevor sie die Software installieren oder verwenden dürfen.

OEM Original Equipment Manufacturer (OEM) bezieht sich auf Softwarelizenzen, die von Hardwareherstellern erworben und auf ihren Geräten vorinstalliert sind. Diese Lizenzen gelten nur für die spezifische Hardware, auf der sie vorinstalliert sind, und sind in der Regel nicht übertragbar. OEM-Lizenzen werden oft zu einem reduzierten Preis angeboten und enthalten möglicherweise Einschränkungen oder zusätzliche Bedingungen im Vergleich zu regulären Einzelhandelslizenzen.

GNU Die GNU General Public License (GPL) ist eine weit verbreitete Open-Source-Lizenz, die von der Free Software Foundation (FSF) entwickelt wurde. Sie gewährt Benutzern das Recht, die Software frei zu verwenden, zu modifizieren und weiterzuverbreiten, solange sie die Bedingungen der Lizenz einhalten. Die GPL ist eine sogenannte Copyleft-Lizenz, die sicherstellt, dass abgeleitete Werke unter denselben Lizenzbedingungen veröffentlicht werden müssen.

Apache 2.0 Die Apache License 2.0 ist eine Open-Source-Lizenz, die von der Apache Software Foundation entwickelt wurde. Sie ermöglicht es Benutzern, die Software frei zu verwenden, zu modifizieren und weiterzuverbreiten, sowohl für kommerzielle als auch für nichtkommerzielle Zwecke. Die Apache License 2.0 ist eine permissive Lizenz, die Benutzern mehr Freiheiten gewährt als die GPL, da sie keine Anforderungen für die Veröffentlichung des Quellcodes für abgeleitete Werke enthält.

GPL 3.0 Die GNU General Public License (GPL) Version 3.0 ist eine aktualisierte Version der GPL, die von der Free Software Foundation veröffentlicht wurde. Sie enthält aktualisierte Bestimmungen, um mit neuen Entwicklungen in der Softwarebranche Schritt zu halten, einschließlich der Behandlung von Patenten, digitalen Rechteverwaltungen und Webanwendungen. Die GPL 3.0 ist eine Copyleft-Lizenz, die sicherstellt, dass abgeleitete Werke unter denselben Lizenzbedingungen veröffentlicht werden müssen, und sie enthält zusätzliche Bestimmungen zum Schutz der Freiheiten der Benutzer.

6.8 Informationspflicht

Namensrecht Das Namensrecht regelt die rechtlichen Aspekte der Verwendung von Namen von Personen oder Unternehmen. Es schützt das Recht einer Person oder Firma, ihren eigenen Namen zu verwenden und verbietet die unbefugte Nutzung oder Verwendung des Namens einer anderen Person oder Firma, insbesondere wenn dadurch Verwechslungen oder Irreführungen entstehen könnten.

Markenrecht Das Markenrecht regelt den Schutz von Marken und Kennzeichen, die verwendet werden, um Produkte oder Dienstleistungen eines Unternehmens von denen anderer zu unterscheiden. Es gewährt einem Unternehmen das ausschließliche Recht, seine Marke zu verwenden und verbietet anderen die unbefugte Verwendung oder Nachahmung dieser Marke, insbesondere wenn dadurch Verwechslungen oder Irreführungen entstehen könnten.

Urheberrecht Das Urheberrecht schützt die geistigen Eigentumsrechte an kreativen Werken wie Literatur, Kunst, Musik, Filmen und Software. Es gewährt dem Urheber das ausschließliche Recht, sein Werk zu reproduzieren, zu verbreiten, öffentlich aufzuführen oder anzuzeigen, abgeleitete Werke zu erstellen und darüber zu entscheiden, wie das Werk verwendet wird. Das Urheberrecht entsteht automatisch, sobald ein Werk in einer fixierten Form erstellt wird, und erfordert keine formale Registrierung.

Nutzungsrecht Das Nutzungsrecht ist das Recht, ein urheberrechtlich geschütztes Werk zu nutzen oder zu verwenden, das dem Inhaber des Urheberrechts gewährt wird. Es erlaubt Dritten die Verwendung des Werks unter bestimmten Bedingungen, die in einer Lizenzvereinbarung festgelegt sind. Diese Bedingungen können die Art und Weise der Nutzung, die Dauer der Nutzung, die Gebühren oder Lizenzgebühren sowie andere Einschränkungen oder Anforderungen umfassen.

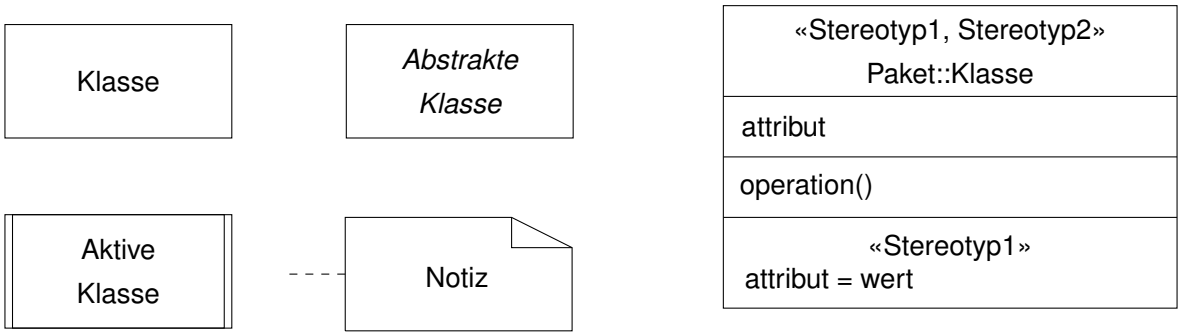
Persönlichkeitsrecht Das Persönlichkeitsrecht schützt die persönlichen und individuellen Rechte einer Person, insbesondere das Recht auf Privatsphäre, das Recht auf informationelle Selbstbestimmung und das Recht am eigenen Bild. Es verbietet die unerlaubte Veröffentlichung oder Verbreitung persönlicher Informationen oder Abbildungen einer Person, insbesondere wenn dadurch die Persönlichkeitsrechte verletzt werden könnten.

Unlauterer Wettbewerb Der unlautere Wettbewerb bezieht sich auf rechtswidrige Praktiken oder Verhaltensweisen von Unternehmen, die darauf abzielen, den Wettbewerb zu verzerren oder andere Marktteilnehmer zu benachteiligen. Dazu gehören beispielsweise irreführende Werbung, Verleumdung, Rufschädigung, Preisabsprachen, Behinderung des Marktzugangs oder Verletzungen von geistigen Eigentumsrechten. Der unlautere Wettbewerb wird durch Gesetze und Vorschriften geregelt, die darauf abzielen, fairen und offenen Wettbewerb zu fördern und den Schutz von Verbrauchern und Unternehmen zu gewährleisten.

6.9 UML Diagramme

Quelle: Oose.de - Notationsübersicht UML [26] & IHK Belegsatz

6.9.1 Klassendiagramm



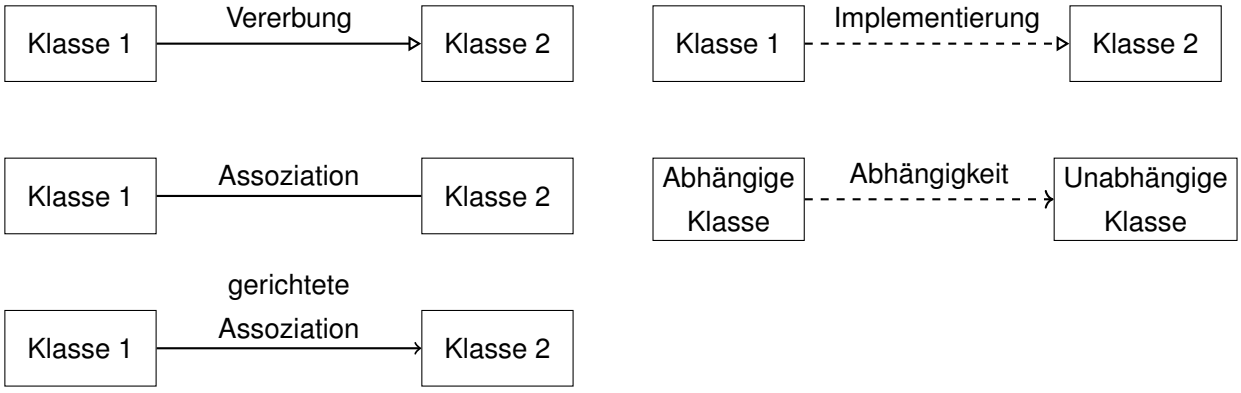
Sichtbarkeit:

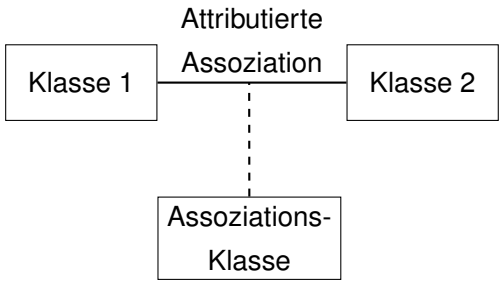
- Öffentlich (+)
- Privat (-)
- Geschützt (#)
- Paket (~)
- Abgeleitet (/)
- Statisch (unterstrichen)

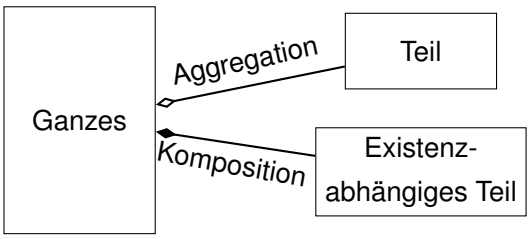
Syntax für Attribute:
Sichtbarkeit Attributname: Typ {Eigenschaften}

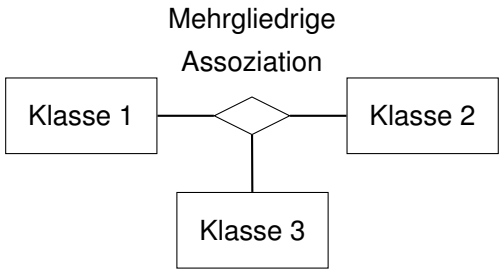
Syntax für Methoden:
Sichtbarkeit Methodenname(parpameter1: Typ, ...): Rückgabewert {Eigenschaften}

Eigenschaften:
{static,final, ...}









Assoziation In der Softwareentwicklung und im Bereich der Datenmodellierung bezeichnet eine Assoziation die Beziehung zwischen zwei oder mehr Klassen oder Objekten. Sie beschreibt, wie Objekte miteinander verbunden sind oder wie sie aufeinander verweisen können. Assoziationen können unidirektional oder bidirektional sein und verschiedene Arten von Beziehungen darstellen, wie z.B. Eins-zu-Eins, Eins-zu-Viele oder Viele-zu-Viele.

Aggregation Aggregation ist eine Beziehung zwischen Objekten, bei der ein Objekt (der Aggregator) ein oder mehrere andere Objekte (die Aggregate) enthalten kann. Die Aggregation beschreibt eine 'Teil-von'-Beziehung, bei der die Lebensdauer der Aggregate nicht unbedingt von der Lebensdauer des Aggregators abhängt. Ein Beispiel für Aggregation ist die Beziehung zwischen einem Unternehmen (Aggregator) und seinen Abteilungen (Aggregate).

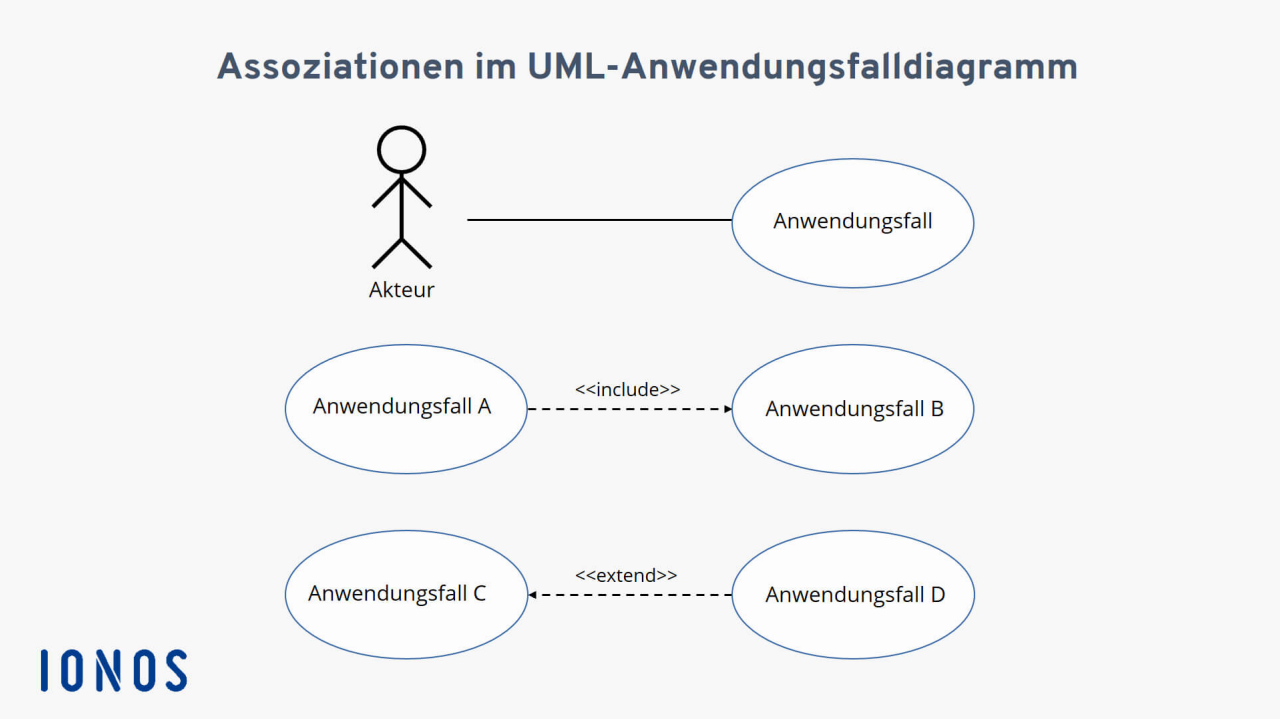
Komposition Komposition ist eine spezielle Form der Aggregation, bei der die Aggregate eng mit dem Aggregator verbunden sind und keine unabhängige Existenz haben können. Das bedeutet, dass die Lebensdauer der Aggregate von der Lebensdauer des Aggregators abhängt. Ein Beispiel für Komposition ist die Beziehung zwischen einem Haus (Aggregator) und seinen Zimmern (Aggregate).

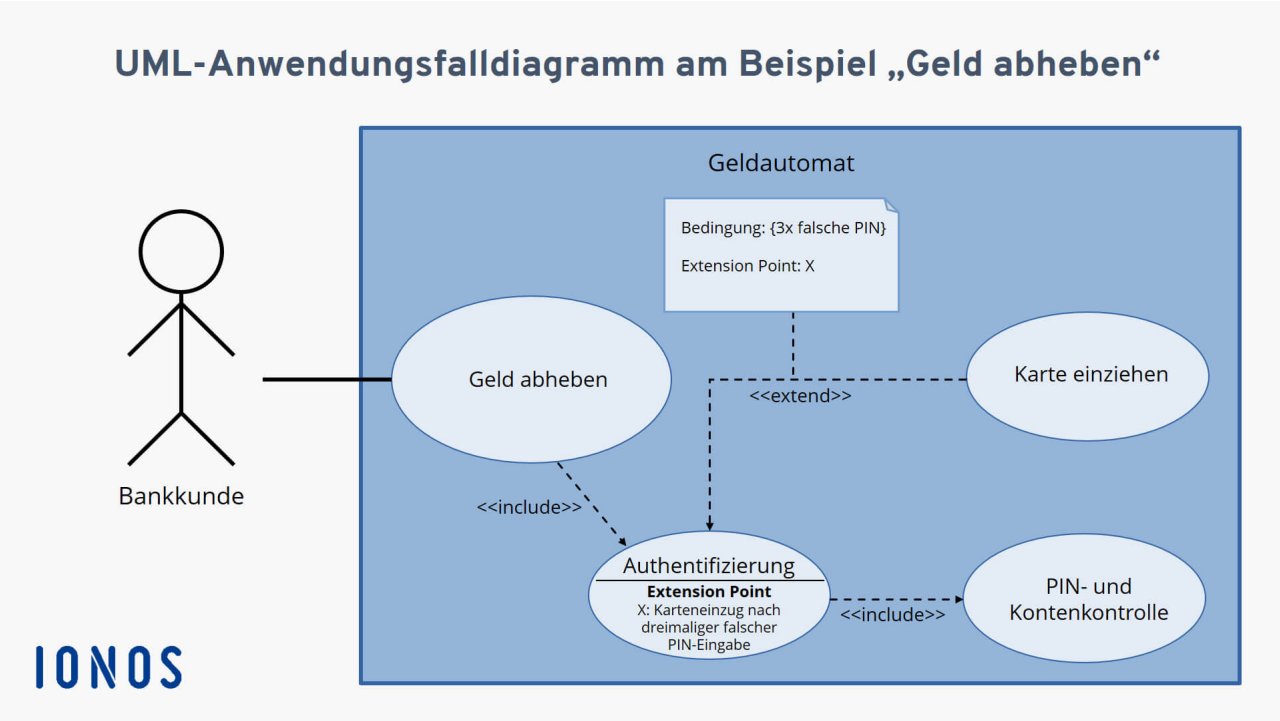
Spezialisierung Spezialisierung ist ein Konzept der Vererbung in der objektorientierten Programmierung, bei dem eine Unterklasse (Spezialisierung) von einer Oberklasse (Generalisierung) abgeleitet wird. Die Spezialisierung erweitert die Eigenschaften und Verhaltensweisen der Oberklasse und fügt spezifische Merkmale hinzu, um die Unterschiede zwischen den Objekten zu modellieren. Ein Beispiel für Spezialisierung ist die Ableitung von spezifischen Tierklassen wie Hund und Katze von der allgemeinen Tierklasse.

Generalisierung Generalisierung ist das Gegenteil von Spezialisierung und bezeichnet den Prozess, bei dem eine allgemeine Oberklasse (Generalisierung) geschaffen wird, um gemeinsame Eigenschaften und Verhaltensweisen von mehreren spezifischen Klassen (Spezialisierungen) zu repräsentieren. Die Generalisierung erfasst die Gemeinsamkeiten zwischen den Objekten und ermöglicht eine einheitliche Behandlung. Ein Beispiel für Generalisierung ist die Schaffung einer Oberklasse Fahrzeug, die gemeinsame Eigenschaften von Autos, Lastwagen und Motorrädern zusammenfasst.

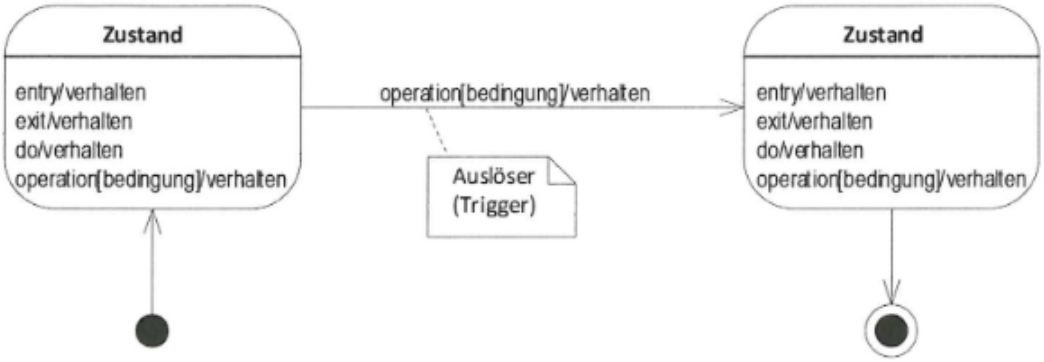
6.9.2 Use-Case-Diagramm (Anwendungsfalldiagramm)

Quelle: Ionos [20]

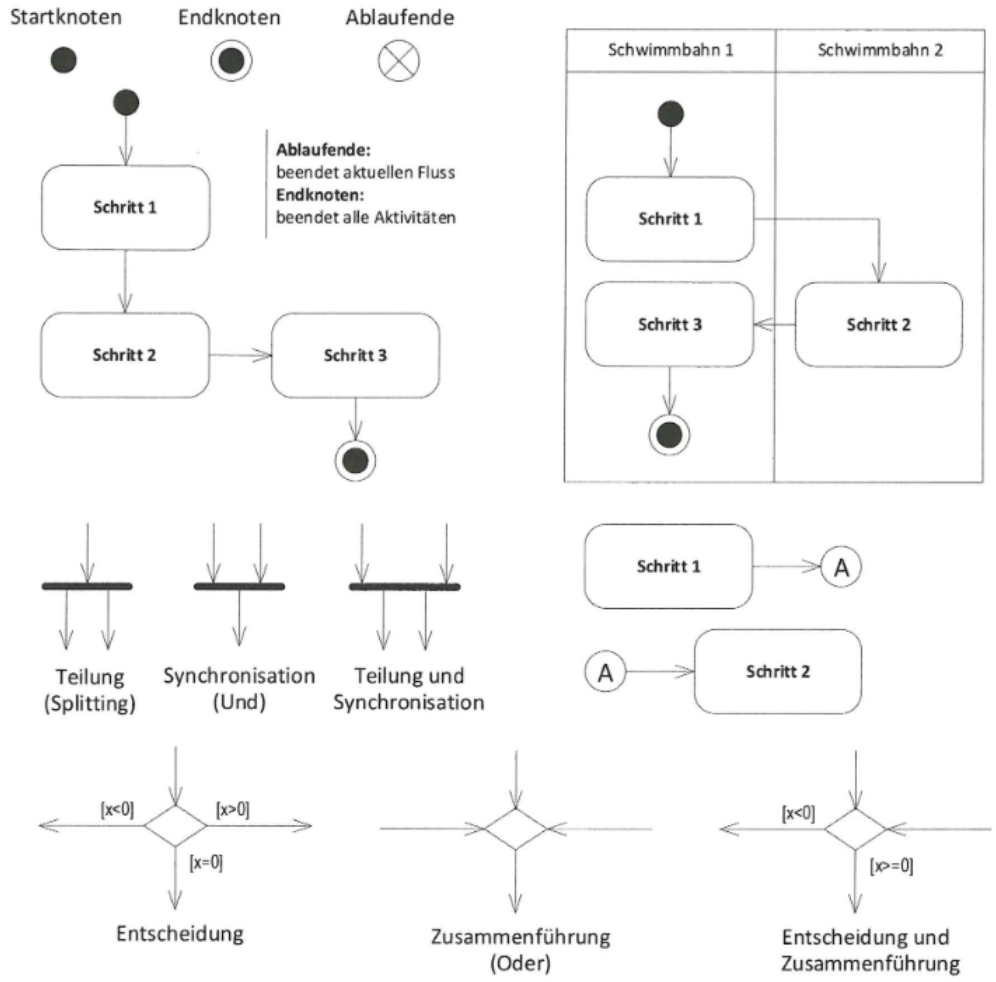




6.9.3 Zustandsdiagramm



6.9.4 Aktivitätsdiagramm



6.10 Softwarearchitektur

- Bottom-Up- und Top-Down-Verfahren bei der Modellierung erläutern
- Funktion/Vorteile der Modularisierung von Programmen
- Softwarearchitektur
  - 3-Schichten-Modell
  - Model View Controller (MVC)
  - Model View Presenter (MVP)
  - Model-View-ViewModel (MVVM)
  - REST

Bottom-Up

3-Schichten-Modell Das 3-Schichten-Modell ist ein Architekturmuster zur Strukturierung von Anwendungen, bei dem die Anwendung in drei Schichten unterteilt wird: die Präsentationsschicht (Presentation Layer), die Logikschicht (Business Logic Layer) und die Datenschicht (Data Layer). Jede Schicht hat eine klar definierte Verantwortung und Aufgabe: Die Präsentationsschicht ist für die Darstellung der Benutzeroberfläche und die Interaktion mit dem Benutzer zuständig, die Logikschicht enthält die Geschäftslogik und die Anwendungslogik und die Datenschicht kümmert sich um den Zugriff auf die Datenbank oder andere externe Datenquellen.

Model View Controller (MVC) MVC ist ein Entwurfsmuster zur Strukturierung von Benutzeroberflächen in Softwareanwendungen. Es teilt die Anwendung in drei Hauptkomponenten auf: das Modell (Model), die Ansicht (View) und den Controller. Das Modell repräsentiert die Daten und die Geschäftslogik der Anwendung, die Ansicht ist für die Darstellung der Benutzeroberfläche zuständig, und der Controller nimmt Benutzereingaben entgegen, verarbeitet sie und aktualisiert das Modell und die Ansicht entsprechend.

Model View Presenter (MVP) MVP ist ein Variante des MVC-Musters, bei dem der Presenter als Vermittler zwischen dem Modell und der Ansicht fungiert. Der Presenter übernimmt die Aufgabe der Steuerung der Benutzerinteraktionen und der Aktualisierung des Modells und der Ansicht. Im Gegensatz zum MVC-Muster ist die Ansicht im MVP-Muster passiv und enthält keine Logik, was zu einer besseren Trennung von Präsentation und Geschäftslogik führt.

Model View ViewModel (MVVM) MVVM ist ein weiteres Entwurfsmuster zur Strukturierung von Benutzeroberflächen, das häufig in der Entwicklung von Desktop- und Webanwendungen verwendet wird. Es ähnelt dem MVP-Muster, aber anstelle eines Presenters wird ein ViewModel verwendet, um die Präsentation der Daten zu steuern. Das ViewModel stellt eine abstrakte Repräsentation der Ansicht dar und erleichtert die Bindung von Daten zwischen dem Modell und der Ansicht.

REST REST (Representational State Transfer) ist ein Architekturstil für die Entwicklung von verteilten Systemen, insbesondere Webanwendungen. Es basiert auf dem Prinzip, dass Ressourcen über einheitliche Schnittstellen (URLs) angesprochen und durch die standardisierten HTTP-Methoden (GET, POST, PUT, DELETE) manipuliert werden können. RESTful-Services verwenden keine speziellen Protokolle oder Technologien, sondern nutzen die vorhandenen HTTP-Standards für die Kommunikation zwischen Client und Server.

6.11 Softwareergonomie

- Mock-up
- Usability vs. User-Experience
- Entwurf der Bildschirmausgabemasken (Softwareergonomie, Barrierefreiheit)
- Barrierefreiheit bzw. Inklusives Design

6.12 Software Engineering

- Entwicklungsprozesse wie das Wasserfallmodell
- Iterative Modelle, z.B. Spiralmodell, V-Modell (XT)
- Agile Modelle: Scrum, Extreme Programming, Kanban
- Top-Down-Entwurf vs. Bottom-Up-Entwurf

6.13 Design Patterns

- Design Patterns kennen/erklären/implementieren
  - Singleton
  - Observer
  - Factory
  - Strategy
  - Decorator

  - MVC

6.14 Softwarequalität

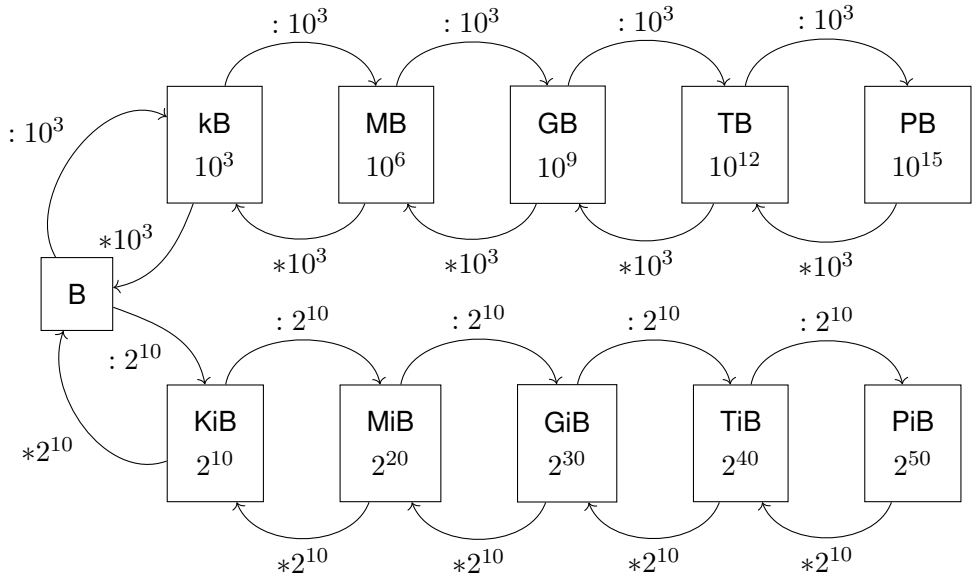
- Software-Qualitätsmerkmale nach ISO 9126 nennen und erläutern
  - Funktionalität: Angemessenheit, Interoperabilität, Ordnungsmäßigkeit, Richtigkeit, Sicherheit
  - Änderbarkeit: Analysierbarkeit, Modifizierbarkeit, Testbarkeit, Stabilität
  - Übertragbarkeit: Anpassbarkeit, Austauschbarkeit, Installierbarkeit, Koexistenz
  - Effizienz: Verbrauchsverhalten, Zeitverhalten
  - Zuverlässigkeit: Fehlertoleranz, Reife, Wiederherstellbarkeit
  - Benutzbarkeit: Attraktivität, Bedienbarkeit, Erlernbarkeit, Verständlichkeit
- Software-Qualitätsmerkmale nach ISO 25010 nennen und erläutern
  - Functional Suitability: Functional Completeness, Functional Correctness, Functional Appropriateness
  - Performance Efficiency: Time Behaviour, Resource Utilization, Capacity
  - Compatibility: Co-existence, Interoperability
  - Usability: Appropriateness Recognizability, Learability, Operability, User Error Protection, User Interface Aesthetics, Accessibility
  - Reliability: Maturity, Availability, Fault Tolerance, Recoverability
  - Security: Confidentiality, Integrity, Non-repudiation, Authenticity, Accountability
  - Maintainability: Modularity, Reusability, Analysability, Modifiability, Testability
  - Portability: Adaptability, Installability, Replaceability
- Maßnahmen zur Qualitätssicherung
  - Continuous Integration/Delivery/Deployment

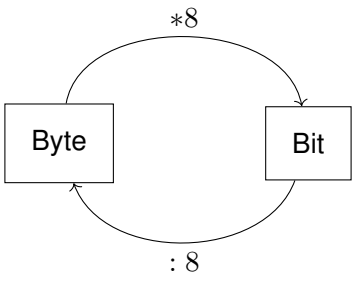
6.15 Webentwicklung

- Web 2.0
  - Social Networks, Wikis, Blogs, Twitter, Forum, Podcast
- Web 3.0
- Angriffsmöglichkeiten gegen Anwendungen abgrenzen
  - SQL-Injection, Session Hijacking, DoS, DDoS

7 Sonstiges

7.1 Umrechung von Datengrößen





7.2 Berechnung von Bildgrößen

Farbtiefe, RGB, Pixel, DPI, etc.

DPI Dots per Inch (Pixel pro Inch)

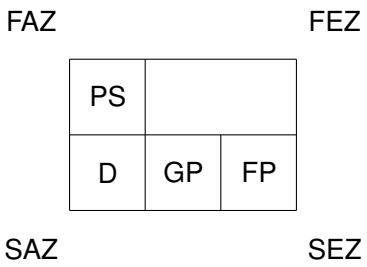
7.3 Netzplan

Quelle: MODU learn [25]

Die Basis für den Netzplan ist ein Prozess, der aus insgesamt sieben Teilschritten besteht. Zu jedem Schritt sind sowohl die Dauer als auch die vorher abzuschließenden Aufgaben bekannt:

Prozessschritt	Dauer in Stunden	Vorher zu beenden
A	2	-
B	4	A
C	3	B
D	2	B
E	1	C,D
F	4	C
G	5	E,F

Diese Schritte müssen gleich in Form von sogenannten Knoten dargestellt und miteinander verbunden werden. Daher sollten wir uns als zusätzliche Vorbereitung den grundlegenden Aufbau eines Knotens anschauen. Er weist die folgende Struktur auf:

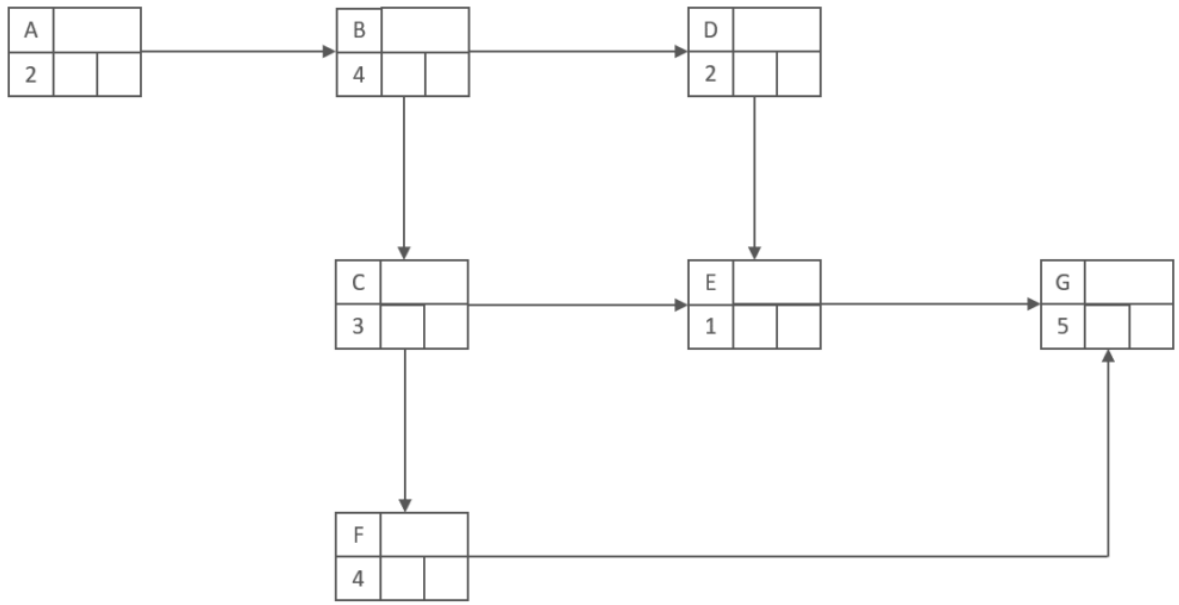


PS = Prozessschritt (in unserem Fall A bis F)
D = Dauer des jeweiligen Vorgangs
FAZ = Frühester Anfangszeitpunkt, zu dem der Prozessschritt begonnen werden kann
FEZ = Frühester Endzeitpunkt, zu dem der Prozessschritt abgeschlossen werden kann
SAZ = Spätester Anfangszeitpunkt, um den Gesamtprozess planmäßig beenden zu können
SEZ = Spätester Endzeitpunkt, zu dem ein Schritt abgeschlossen sein muss, um den geplanten Abschlusstermin nicht zu gefährden

GP = Gesamtpuffer, der genutzt werden kann, bevor der pünktliche Abschluss des Gesamtprozesses gefährdet wird
FP = Freier Puffer, der zur Verfügung steht, bevor der unmittelbar folgende Prozessschritt beeinflusst wird

7.3.1 Schritt 1: Knoten verknüpfen

Zuerst erstellst Du einen vorläufigen Netzplan, in dem einerseits die Prozessschritte und ihre Abhängigkeiten abgebildet sind und andererseits die jeweilige Dauer der Knoten. Das sieht für unseren konkreten Fall dann so aus:



7.3.2 Schritt 2: Vorwärtsterminierung

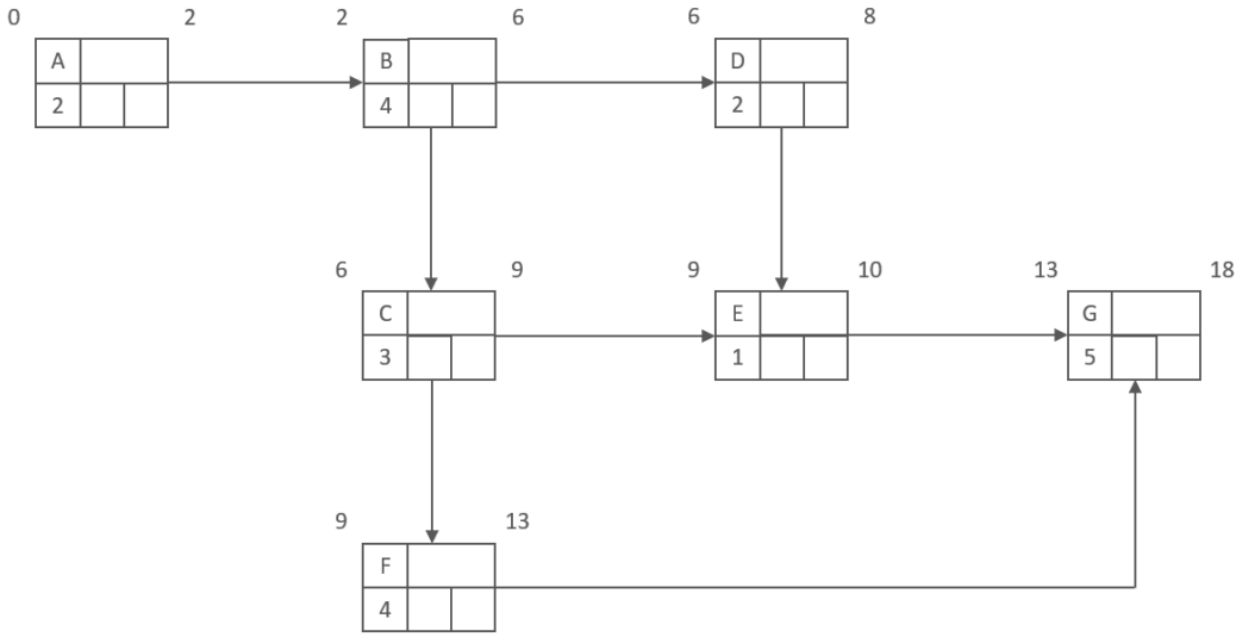
Bei der sogenannten Vorwärtsterminierung wanderst du alle Pfade vom Anfang bis zum Ende, also in unserem Fall von Schritt A zu Schritt G. Dabei trägst du jeweils den frühesten Anfangszeitpunkt (FAZ, oben links) und den frühesten Endzeitpunkt (FEZ, oben rechts) für jeden Prozessschritt in das Schema ein.

Um die jeweiligen Zeitpunkte zu berechnen, musst du folgendes beachten:

- Der FAZ des ersten Prozessschrittes (A) ist immer gleich 0.
- Der FEZ eines Vorgangs ergibt sich immer aus der Summe von FAZ und Dauer. Bei Schritt A wären das beispielsweise $0 + 2 = 2$.

- Der FEZ eines Vorgangs ist gleichzeitig der FAZ des nachfolgenden Prozessschrittes bzw. der nachfolgenden Prozessschritte. Dieses Übertragen der Werte kannst du auf der folgenden Grafik an allen Knoten erkennen.
- Hat ein Knoten mehrere Vorgänger (z. B. der Knoten E) wird derjenige Vorgänger-FEZ genommen, der den höchsten Wert aufweist (hier also die 9 von Schritt C).

Nach diesen Rechenregeln erhalten wir den Zwischenstand, den du in der nächsten Grafik siehst:

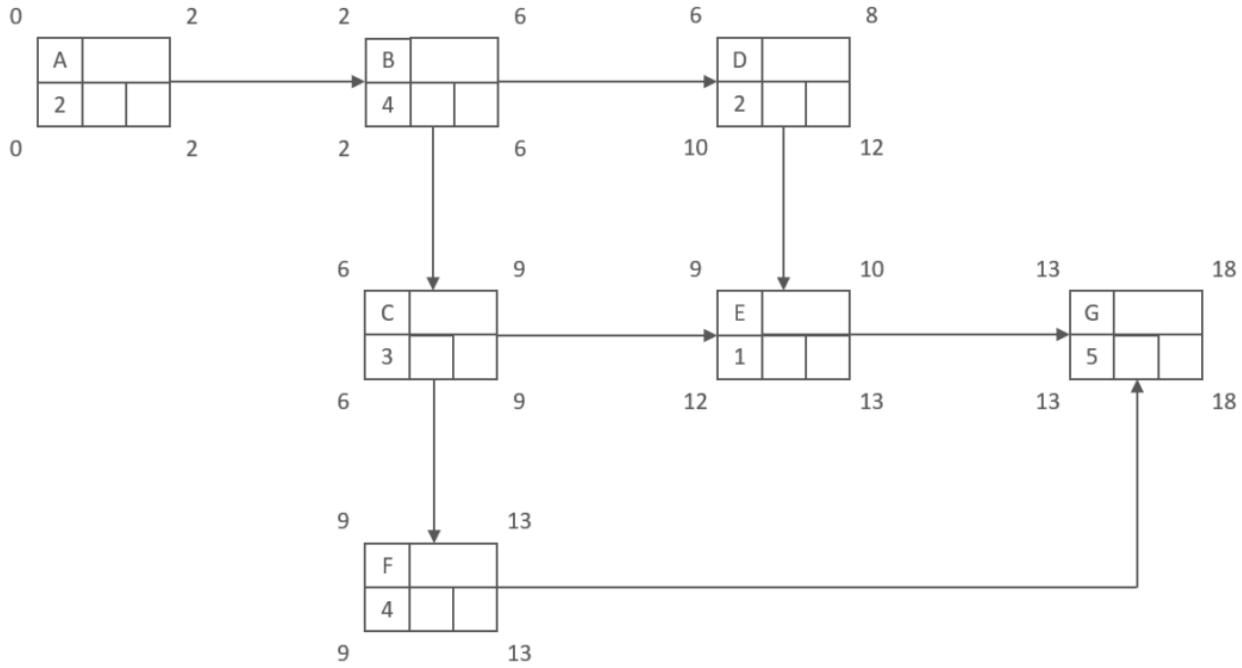


7.3.3 Schritt 3: Rückwärtsterminierung

Anschließend werden die Pfade nochmal rückwärts durchlaufen, um jeweils den spätesten Anfangszeitpunkt (SAZ, unten links) und den spätesten Endzeitpunkt (SEZ, unten rechts) zu ergänzen. Auch dabei sind einige Regeln zu beachten.

- Der SEZ des letzten Prozessschrittes, in unserem Fall also G, entspricht immer seinem FEZ. Er ist der Ausgangspunkt für die Rückwärtsterminierung.
- Der SAZ ergibt sich immer aus der Differenz von SEZ und Dauer des Vorgangs. Er zeigt an, wann ein Prozessschritt spätestens begonnen werden muss.
- Der SAZ eines Schrittes (z. B. 13 für Schritt G) ist identisch zum SEZ des vorherigen Schrittes (z. B. ebenfalls 13 für Schritt F und Schritt E). Er muss also nur korrekt übertragen werden.
- Hat ein Prozessschritt mehrere Nachfolger (z. B. E und F, die auf C folgen), wird als SEZ der jeweils kleinste Wert der möglichen SAZ übernommen (hier also 9 von Schritt F und nicht 12 von Schritt E).

Nach diesem Schema arbeitest du dich nun Stück für Stück zum ersten Teilschritt A vor. Nach diesem Verfahren ergibt sich in unserem Beispiel folgender, vorläufiger Netzplan:

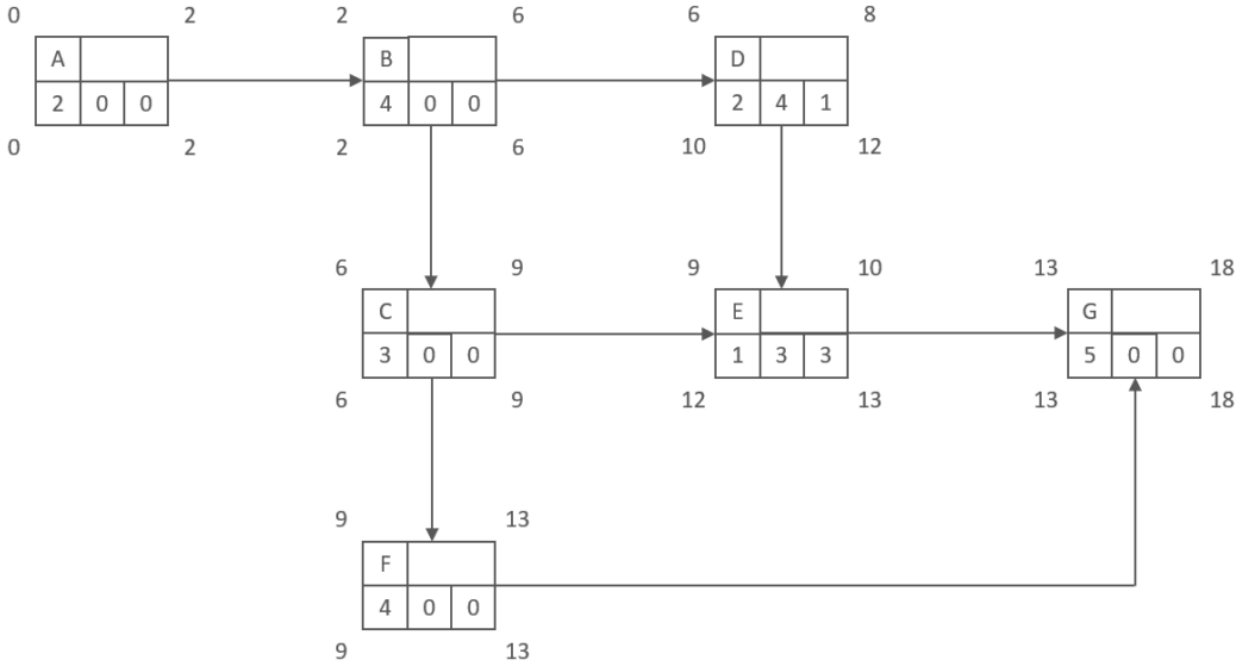


7.3.4 Schritt 4: Pufferzeiten berechnen

Nun kannst du für jeden Teilschritt die jeweiligen Puffer berechnen:

- Für den Gesamtpuffer GP ermittelst du dafür die Differenz aus SAZ und FAZ. Er zeigt dir an, wie viel Verzögerung akzeptabel ist, bevor der **pünktliche Abschluss des Gesamtprozesses** gefährdet wird.
- Für den freien Puffer eines Prozessschritts (FP) berechnest du jeweils die Differenz aus dem FAZ des nachfolgenden Schritts und dem eigenen FEZ. Sollte es mehrere Nachfolger geben, wird stets die kleinste Alternative der FAZ genommen. Hiermit wird abgebildet, wie viel Puffer vorhanden ist, bevor **ein unmittelbar folgender Prozessschritt beeinflusst**

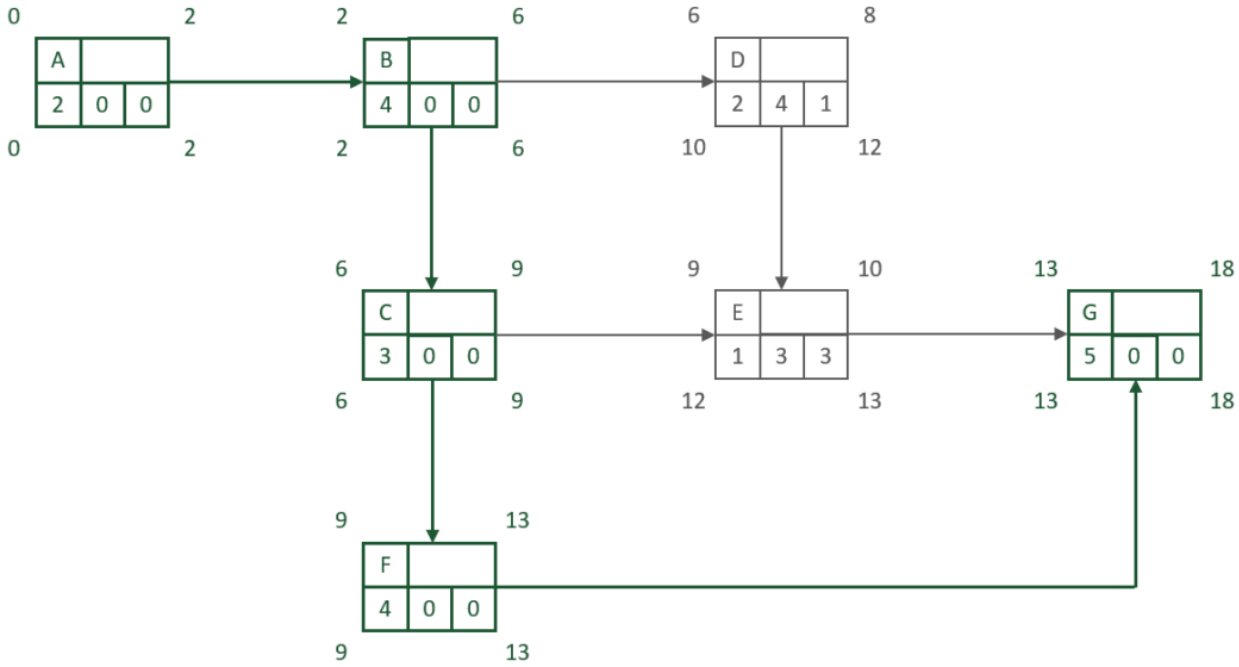
Anschließend ergibt sich daraus der vollständige Netzplan:



7.3.5 Schritt 5: Kritischen Pfad ermitteln

Abschließend kannst du den kritischen Pfad des gesamten Prozesses ermitteln; also alle Teilschritte, in denen sich keine Verzögerung ergeben darf, wenn der ursprüngliche Termin eingehalten werden soll.

Dazu gehören **alle Prozessschritte, die weder einen freien Puffer noch einen Gesamtpuffer aufweisen**. Im Beispiel ist das der Pfad ABCFG, der in der nächsten Grafik farblich hervorgehoben ist.



Quellen

[1] Amazon. Was ist SLA (Service Level Agreement)? https://aws.amazon.com/de/what-is/service-level-agreement/, 2023.

[2] Amazon. Was ist NoSQL? https://aws.amazon.com/de/nosql/, 2024.

[3] Atlassian.com. Git kennenlernen: Git-Befehle. https://www.atlassian.com/de/git/glossary#commands, 2024.

[4] b-quadrat, Carsten. Was ist der Unterschied zwischen DSGVO und BDSG? https://b-quadrat.de/was-ist-der-unterschied-zwischen-dsgvo-und-bdsg/, September 2023.

[5] BSI. Schutzbedarfskategorien. https://www.bsi.bund.de/dok/10990084, 2024.

[6] Bundesbeauftragte für den Datenschutz und die Informationsfreiheit. Meine Rechte - die Betroffenenrechte. https://www.bfdi.bund.de/DE/Buerger/Basiswissen/Betroffenenrechte/BetroffenenRechte_node.html, 2024.

[7] cisco. Was ist eine Firewall? https://www.cisco.com/c/de_de/products/security/firewalls/what-is-a-firewall.html, 2024.

[8] Database camp. Normalisierung. https://databasecamp.de/daten/normalisierung, May 2023.

[9] Database camp. Anomalien in Datenbanken. https://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/anomalien-datenbank/, 2024.

[10] Datenbanken-verstehen.de. Entity Relationship Diagram. https://www.datenbanken-verstehen.de/lexikon/entity-relationship-diagram/, 2024.

[11] der-prozessmanager, Michael Durst, Sascha Hertkorn, Christopher Eischer, Nico Schweisser. Was ist ein PDCA-Zyklus? Plan-Do-Check-Act einfach erklärt. https://der-prozessmanager.de/aktuell/wissensdatenbank/pdca-zyklus, 2021.

[12] dr-datenschutz, intersoft-consulting-services-ag. Was schützt das allgemeine Persönlichkeitsrecht? https://www.dr-datenschutz.de/was-schuetzt-das-allgemeine-persoenlichkeitsrecht/, 2024.

[13] Enginsight, Max Tarantik. Vertraulichkeit, Integrität, Verfügbarkeit – Einfach erklärt! https://enginsight.com/de/blog/vertraulichkeit-integritaet-verfuegbarkeit-einfach-erklaert/, January 2022.

[14] Eric Amberg, Daniel Schmid. *Hacking; Der umfassende Praxis-Guide Inkl. Prüfungsvorbereitung zum CEHv11*. mitp, Frechen, 2. auflage edition, 2022.

[15] GeeksforGeeks. SQL | DDL, DQL, DML, DCL and TCL Commands. https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/, October 2023.

[16] hagel-it, Sebastian Hansen. WAS IST EINE USV ? https://www.hagel-it.de/it-service/was-ist-eine-usv.html, 2024.

[17] hs-augsburg, Kowa. Mengenoperatoren in SQL. https://glossar.hs-augsburg.de/Mengenoperatoren_in_SQL, July 2012.

[18] IHK Rhein-Neckar. Die Grundsätze der Datenverarbeitung. https://www.ihk.de/rhein-neckar/recht/datenschutz-it-sicherheit/grundsaetze-datenverarbeitung-4554126, 2024.

[19] Industrie und Handelskammer. Belegsatz. https://cloud.agb.schule/apps/files/?dir=/Klassen/FIA101/Tauschen/3.%20Jahr/02%20-%20Pr%C3%BCfungsvorbereitung/03%20-%20Sommer%202023&fileid=10240654, February 2024.

[20] Ionos. Das Use-Case-Diagramm (Anwendungsfalldiagramm) in UML. https://www.ionos.de/digitalguide/websites/web-entwicklung/anwendungsfalldiagramm/, March 2020.

[21] Kaspersky. Was ist Spoofing? https://www.kaspersky.de/resource-center/definitions/ip-and-email-spoofing, 2024.

[22] kaspersky. Was ist SQL-Injection? Definition und Erläuterung. https://www.kaspersky.de/resource-center/definitions/sql-injection, 2024.

[23] Lehrerfortbildung-bw.de. Struktogramme. https://lehrerfortbildung-bw.de/u_matnatech/informatik/gym/bp2016/fb1/2_algorithmen/1_hintergrund/2_hintergrund/6_struktogramm/, April 2024.

[24] Microsoft, Zoiner Tejada. Nicht relationale Daten und NoSQL. https://learn.microsoft.com/de-de/azure/architecture/data-guide/big-data/non-relational-data, 2024.

[25] MODU-learn. Netzplantechnik: Eine Schritt-für-Schritt-Anleitung mit Beispiel. https://www.modu-learn.de/verstehen/management/netzplantechnik/, 2024.

[26] oose.de. Notationsübersicht UML 2.5. https://www.oose.de/wp-content/uploads/2012/05/UML-Notations%C3%BCbersicht-2.5.pdf, 2013.

[27] Prashanth Jayaram, sqlshack.com. CRUD operations in SQL Server. https://www.sqlshack.com/crud-operations-in-sql-server/, December 2018.

[28] processmaps.com, Stefan Kempter. Incident Management. https://wiki.de.it-processmaps.com/index.php/Incident_Management, 2024.

[29] redbots.de, Thomas Klein, Elena Semenova. Tests in der Softwareentwicklung: Ein Klassifizierungsansatz. https://www.redbots.de/blog/software-tests-klassifizierung/, 2021.

[30] Tino Hempel. Selektion, Projektion und Join in PROLOG. https://www.tinohempel.de/info/info/datenbanken_prolog/abfragen_II.htm, 2006.