



EUROPA FACHBUCHREIHE  
für IT-Berufe

# **Fachinformatiker/-in Anwendungsentwicklung**

**Lehrbuch mit Aufgaben und Lernsituationen,  
mit C#, Java, Python und UML**

1. Auflage

Verlag Europa-Lehrmittel · Nourney, Vollmer GmbH & Co. KG  
Düsseldorfer Straße 23 · 42781 Haan-Gruiten

**Europa-Nr.: 33679**

**Autor:**

Dirk Hardy, Oberhausen

**Verlagslektorat:**

Andreas Nies, Dipl.-Ing. Elektrotechnik, Aachen

**Bildquellenverzeichnis:**

Alle Bilder im Buch ohne Quellenangaben wurden von der Setzerei Typework oder vom Autor erstellt und bearbeitet.

Die in diesem Lehr- und Übungsbuch genannten Software-, Hardware- und Handelsnamen sind in der Mehrzahl auch eingetragene Warenzeichen.

Unter Verwendung von Screenshots aus:

Apache NetBeans 26 IDE, © The Apache Software Foundation

Apache Tomcat, © The Apache Software Foundation

Google Stitch/ Google Labs <https://stitch.withgoogle.com/>

Microsoft Copilot <https://copilot.microsoft.com/>

Python.org, (© Python Software Foundation PSF)

StarUML (© MKLab Co., Ltd.)

Visual Studio Community Edition 2022, © Microsoft

XAMPP (© The Apache Software Foundation)

Adobe Systems Software Irland Ltd., Adobe Stock, IRL-Dublin: 15/1, © Dewa Artwork; 15/2, © Looker\_Studio; 18/1, 19/1, © Lumos Maxima; 21/1, © Razulation; 21/3, © microstock77; 449/1 © C.M., peopleimages.com; 450/1, © Kostiantyn; 454/1, © Elnur

wikipedia.org, Berlin: 14/2, © Kephir; 22/2, © Qhx

1. Auflage 2026

Druck 5 4 3 2 1

Alle Drucke derselben Auflage sind parallel einsetzbar, da sie bis auf die Korrektur von Druckfehlern identisch sind.

ISBN 978-3-7585-3367-9

Bei Fragen zur Produktsicherheit wenden Sie sich bitte an [produktsicherheit@europa-lehrmittel.de](mailto:produktsicherheit@europa-lehrmittel.de).

Alle Rechte vorbehalten. Das Werk ist urheberrechtlich geschützt. Jede Verwertung außerhalb der gesetzlich geregelten Fälle muss vom Verlag schriftlich genehmigt werden.

© 2026 by Verlag Europa-Lehrmittel, Nourney, Vollmer GmbH & Co. KG, 42781 Haan-Gruiten

[www.europa-lehrmittel.de](http://www.europa-lehrmittel.de)

Umschlagfotos: Christian Worrington, imageteam, bildbox, issaronow – alle adobestock.com

Umschlag: braunwerbeagentur, 42477 Radevormwald

Satz: Typework Layoutsatz & Grafik GmbH, 86153 Augsburg

Druck: UAB BALTO print, 08217 Vilnius (LT))

## Inhaltsverzeichnis

|          |                                       |          |
|----------|---------------------------------------|----------|
| <b>0</b> | <b>Aufbau und Nutzung des Buches</b>  | <b>9</b> |
| 0.1      | Aufbau des Buches                     | 9        |
| 0.2      | Unterrichten der einzelnen Lernfelder | 11       |
| 0.3      | Hinweise zu digitalen Zusatzbüchern   | 12       |

## Lernfeld 10a

### Benutzerschnittstellen gestalten und entwickeln

|          |                                                          |           |
|----------|----------------------------------------------------------|-----------|
| <b>1</b> | <b>Benutzerschnittstellen (UI) kennenlernen</b>          | <b>14</b> |
| 1.1      | Schnittstellen allgemein (Interfaces)                    | 14        |
| 1.2      | Grundlagen von Benutzerschnittstellen                    | 14        |
| 1.3      | Ergonomie von Benutzerschnittstellen                     | 15        |
| 1.4      | User Experience (UX)                                     | 16        |
| 1.5      | User Centered Design (UCD)                               | 17        |
| 1.6      | Aufgaben zu Benutzerschnittstellen (UI)                  | 20        |
| <b>2</b> | <b>Grafische Benutzerschnittstellen (GUI) entwickeln</b> | <b>21</b> |
| 2.1      | GUI-Entwicklung mit Windows-Forms                        | 24        |
| 2.1.1    | Das erste Windows-Forms-Programm                         | 25        |
| 2.1.2    | Windows-Forms-Designer                                   | 37        |
| 2.1.3    | Einfache Steuerelemente verwenden                        | 41        |
| 2.1.4    | Standard-Dialogfelder                                    | 49        |
| 2.1.5    | Die Baumansicht (TreeView)                               | 53        |
| 2.1.6    | Die Listenansicht (ListView)                             | 56        |
| 2.1.7    | Menüs erstellen                                          | 60        |
| 2.1.8    | Neue Formulare hinzufügen                                | 61        |
| 2.2      | GUI-Entwicklung mit WPF                                  | 63        |
| 2.2.1    | Windows-Programmierung mit WPF                           | 63        |
| 2.2.2    | Das erste WPF-Programm                                   | 64        |
| 2.2.3    | Grundkonzepte der WPF                                    | 69        |
| 2.2.4    | XAML                                                     | 83        |
| 2.2.5    | Der WPF-Designer                                         | 84        |
| 2.2.6    | Einfache und komplexe Steuerelemente                     | 90        |
| 2.2.7    | Weitere WPF-Konzepte                                     | 98        |
| 2.3      | GUI-Entwicklung mit .NET MAUI-Apps                       | 105       |
| 2.3.1    | Multi-platform App User Interface                        | 105       |
| 2.3.2    | Die erste .NET MAUI App                                  | 106       |
| 2.3.3    | Steuerelemente und visuelle Hierarchie                   | 111       |

|       |                                               |            |
|-------|-----------------------------------------------|------------|
| 2.3.4 | Datenbindung und MVVM .....                   | 119        |
| 2.4   | Einsatz von KI in der GUI-Entwicklung .....   | 127        |
| 2.5   | Aufgaben zur GUI-Entwicklung .....            | 130        |
| 3     | <b>Lernsituationen zu Lernfeld 10a.</b> ..... | <b>136</b> |

## Lernfeld 11a

|                                                        |            |
|--------------------------------------------------------|------------|
| <b>Funktionalität in Anwendungen realisieren</b> ..... | <b>141</b> |
|--------------------------------------------------------|------------|

|          |                                                                              |            |
|----------|------------------------------------------------------------------------------|------------|
| <b>4</b> | <b>Objektorientierte Programmierung (OOP) vertiefen</b> .....                | <b>142</b> |
| 4.1      | .NET und C# .....                                                            | 142        |
| 4.1.1    | Das .NET-Framework und .NET .....                                            | 142        |
| 4.1.2    | Die Sprache C# .....                                                         | 144        |
| 4.2      | Das Klassenkonzept in C# .....                                               | 153        |
| 4.2.1    | Die erste Klasse in C# .....                                                 | 154        |
| 4.2.2    | Methoden in C# .....                                                         | 157        |
| 4.2.3    | Weitere Elemente von Klassen .....                                           | 169        |
| 4.2.4    | Strukturen in C# .....                                                       | 176        |
| 4.3      | Vererbung in C# .....                                                        | 178        |
| 4.3.1    | Die Vererbung in C# .....                                                    | 178        |
| 4.3.2    | Polymorphismus .....                                                         | 182        |
| 4.3.3    | Abstrakte Basisklassen .....                                                 | 187        |
| 4.3.4    | Interfaces in C# .....                                                       | 190        |
| 4.4      | Überladen von Operatoren .....                                               | 194        |
| 4.4.1    | Operator-Methoden in C# .....                                                | 195        |
| 4.4.2    | Konvertierungsoperatoren überladen .....                                     | 198        |
| 4.5      | Arrays in C# .....                                                           | 201        |
| 4.5.1    | Ein- und mehrdimensionale Arrays .....                                       | 202        |
| 4.5.2    | Sortieren von Arrays .....                                                   | 211        |
| 4.5.3    | Besondere Array-Klassen .....                                                | 218        |
| 4.6      | Einsatz von KI zur Codegenerierung .....                                     | 222        |
| 4.7      | Aufgaben zur objektorientierten Programmierung mit C# .....                  | 228        |
| <b>5</b> | <b>Objektorientierte Analyse und objektorientiertes Design mit UML</b> ..... | <b>233</b> |
| 5.1      | Die Unified Modeling Language (UML) .....                                    | 233        |
| 5.2      | Das Anwendungsfalldiagramm .....                                             | 235        |
| 5.2.1    | Anwendungsfälle und Akteure .....                                            | 235        |
| 5.2.2    | Beziehungen zwischen Anwendungsfall und Akteur .....                         | 237        |
| 5.2.3    | Beziehungen zwischen Anwendungsfällen .....                                  | 238        |
| 5.2.4    | Beziehungen zwischen Akteuren .....                                          | 241        |
| 5.3      | Das Klassendiagramm .....                                                    | 243        |

|          |                                                                   |            |
|----------|-------------------------------------------------------------------|------------|
| 5.3.1    | Die Darstellung der Klasse . . . . .                              | 243        |
| 5.3.2    | Beziehungen zwischen Klassen . . . . .                            | 248        |
| 5.3.3    | Die Assoziation . . . . .                                         | 249        |
| 5.3.4    | Umsetzung von Assoziationen . . . . .                             | 252        |
| 5.3.5    | Die Aggregation . . . . .                                         | 256        |
| 5.3.6    | Die Komposition . . . . .                                         | 261        |
| 5.3.7    | Die Generalisierung und Spezialisierung . . . . .                 | 264        |
| 5.3.8    | Stereotype. . . . .                                               | 268        |
| 5.4      | Das Objektdiagramm . . . . .                                      | 273        |
| 5.4.1    | Die Darstellung eines Objektes . . . . .                          | 273        |
| 5.4.2    | Beziehungen zwischen Objekten . . . . .                           | 273        |
| 5.4.3    | Umsetzung eines Objektdiagramms . . . . .                         | 275        |
| 5.5      | Das Sequenzdiagramm . . . . .                                     | 278        |
| 5.5.1    | Allgemeine Darstellung . . . . .                                  | 278        |
| 5.5.2    | Fragmente. . . . .                                                | 281        |
| 5.5.3    | Umsetzung eines Sequenzdiagramms . . . . .                        | 283        |
| 5.6      | Das Aktivitätsdiagramm . . . . .                                  | 286        |
| 5.6.1    | Allgemeine Darstellung . . . . .                                  | 286        |
| 5.6.2    | Besondere Kommunikation . . . . .                                 | 291        |
| 5.6.3    | Selektion und Iteration . . . . .                                 | 296        |
| 5.6.4    | Umsetzung eines Aktivitätsdiagramms. . . . .                      | 298        |
| 5.7      | Aufgaben zu UML-Diagrammen mit C# . . . . .                       | 303        |
| <b>6</b> | <b>SQL anwenden und vertiefen. . . . .</b>                        | <b>306</b> |
| 6.1      | Grundlagen der Sprache SQL . . . . .                              | 306        |
| 6.1.1    | Entwicklung relationaler Datenbanken und der Sprache SQL. . . . . | 306        |
| 6.1.2    | Einordnung der Sprache SQL. . . . .                               | 307        |
| 6.1.3    | Relationenalgebra als Basis von SQL . . . . .                     | 308        |
| 6.1.4    | Sprachbestandteile von SQL . . . . .                              | 310        |
| 6.1.5    | Grundbegriffe des relationalen Datenmodells . . . . .             | 310        |
| 6.1.6    | Konzeption einer relationalen Datenbank . . . . .                 | 312        |
| 6.1.7    | Implementieren der Beispieldatenbank . . . . .                    | 315        |
| 6.2      | Daten abfragen mit SELECT . . . . .                               | 318        |
| 6.2.1    | Aufbau der SELECT-Anweisung . . . . .                             | 319        |
| 6.2.2    | Datentypen und Operatoren . . . . .                               | 321        |
| 6.2.3    | Bedingungen mit der WHERE-Klausel formulieren . . . . .           | 323        |
| 6.2.4    | Sortieren mit ORDER BY . . . . .                                  | 327        |
| 6.3      | Funktionen nutzen . . . . .                                       | 330        |
| 6.3.1    | Numerische Funktionen . . . . .                                   | 331        |
| 6.3.2    | Zeichenkettenfunktionen. . . . .                                  | 334        |

|        |                                                         |     |
|--------|---------------------------------------------------------|-----|
| 6.3.3  | Datumsfunktionen. . . . .                               | 339 |
| 6.3.4  | Konvertierungsfunktionen . . . . .                      | 342 |
| 6.3.5  | Konditionale Funktionen und Ausdrücke . . . . .         | 343 |
| 6.4    | Daten gruppieren . . . . .                              | 346 |
| 6.4.1  | Daten gruppieren mit GROUP BY . . . . .                 | 347 |
| 6.4.2  | Aggregatfunktionen nutzen. . . . .                      | 348 |
| 6.4.3  | Daten einschränken mit HAVING. . . . .                  | 350 |
| 6.5    | Daten zusammenführen mit JOIN. . . . .                  | 351 |
| 6.5.1  | Das Kreuzprodukt als JOIN-Grundlage. . . . .            | 352 |
| 6.5.2  | Einfache JOINS . . . . .                                | 352 |
| 6.5.3  | Komplexe JOINS . . . . .                                | 354 |
| 6.5.4  | INNER-JOINS und OUTER-JOINS . . . . .                   | 356 |
| 6.5.5  | Spezielle JOINS . . . . .                               | 357 |
| 6.6    | Unterabfragen einsetzen. . . . .                        | 358 |
| 6.6.1  | Ein-Wert-Unterabfragen . . . . .                        | 359 |
| 6.6.2  | Mehrere-Werte-Unterabfragen. . . . .                    | 359 |
| 6.6.3  | Korrelierte Unterabfragen . . . . .                     | 361 |
| 6.6.4  | Unterabfragen als Ausdruck. . . . .                     | 361 |
| 6.7    | Daten manipulieren . . . . .                            | 363 |
| 6.7.1  | Daten einfügen mit INSERT. . . . .                      | 363 |
| 6.7.2  | Daten ändern mit UPDATE . . . . .                       | 365 |
| 6.7.3  | Daten löschen mit DELETE. . . . .                       | 367 |
| 6.7.4  | Transaktionen mit COMMIT und ROLLBACK . . . . .         | 368 |
| 6.8    | Tabellen erstellen . . . . .                            | 371 |
| 6.8.1  | SQL-Datentypen . . . . .                                | 371 |
| 6.8.2  | Tabellen mit CREATE TABLE erstellen . . . . .           | 373 |
| 6.8.3  | Regeln erstellen. . . . .                               | 376 |
| 6.8.4  | Eigene Regeln benennen und das Data-Dictionary. . . . . | 377 |
| 6.8.5  | Beziehungen zwischen Tabellen umsetzen. . . . .         | 379 |
| 6.8.6  | Tabellen ändern und löschen. . . . .                    | 382 |
| 6.9    | Views erstellen . . . . .                               | 385 |
| 6.9.1  | Einfache Views erstellen. . . . .                       | 387 |
| 6.9.2  | Komplexe Views erstellen . . . . .                      | 389 |
| 6.10   | Rechteverwaltung. . . . .                               | 390 |
| 6.10.1 | Benutzer und Datenbanken verwalten. . . . .             | 390 |
| 6.10.2 | Objektrechte, Systemrechte und Rollen. . . . .          | 391 |
| 6.10.3 | Rechte zuweisen mit GRANT . . . . .                     | 393 |
| 6.10.4 | Rechte entziehen mit REVOKE. . . . .                    | 395 |
| 6.11   | Aufgaben zu SQL anwenden und vertiefen . . . . .        | 396 |

|          |                                                   |            |
|----------|---------------------------------------------------|------------|
| <b>7</b> | <b>Datenbankanbindung umsetzen</b>                | <b>400</b> |
| 7.1      | Datenbankzugriff                                  | 400        |
| 7.1.1    | Datenbankanbindung unter dem .NET-Framework       | 400        |
| 7.1.2    | Provider nutzen und eine Verbindung aufbauen      | 401        |
| 7.1.3    | Beispiel eines Zugriffs auf eine ACCESS-Datenbank | 401        |
| 7.1.4    | Nicht-Select-Befehle absetzen                     | 404        |
| 7.1.5    | DataAdapter und DataSet                           | 406        |
| 7.2      | Den Datenbankassistenten von Visual C# nutzen     | 409        |
| 7.2.1    | Eine Datenbank einbinden.                         | 409        |
| 7.2.2    | Windows-Forms-Steuerelemente automatisch anbinden | 413        |
| 7.2.3    | WPF-Steuerelemente automatisch anbinden           | 415        |
| 7.3      | Aufgaben zum Datenbankzugriff mit ADO.NET         | 420        |
| <b>8</b> | <b>Lernsituationen zu Lernfeld 11a.</b>           | <b>422</b> |

## **Lernfeld 12a**

|                                                            |            |
|------------------------------------------------------------|------------|
| <b>Kundenspezifische Anwendungsentwicklung durchführen</b> | <b>429</b> |
|------------------------------------------------------------|------------|

|          |                                          |            |
|----------|------------------------------------------|------------|
| <b>9</b> | <b>Projektmanagement</b>                 | <b>430</b> |
| 9.1      | Projekte                                 | 430        |
| 9.2      | Projektmanagement allgemein              | 430        |
| 9.3      | Vorgehensmodelle Softwareentwicklung.    | 432        |
| 9.4      | Dokumente im Softwareentwicklungsprozess | 433        |
| 9.4.1    | Fachkonzept                              | 434        |
| 9.4.2    | Lastenheft                               | 434        |
| 9.4.3    | Pflichtenheft                            | 435        |
| 9.4.4    | DV-Konzept.                              | 437        |
| 9.4.5    | Testkonzept und Testprotokoll.           | 439        |
| 9.4.6    | Quellcode-Dokumentation                  | 441        |
| 9.4.7    | Kundendokumentation                      | 442        |
| 9.4.8    | Wartungsdokumentation.                   | 443        |
| 9.5      | Wasserfallmodell                         | 445        |
| 9.6      | V-Modell.                                | 446        |
| 9.7      | Spiralmodell                             | 448        |
| 9.8      | Scrum.                                   | 449        |
| 9.9      | Extreme Programming XP.                  | 452        |
| 9.10     | Kanban                                   | 453        |
| 9.11     | Einsatz von KI im Projektmanagement      | 455        |
| 9.12     | Aufgaben zum Projektmanagement           | 459        |

|                     |                                                                |            |
|---------------------|----------------------------------------------------------------|------------|
| <b>10</b>           | <b>Softwarearchitektur</b>                                     | <b>461</b> |
| 10.1                | Software-Architekturmuster                                     | 462        |
| 10.1.1              | Schichtenarchitektur                                           | 462        |
| 10.1.2              | Pipes & Filters                                                | 464        |
| 10.1.3              | Microservices-Architektur                                      | 465        |
| 10.1.4              | Model-View-Controller-Architektur                              | 466        |
| 10.1.5              | Vor- und Nachteile der vorgestellten Architekturmuster         | 467        |
| 10.2                | Entwurfsmuster                                                 | 468        |
| 10.2.1              | Fabrik-Entwurfsmuster mit Spezialfall Singleton-Entwurfsmuster | 468        |
| 10.2.2              | Kompositum-Entwurfsmuster                                      | 469        |
| 10.2.3              | Beobachter-Entwurfsmuster                                      | 470        |
| 10.3                | Aufgaben zu Softwarearchitektur und Entwurfsmustern            | 472        |
| <b>11</b>           | <b>Qualitätsmanagement</b>                                     | <b>475</b> |
| 11.1                | Softwarequalität                                               | 475        |
| 11.2                | Testmethoden                                                   | 476        |
| 11.2.1              | Black-Box-Tests                                                | 476        |
| 11.2.2              | White-Box-Tests                                                | 477        |
| 11.2.3              | Weitere Tests                                                  | 481        |
| 11.2.4              | Debugging                                                      | 482        |
| 11.3                | Automatische Testgenerierung mit KI                            | 483        |
| 11.4                | Datenschutzaspekte                                             | 488        |
| 11.5                | Aufgaben zum Qualitätsmanagement                               | 490        |
| <b>12</b>           | <b>Lernsituationen zu Lernfeld 12a</b>                         | <b>492</b> |
| Anhang              |                                                                | 497        |
| Sachwortverzeichnis |                                                                | 501        |

## 0 Aufbau und Nutzung des Buches

### 0.1 Aufbau des Buches

Die Inhalte des Buches orientieren sich an den Lernfeldern 10a, 11a und 12a des Rahmenlehrplans für die IT-Berufe (speziell Fachinformatiker-Anwendungsentwicklung). Die Lernfelder<sup>1</sup> sind hier dargestellt. Der darauffolgende Abschnitt leitet die konkreten Inhalte dieses Buches aus den Beschreibungen der Lernfelder ab.

#### Lernfeld 10a: Benutzerschnittstellen gestalten und entwickeln

**Die Schülerinnen und Schüler verfügen über die Kompetenz, Benutzeroberflächen für softwarebasierte Arbeitsabläufe und Geschäftsprozesse zu gestalten und zu entwickeln.**

- Die Schülerinnen und Schüler **informieren** sich über die vorhandenen betrieblichen Abläufe und Geschäftsprozesse.
- Sie **stellen** diese modellhaft **dar** und leiten Optimierungsmöglichkeiten ab.
- Sie **gestalten** und **entwickeln** mit agilen Methoden die Benutzeroberflächen für unterschiedliche Endgeräte und Betriebssysteme und stellen die vollständige Abbildung des Informationsflusses unter Berücksichtigung der Prozessbeschreibung sicher.
- Die Schülerinnen und Schüler **stellen** die Funktionalität der Softwarelösung **her** und nutzen hierzu bereits vorhandene Bibliotheken und Module.
- Sie **überprüfen** das Produkt auf Datenschutzkonformität und Benutzerfreundlichkeit.
- Die Schülerinnen und Schüler **testen** die funktionale Richtigkeit. Sie quantifizieren die Reduktion der Prozesskosten des digitalisierten, optimierten Geschäftsprozesses und stellen diese den Entwicklungskosten gegenüber.

#### Lernfeld 11a: Funktionalität in Anwendungen realisieren

**Die Schülerinnen und Schüler verfügen über die Kompetenz, modulare Komponenten zur informationstechnischen Verarbeitung von Arbeitsabläufen und Geschäftsprozessen zu entwickeln und deren Qualität zu sichern.**

- Die Schülerinnen und Schüler **leiten** aus den Informationsobjekten der vorgegebenen Prozessbeschreibungen der Kunden die dazu notwendigen Datenstrukturen und Funktionalitäten **ab**.
- Sie **planen** modulare Softwarekomponenten und beschreiben deren Funktionsweise mit Diagrammen und Modellen.
- Sie **wählen** eine Methode zur Softwareentwicklung aus. Dabei beachten sie, dass Planung, Realisierung und Tests iterativ in Abstimmung mit den Kunden erfolgen.
- Die Schülerinnen und Schüler **realisieren**, auch im Team, die Softwarekomponenten und binden diese an Datenquellen an. Sie dokumentieren die Schnittstellen.
- Sie testen die erforderliche Funktionalität, indem sie Testfälle formulieren und automatisierte Testverfahren anwenden.
- Die Schülerinnen und Schüler **beurteilen** die Funktionalität anhand festgelegter Kriterien der Kunden und leiten Maßnahmen zur Überarbeitung der erstellten Module ein.

<sup>1</sup> **Quelle:** Rahmenlehrplan für die Ausbildungsberufe Fachinformatiker und Fachinformatikerin IT-System-Elektroniker und IT-System-Elektronikerin (Beschluss der Kultusministerkonferenz vom 13.12.2019)

Lernfeld 12a: Kundenspezifische Anwendungsentwicklung durchführen

Die Schülerinnen und Schüler verfügen über die Kompetenz, einen Kundenauftrag zur Anwendungsentwicklung vollständig durchzuführen und zu bewerten.

- Die Schülerinnen und Schüler **führen** in Zusammenarbeit mit den Kunden eine Anforderungsanalyse **durch** und leiten daraus Projektziele, Anforderungen, gewünschte Ergebnisse, Schulungsbedarfe und Rahmenbedingungen ab.
- Auf dieser Basis **planen** und kalkulieren sie ein Projekt mit den dazugehörigen personellen und technischen Ressourcen.
- Die Schülerinnen und Schüler entwickeln Lösungsvarianten, vergleichen diese anhand festgelegter Kriterien sowie unter Berücksichtigung von Datenschutz und Datensicherheit. Sie **wählen** mit den Kunden die beste Lösung **aus**. Für den vereinbarten Auftrag erstellen sie ein Dokument über die zu erbringenden Leistungen und ein Angebot.
- Die Schülerinnen und Schüler **implementieren** die gewünschte Lösung. Dabei nutzen sie Maßnahmen zur Qualitätssicherung. Sie präsentieren den Kunden das Projektergebnis und führen eine Schulung durch. Sie übergeben den Kunden das Produkt sowie die Dokumentation.
- Die Schülerinnen und Schüler **bewerten** das Projektergebnis auch hinsichtlich Zielerreichung, Wirtschaftlichkeit, Skalierbarkeit und Verlässlichkeit.
- Sie **reflektieren** die Projektdurchführung und das Projektergebnis auch unter Berücksichtigung der kritisch-konstruktiven Kundenrückmeldungen.

Basierend auf diesen Lernfeldern wurden entsprechende Fachkompetenzen identifiziert. Diese Fachkompetenzen können in *spezielle Fachkompetenzen* sowie *übergeordnete Fachkompetenzen* unterteilt werden.

Die Fachkompetenzen im Überblick:

| Spezielle Fachkompetenzen                                 |                                                                                                                  | Übergeordnete Fachkompetenzen |                                                                                                                        |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>UI</b><br>Kapitel 1                                    | <ul style="list-style-type: none"><li>• Grundlagen</li><li>• Ergonomie</li><li>• UX, UCD</li></ul>               | <b>UML</b><br>Kapitel 5       | <ul style="list-style-type: none"><li>• Objektorientiertes Design</li><li>• Objektorientierte Analyse</li></ul>        |
| <b>GUI</b><br>Kapitel 2                                   | <ul style="list-style-type: none"><li>• Forms, WPF</li><li>• .NET MAUI</li><li>• AWT, Swing, Tkinter</li></ul>   | <b>PM</b><br>Kapitel 9        | <ul style="list-style-type: none"><li>• Projektaspekte</li><li>• Softwareentwicklungsmodelle</li></ul>                 |
| <b>OOP</b><br>Kapitel 4                                   | <ul style="list-style-type: none"><li>• Klassenkonzept</li><li>• Vererbung</li><li>• Arrays</li></ul>            | <b>SWA</b><br>Kapitel 10      | <ul style="list-style-type: none"><li>• Architekturmuster</li><li>• Entwurfsmuster</li></ul>                           |
| <b>SQL</b><br>Kapitel 6                                   | <ul style="list-style-type: none"><li>• DML</li><li>• DDL</li><li>• DCL</li></ul>                                | <b>QM</b><br>Kapitel 11       | <ul style="list-style-type: none"><li>• Qualitätsaspekte</li><li>• Datenschutzaspekte</li><li>• Testmethoden</li></ul> |
| <b>DB</b><br>Kapitel 7                                    | <ul style="list-style-type: none"><li>• ADO.NET</li><li>• JDBC, DB-API</li><li>• Assistenten einsetzen</li></ul> |                               |                                                                                                                        |
| Einsatz von KI in der Softwareentwicklung                 |                                                                                                                  |                               |                                                                                                                        |
| KI-gestütztes Projekt- und Qualitätsmanagement            |                                                                                                                  |                               |                                                                                                                        |
| KI-gestützte Programmgenerierung                          |                                                                                                                  |                               |                                                                                                                        |
| Wesentliche Aspekte des EU AI Acts (KI-Verordnung der EU) |                                                                                                                  |                               |                                                                                                                        |

## 0.2 Unterrichten der einzelnen Lernfelder

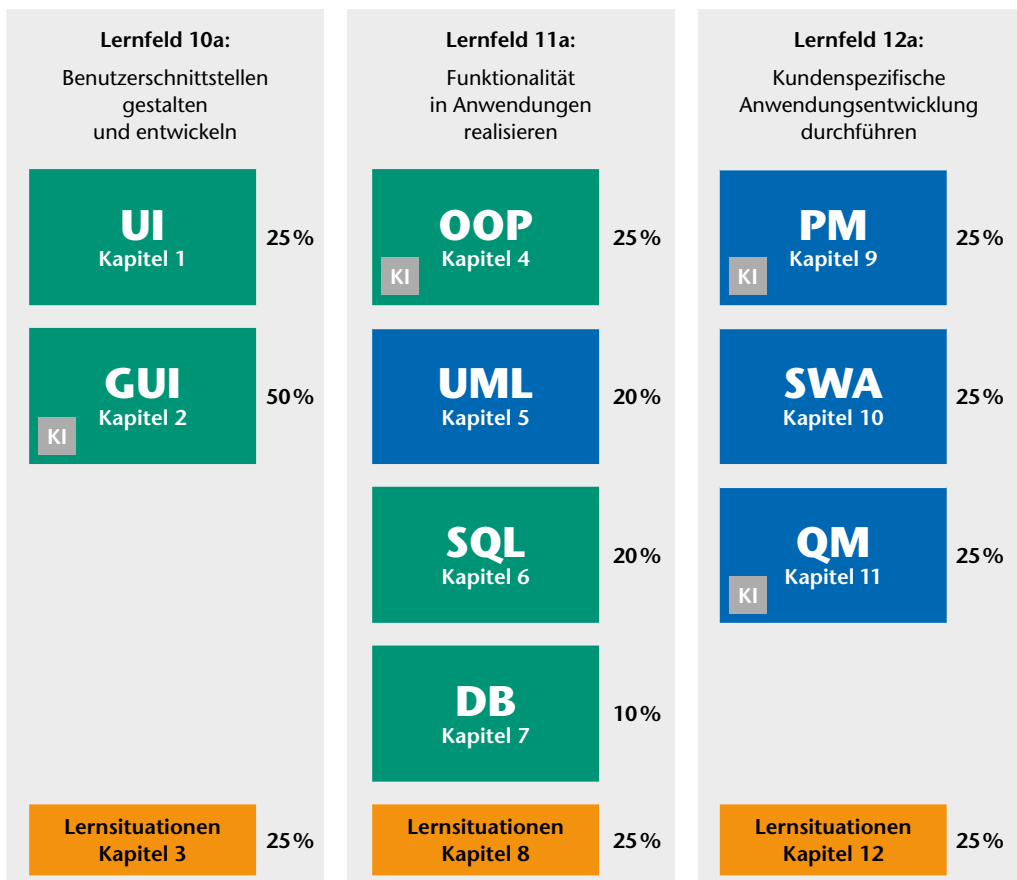
Die Lernfelder 10a–12a sind zwar getrennt zu betrachten, haben aber große Schnittmengen in den Voraussetzungen und den zu erwerbenden Fachkompetenzen. Aus diesen Gründen werden hier zwei Wege vorgeschlagen, den Unterricht zu strukturieren. Die angegebenen prozentualen Verteilungen sind dabei jeweils als grobe Richtschnur zu sehen. Die tatsächlichen Verteilungen hängen von den unterrichtlichen Umständen, den Vorkenntnissen und der Leistungsfähigkeit der jeweiligen Lerngruppe ab.

### Vorschlag 1: Lernfelder separat unterrichten

Der erste Vorschlag zeigt eine klare Trennung der Lernfelder. Die Reihenfolge der Kapitel dieses Buches entspricht diesem Vorschlag. Damit bietet das Werk eine klare Vorgehensweise für den Unterricht in Lernfeldern. Die oben identifizierten Fachkompetenzen sind entsprechend in die Struktur integriert.

Es ist zu beachten, dass bestimmte Themen in den Lernfeldern andere Themen voraussetzen (beispielsweise ist für die GUI-Entwicklung wichtig, dass die objektorientierte Programmierung vorab behandelt wurde). Daher sind Absprachen im Kollegium sinnvoll, falls die Lernfelder von verschiedenen Lehrkräften unterrichtet werden.

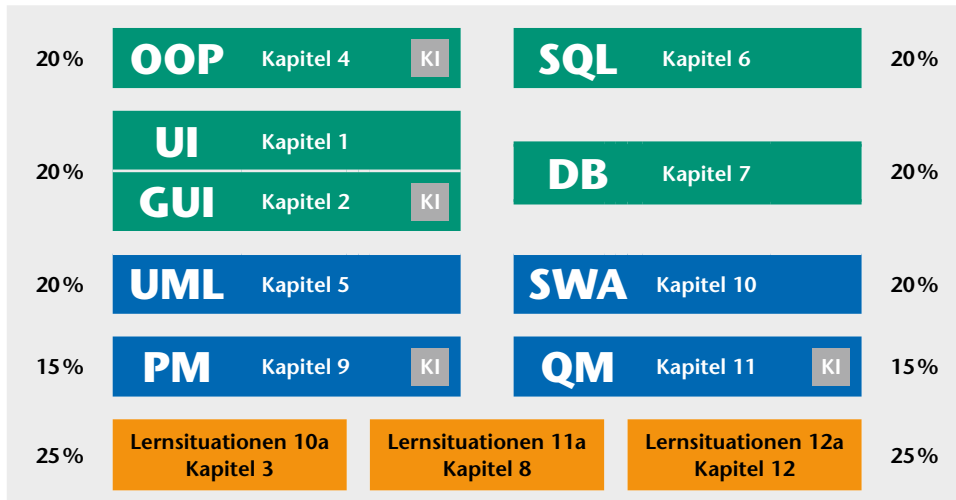
Die Lernsituationen können ein Themengebiet entweder abschließen oder themenübergreifend sein. In einigen Fällen haben die Lernsituationen auch Projektcharakter und können als Abschluss des kompletten Lernfeldes dienen.



## Vorschlag 2: Lernfelder übergreifend unterrichten

Der zweite Vorschlag ist die Zusammenlegung der Stunden der drei Lernfelder. Dabei werden zunächst die identifizierten Fachkompetenzen aufgebaut und anschließend eine Differenzierung der Lernfelder anhand von Lernsituationen vorgenommen, sodass auch eine differenzierte Leistungsüberprüfung möglich ist.

Der Vorteil dieser Vorgehensweise ist, dass zuerst alle Grundlagen-Themen unterrichtet werden können (beispielsweise **objektorientierte Programmierung** und **SQL** als Grundlage für **GUI-Entwicklung** und **Datenbankanbindung**). Zusätzlich erleichtert dies den Unterricht der übergeordneten Fachkompetenzen. Absprachen unter Kolleginnen und Kollegen sind auch hier sinnvoll, wenn die Themen von verschiedenen Lehrkräften unterrichtet werden.



## 0.3 Hinweise zu digitalen Zusatzbüchern

Zur **Wiederholung der Inhalte aus Lernfeld 5** sind im Buchpreis digitale Zusatzbücher enthalten:

- Europa-Nr.: 87423 | Elementare Programmierung mit C# → Weblink: [vel.plus/fit01](http://vel.plus/fit01)
- Europa-Nr.: 87669 | Elementare Programmierung mit Java → Weblink: [vel.plus/fit02](http://vel.plus/fit02)
- Europa-Nr.: 87874 | Elementare Programmierung mit Python → Weblink: [vel.plus/fit03](http://vel.plus/fit03)

In diesem Buch wird die weit verbreitete Sprache **C#** verwendet. Für den Unterricht mit **Java** und **Python** sind weitere Zusatzbücher enthalten (Wegweiser s. auch Kapitel 2, 4, 5 und 7):

- Europa-Nr.: 87676 | Objektorientierte Programmierung mit Java → Weblink: [vel.plus/fit04](http://vel.plus/fit04)
- Europa-Nr.: 87690 | GUI-Programmierung mit Java → Weblink: [vel.plus/fit05](http://vel.plus/fit05)
- Europa-Nr.: 87904 | Objektorientierte Programmierung mit Python → Weblink: [vel.plus/fit06](http://vel.plus/fit06)
- Europa-Nr.: 88123 | GUI-Programmierung mit Python → Weblink: [vel.plus/fit07](http://vel.plus/fit07)

Zur **Binnendifferenzierung** dienen die ebenfalls enthaltenen fortgeschrittenen Zusatzbücher:

- Europa-Nr.: 87447 | Fortgeschrittene Programmierung mit C# → Weblink: [vel.plus/fit08](http://vel.plus/fit08)
- Europa-Nr.: 87683 | Fortgeschrittene Programmierung mit Java → Weblink: [vel.plus/fit09](http://vel.plus/fit09)
- Europa-Nr.: 87935 | Fortgeschrittene Programmierung mit Python → Weblink: [vel.plus/fit10](http://vel.plus/fit10)

Achtung: Wenn Sie das vorliegende Buch als Druckversion gekauft haben, müssen die Inhalte zunächst freigeschaltet werden. Eine Anleitung dazu finden Sie auf der Umschlaginnenseite.

## Lernfeld 10a

Benutzerschnittstellen  
gestalten und entwickeln

# 1 Benutzerschnittstellen (UI) kennenlernen

## 1.1 Schnittstellen allgemein (Interfaces)

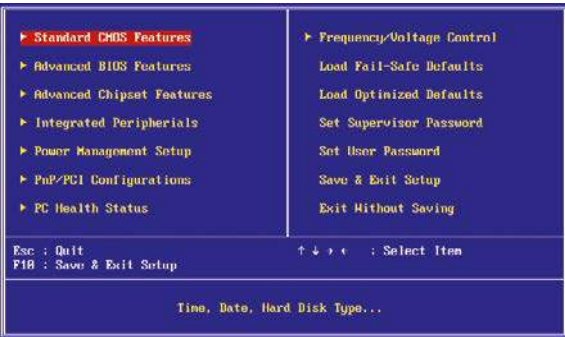
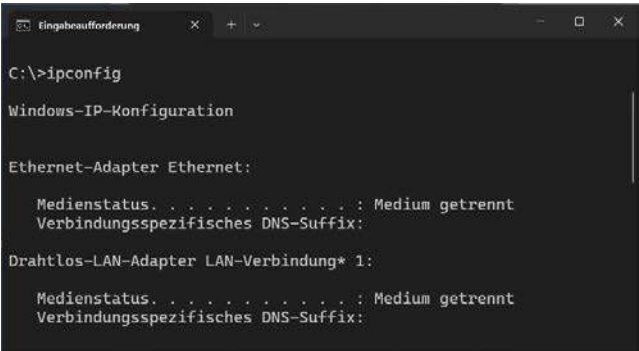
Mithilfe von Schnittstellen (engl. Interfaces) können Maschinen, Systeme oder auch Anwendungen miteinander kommunizieren. Die Schnittstellenbeschreibung legt dabei genau fest, wie diese Kommunikation stattfinden kann. Im IT-Bereich können vor allem Hardware- und Software-Schnittstellen unterschieden werden. Die folgende Tabelle zeigt einige dieser Schnittstellen.

| Hardwareschnittstellen                       | Softwareschnittstellen                                                        |
|----------------------------------------------|-------------------------------------------------------------------------------|
| USB (serielles Datenübertragungssystem)      | ODBC/JDBC (Datenbankverbindungen)                                             |
| WLAN (lokales Funknetz zur Datenübertragung) | Webservices (Datenaustausch über Netzwerke zwischen Maschinen bzw. Computern) |
| IEEE 1248 (parallele Datenschnittstelle)     | API (Programmierschnittstelle)                                                |

## 1.2 Grundlagen von Benutzerschnittstellen

Als Spezialisierung der allgemeinen Schnittstellen können die Benutzerschnittstellen (engl. User Interfaces UI) angesehen werden. Bei diesen Schnittstellen geht es vor allem um die Kommunikation eines Benutzers (Menschen) mit einer Maschine, einem System oder auch einer Anwendung. Diese Schnittstellen treten in verschiedenen Varianten auf. Die folgende Auflistung zeigt die gängigsten dieser Schnittstellen:

**CLI (Command Line Interface) – Kommandozeilenschnittstelle:** Bei diesem Interface kommuniziert der Benutzer über Kommandozeilen (Befehle) mit dem System (Computer, Maschine). Die Befehle werden in der Regel über die Tastatur eingegeben. Das wird beispielsweise genutzt bei *Windows-cmd* oder der Versionsverwaltung *Git*. Die nebenstehende Abbildung zeigt ein Beispiel: *ipconfig-Befehl über cmd*.



**TUI (Text User Interface) – Zeichenorientierte Benutzerschnittstellen:** Diese Schnittstellen sind wie bei der CLI textbasiert, aber es müssen keine Befehle über die Tastatur eingegeben werden. Die Kommunikation erfolgt über textbasierte Menüs, die mit Tastatur oder Maus gesteuert werden. Das *BIOS* eines Computers ist ein Beispiel für diese Art der Schnittstelle. Die nebenstehende Abbildung zeigt ein Beispiel: *BIOS-Screenshot*

**GUI (Graphical User Interface) – grafische Benutzerschnittstelle:** Im Laufe der letzten Jahrzehnte hat die grafische Benutzerschnittstelle oder Benutzeroberfläche die Kommandozeilenschnittstellen und die textbasierten Schnittstellen verdrängt. Die moderne Kommunikation mit Computern, Tablets oder Smartphones ist ohne grafische Oberflächen nicht mehr vorstellbar. Alle relevanten Betriebssysteme, die den Alltag der Menschen bestimmen, haben inzwischen eine GUI, mit der die Benutzer die Bedienung vornehmen. Dabei werden mithilfe der GUI komplexe grafische Oberflächen angezeigt, die vom Benutzer mit Eingabegeräten wie der Maus komfortabel bedient werden können. Wichtige Betriebssysteme, die GUIs einsetzen sind Windows, macOS, Linux, Android oder iOS. Die nebenstehende Abbildung zeigt ein Beispiel einer GUI.



☞ Zu GUI siehe auch Kapitel 2

**VUI (Voice User Interfaces VUI) – Sprachbasierte Benutzerschnittstellen:** Diese Schnittstellen erlauben eine Kommunikation mithilfe von gesprochener Sprache. Die Systeme verfügen über eine Spracherkennung und über eine entsprechend synthetische Sprachausgabe. Die Sprachassistenten/-innen der modernen Endgeräte sind Beispiele für diese Schnittstellen (**Amazon Alexa oder Apple Siri**).



**Weitere Schnittstellen:** Neben den bereits angegeben weit verbreiteten Benutzerschnittstellen sind auch noch viele weitere entwickelt worden oder befinden sich in der Entwicklung.

Dazu gehören beispielsweise die **NUI (Natural User Interfaces)**, die eine „natürliche“ Kommunikation für den Benutzer bieten wollen. Das kann zum Beispiel ein Touchscreen (in seiner Doppelfunktion als Eingabegerät und Interface) sein, der dem Benutzer Möglichkeiten der Kommunikation bietet, die seinem natürlichen Bewegungsverhalten besser entsprechen.

Die sogenannten **BCI (Brain Computer Interface)** gehen noch einen Schritt weiter: Sie ermöglichen eine direkte Kommunikation des menschlichen Gehirns mit dem System oder der Maschine. Derzeit werden nach offiziellen Informationen vor allem medizinische Anwendungen verfolgt, was mit der Hoffnung auf eine Hilfe für Menschen mit Einschränkungen verbunden wird. Die zentralen Herausforderungen sind Datenschutz, geistige Autonomie und fehlende Regulierung.

## 1.3 Ergonomie von Benutzerschnittstellen

Eine Benutzerschnittstelle soll die Kommunikation zwischen Mensch und Maschine ermöglichen. Dabei ist eine möglichst effiziente Kommunikation entscheidend. Mit dieser Problematik beschäftigt sich die Ergonomie<sup>1</sup> von Benutzerschnittstellen (speziell auch die Software-Ergonomie). Die Normenreihe **DIN EN ISO 9241** bietet einen geeigneten Rahmen, um auf diese Anforderungen reagieren zu können. Die Normenreihe bietet neben einer allgemeinen Einführung viele Kriterien zu Gestaltung von Benutzeroberflächen. Exemplarisch werden hier einige aufgeführt:

<sup>1</sup> Die Ergonomie beschäftigt sich mit menschlicher bzw. automatisierter Arbeit. Sie hat das Ziel, die Arbeitsbedingungen und den Arbeitsablauf so zu gestalten, dass der Mensch nicht unnötig belastet oder sogar geschädigt wird und das Ergebnis der Arbeit optimal ist.

- Leitfaden zu visuellen User-Interface-Elementen
- Anforderungen an Tastaturen und weitere Eingabegeräte
- Anforderungen an die Arbeitsplatzgestaltung und Körperhaltung
- Grundsätze und Empfehlungen der visuellen Informationsdarstellung
- Grundsätze der Dialoggestaltung und Benutzerführung

Als besonders relevant können die folgenden **sieben Interaktionsprinzipien** der Norm angesehen werden, die für die Gestaltung von Benutzerschnittstellen beachtet werden sollten:

| Interaktionsprinzip                   | Beschreibung/Beispiel                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1 Aufgabenangemessenheit</b>       | Die Schnittstelle sollte so gestaltet sein, dass die zu erledigende Aufgabe vom Benutzer reibungslos bearbeitet werden kann. Wenn beispielsweise Daten (Messwerte) in ein System eingetragen werden sollen, so sollte die Schnittstelle es ermöglichen, dass die Werte einzeln eingetragen, aber auch im Gesamten vom System übernommen werden können (beispielsweise die Möglichkeit, dass die Daten aus einer csv-Datei komplett eingelesen werden). |
| <b>2 Selbstbeschreibungsfähigkeit</b> | Die Schnittstelle informiert den Benutzer über Vorgänge oder Fehler und beschreibt mögliche Alternativen, sodass Arbeitsabläufe nicht an fehlenden Informationen scheitern. Beispielsweise ändert ein Button durch eine Nutzeraktion seine Beschriftung und zeigt auf diese Weise eine alternative Vorgehensweise an.                                                                                                                                  |
| <b>3 Erwartungskonformität</b>        | Die Schnittstelle sollte die Interaktionsmöglichkeiten so anbieten, wie es der Benutzer erwarten würde. Beispielsweise sollte das Schließen-Kreuz eines Fensters am oberen rechten Rand angeordnet sein und nicht an einer unerwarteten anderen Stelle.                                                                                                                                                                                                |
| <b>4 Erlernbarkeit</b>                | Die Schnittstelle sollte Möglichkeiten der Erlernbarkeit bieten, den Nutzer also beim Erlernen von Bedienungsabläufen unterstützen. Möglichkeiten sind Hilfe-Icons, schrittweise Anleitungen, Tooltips sowie begleitende Lernvideos oder Tutorials.                                                                                                                                                                                                    |
| <b>5 Steuerbarkeit</b>                | Der Benutzer sollte die Möglichkeiten erhalten, die Nutzung der Schnittstelle zu steuern. Das kann beispielsweise die Möglichkeit der Rücknahme eines Befehls sein oder das sinnvolle Überspringen bestimmter Standard-Einstellungen.                                                                                                                                                                                                                  |
| <b>6 Robustheit</b>                   | Die Robustheit ist ein sehr wichtiger Punkt bei der Nutzung einer Schnittstelle. Auf Fehleingaben oder auch das versehentliche Drücken eines Buttons sollte die Schnittstelle geeignet reagieren und Möglichkeiten der Korrektur anbieten.                                                                                                                                                                                                             |
| <b>7 Benutzerbindung</b>              | Schnittstellen sollten so gestaltet sein, dass sie die Benutzer binden – die Benutzer sollten gerne mit der Schnittstelle arbeiten. Die Schnittstelle vermittelt dem Benutzer, dass die Verwendung sicher ist und keine unbeabsichtigten Fehler zulässt. Eventuell motiviert die Schnittstelle den Benutzer auch bei der Bedienung – durch positive Rückmeldungen o.Ä.                                                                                 |

### 1.4 User Experience (UX)

Die User Experience (Nutzererfahrung, Nutzererlebnis) ist eine umfassende Beschreibung der Interaktion eines Benutzers mit einer Schnittstelle. Dabei werden nicht nur die technischen Aspekte beachtet, sondern auch die individuellen Eindrücke und Erfahrungen, die ein Benutzer bei der Interaktion macht. Ebenso werden die ästhetischen Aspekte bei der Nutzung einbezogen. Es entsteht also ein umfassendes Bild von der Nutzung. Dieses Bild soll dabei helfen, die Schnittstellen (Maschinen, Software etc.) so zu gestalten, dass es für den Benutzer insgesamt zu einer positiven Erfahrung kommt. Das kann folgende Aspekte beinhalten:

- Emotionen des Benutzers
- Erwartungen des Benutzers
- Werte des Benutzers
- Effizienz der Schnittstelle
- Effektivität der Schnittstelle
- Ästhetik der Schnittstelle
- und weitere ...

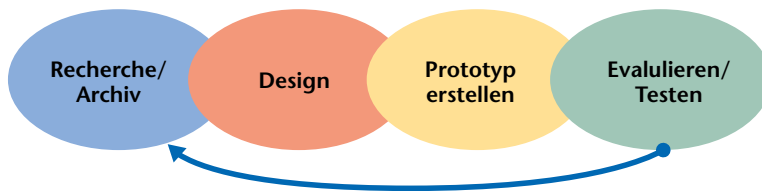
Mithilfe des nutzerzentrierten Designs (**User Centered Design UCD**) sollen diese Nutzererfahrungen in einem Designprozess einbezogen und damit optimale Ergebnisse für die Gestaltung von Schnittstellen erzielt werden. Der Ablauf des nutzerzentrierten Designs wird im folgenden Unterkapitel erläutert.

Weitergehende Konzepte betrachten auch **User Diversity** und **Technikakzeptanz**. User Diversity bezeichnet die Verschiedenheit der Nutzer im Hinblick auf Alter, Geschlecht, Sprache, Kultur, Bildung etc. Es ist eine Herausforderung, Anwendungen zu gestalten, die sowohl für 18-jährige Digital Natives als auch für 75-jährige Senioren gut passen. Technikakzeptanz beschreibt, inwieweit Nutzer überhaupt bereit sind, eine bestimmte neue Technologie einzusetzen. So können Anwendungen beispielsweise aus Datenschutzgründen oder wegen fehlender Transparenz abgelehnt werden.

## 1.5 User Centered Design (UCD)

Das nutzerzentrierte Design (engl. User Centered Design **UCD**) stellt den Nutzer in den Vordergrund des Designprozesses. Vor der Entwicklung der Benutzerschnittstelle (oder ganz allgemein eines Produktes) werden die Bedürfnisse (Erwartungen, Wünsche, Motive) des potenziellen Nutzers evaluiert oder zumindest antizipiert, um den Entwicklungsprozess danach auszurichten. Ein wesentliches Ziel von **UCD** ist eine hohe Gebrauchstüchtigkeit (Usability) der Benutzerschnittstelle oder des Produktes.

Der Prozess des User Centered Designs kann in vier grundsätzliche Phasen eingeteilt werden.



Wie in den folgenden Ausführungen deutlich wird, kann es nötig sein, dass die Phasen während des Designprozesses erneut durchlaufen werden. Die einzelnen Phasen werden im Folgenden genauer definiert. Ebenso werden Methoden vorgestellt, die in den jeweiligen Phasen zum Einsatz kommen können.

### Phase 1: Recherche/Analyse

In dieser Phase werden die Anforderungen an die Schnittstelle analysiert. Dazu gehört auch die Recherche zu allen Aspekten, die bei der Schnittstelle berücksichtigt werden müssen. Im Vordergrund steht jedoch der Nutzer, der mit geeigneten Methoden befragt oder analysiert wird.

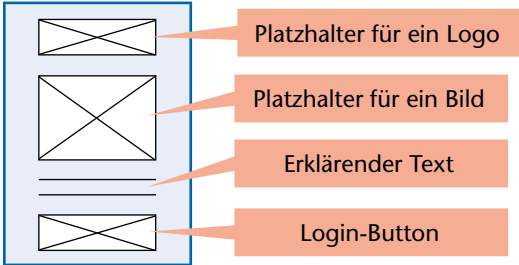
Folgende Methoden können eingesetzt werden:

| Methode                              | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Interview im Kontext</b>          | Der zukünftige Nutzer wird nicht nur einfach befragt, sondern das Interview findet dort statt, wo die zu entwickelnde Benutzerschnittstelle eingesetzt werden soll. Damit kann der Interviewer leichter erkennen, welche Besonderheiten in dieser Umgebung beachtet werden müssen. Beispielsweise muss eine Benutzerschnittstelle in einer Werkstatt völlig anderen Kriterien (Berücksichtigung von Lärm, Dreck etc.) genügen als eine Benutzerschnittstelle in einem Büro. |
| <b>Recherchen</b>                    | Umfassende Recherchen zum Umfeld der neuen Benutzerschnittstelle helfen später bei der zielgenauen Konzeption. Solche Recherchen sind mithilfe von Marktforschungsberichten, Data-Analytics-Tools oder auch klassischer Literatur möglich.                                                                                                                                                                                                                                  |
| <b>Umfragen/<br/>Online-Umfragen</b> | Mithilfe von (Online-)Umfragen können die Einschätzungen von vielen Benutzern abgefragt werden. Als Ergänzung zu den individuellen Interviews können damit weitere wichtige Aspekte der zu entwickelnden Benutzerschnittstelle identifiziert werden.                                                                                                                                                                                                                        |

### Phase 2: Design

Diese Phase basiert auf der Analyse/Recherche und setzt deren Erkenntnisse in einem Designprozess um. Am Ende dieser Phase steht eine Vorlage, die in der nächsten Phase in einen Prototyp überführt werden kann.

Folgende Methoden können eingesetzt werden:

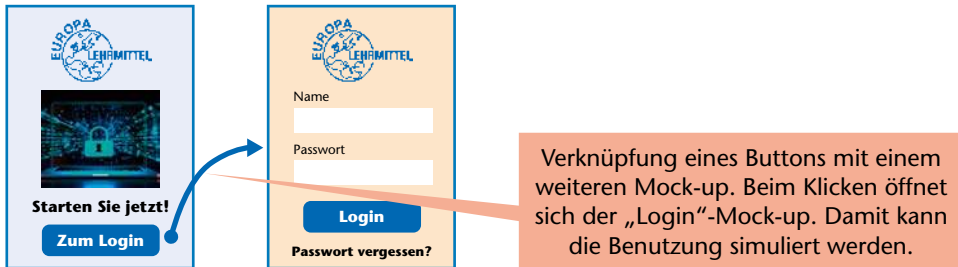
| Methode           | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Wireframes</b> | <p>Mithilfe von Wireframes<sup>2</sup> werden die Anforderungen in einem ersten Schritt grob umgesetzt. Die Wireframes können dabei ganz simple handschriftliche Skizzen sein, wie das folgende Beispiel (grober Entwurf einer Login-Benutzeroberfläche) zeigt:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Mock-ups</b>   | <p>Auf Basis der Wireframes werden die konkreten Oberflächen umgesetzt. Die Mock-ups<sup>3</sup> spiegeln die finalen Benutzerschnittstellen schon sehr konkret wider, haben aber noch keine echten Funktionalitäten. Diese werden erst bei der Erstellung des Prototyps eingebaut.</p> <p>Das folgende Beispiel zeigt einen Mock-up, der das Wireframe aus dem obigen Beispiel konkretisiert:</p>  <p>Bei einem Mock-up werden</p> <ul style="list-style-type: none"><li>• Farben</li><li>• Formen</li><li>• Typografie</li><li>• Muster (Patterns)</li></ul> <p>berücksichtigt.</p> <p>Mock-ups können mithilfe folgender Tools erstellt werden: <i>Figma, Adobe XD, Sketch, Canva</i> und weitere (einige sind nach einer Registrierung kostenfrei nutzbar).</p> |

### Phase 3: Prototypen erstellen

Nach der Designphase und der Erstellung von Mock-ups können Prototypen erstellt werden. Im Gegensatz zum Mock-up verfügen die Prototypen bereits über (eingeschränkte) Funktionalitäten. Damit spiegeln sie das Endprodukt schon sehr realistisch wider. Der Benutzer/Kunde kann mit einem Prototyp interagieren und somit schon erste Erfahrungen im Umgang mit dem möglichen Endprodukt sammeln. Das Erstellen eines Prototyps kann als Fortführung des Designs betrachtet werden. Die o. a. Tools (wie Figma) stellen die nötigen Funktionalitäten zur Verfügung, um einen Mock-up interaktiv zu gestalten und bestimmte Funktionalitäten zu implementieren. Beispielsweise könnte das Mock-up aus dem obigen Beispiel mit entsprechenden Interaktionsmöglichkeiten versehen werden:

<sup>2</sup> deutsch: Drahtmodelle. Der Begriff stammt ursprünglich aus dem Schneiderberuf und bezeichnet dort eine sehr einfache, aus Drähten gefertigte Schneiderpuppe.

<sup>3</sup> deutsch: Attrappe, Nachbildung



#### Phase 4: Evaluieren/Testen

Diese letzte Phase dient dazu, den Prototyp intensiv mit Benutzern/Kunden zu testen. Mithilfe dieser Ergebnisse können die Prototypen so lange weiterentwickelt, geändert und verbessert werden, bis die finale Anwendung oder das finale Produkt dem Benutzer/Kunden übergeben werden kann. Dazu kann es nötig sein, dass noch einmal mit der Recherche/Analyse-Phase begonnen wird und die weiteren Phasen dann ebenfalls (zumindest teilweise) erneut durchlaufen werden. Es ist also insgesamt ein iterativer, agiler Prozess.

Folgende Methoden können für diese Phase eingesetzt werden:

| Methode        | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Usability-Test | <p>Mit dem Usability-Test wird die Benutzerschnittstelle auf ihre <b>Gebrauchstüchtigkeit</b> getestet. Als Testpersonen werden Benutzer aus der Zielgruppe der Schnittstelle herangezogen. Es sind also nicht die Entwickler/Designer, die diesen Test durchführen. Entwickler/Designer testen auf andere Weise die Schnittstelle oder das Produkt (beispielsweise durch White-Box-Tests o. Ä.). Der Usability-Test kann von den Benutzern alleine durchgeführt werden, oder er wird durch das Entwicklungsteam/Design team begleitet. Methoden wie das Eye-Tracking (siehe unten) können ebenfalls Bestandteil des Tests sein.</p> <p>Ein <b>Usability-Test</b> kann in drei Schritte eingeteilt werden. Für jeden Schritt sind bestimmte Voraussetzungen zu schaffen bzw. bestimmte Aspekte zu beachten.</p> <div style="text-align: center;"> <p><b>Vorbereitung:</b></p> <p>Testkonzept erstellen, Ziele definieren, Rollen festlegen,<br/>Durchführungsort festlegen, Testpersonen einladen,<br/>Messmethoden festlegen</p> <p>↓</p> <p><b>Durchführung:</b></p> <p>Testpersonen begrüßen, Ablauf und Aufgabenstellung erklären,<br/>während der Durchführung beobachten und Notizen machen</p> <p>↓</p> <p><b>Auswertung:</b></p> <p>Auswertung der Beobachtungen und Notizen, Befragung der Testpersonen</p> <p>↓</p> <p><b>nächsten Prototyp auf Basis der Erkenntnisse erstellen</b></p> </div> |
| Eye-Tracking   | <p>Mithilfe des Eye-Tracking wird gemessen, in welcher Reihenfolge und wie lange Benutzer bestimmte Bereiche einer Benutzeroberfläche betrachten.</p> <p>Dadurch können Bereiche oder Elemente identifiziert werden, die der Benutzer beispielsweise gar nicht oder verstärkt betrachtet hat. Mit diesen Messergebnissen können Veränderungen an der Benutzerschnittstelle vorgenommen werden. Beispielsweise könnte ein wichtiger Button, den der Benutzer bei den Messungen meistens übersehen hatte, mehr im Fokus des Benutzers platziert werden.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## 1.6 Aufgaben zu Benutzerschnittstellen (UI)

### Aufgabe 1:

Zu den Begriffen **UX** und **Usability** liegen verschiedene Informationen bzw. Beschreibungen vor. Ordnen Sie den Beschreibungen den korrekten Begriff zu (entweder **UX** oder **Usability**).

| Beschreibungen                                                                                                                                                             | Begriff |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| Es ist das Ausmaß, in dem ein Produkt durch Nutzer in einem bestimmten Kontext genutzt werden kann. Dabei werden Ziele effektiv, effizient und zufriedenstellend erreicht. |         |
| Es ist eine umfassende Beschreibung der Interaktion eines Benutzers mit einer Schnittstelle.                                                                               |         |
| Eine treffende Übersetzung für den Begriff ist die Gebrauchstauglichkeit.                                                                                                  |         |
| Es werden nicht nur die technischen Aspekte beachtet, sondern auch die individuellen Eindrücke und Erfahrungen, die ein Benutzer bei der Interaktion macht.                |         |
| Es werden auch ästhetische Aspekte bei der Nutzung beachtet.                                                                                                               |         |

### Aufgabe 2:

Die folgenden Mock-ups sollen Oberflächen einer neu zu entwickelnden App darstellen (Login-Seite, Startseite nach erfolgreichem Login sowie Profil-Seite). Prüfen Sie die Mock-ups kritisch auf die Verträglichkeit mit den sieben Interaktionsprinzipien der Normenreihe **DIN EN ISO 9241** und kommentieren Sie entsprechend.

App Mock-up-Version 1.0 – Login

Name:

Passwort:

**LOGIN**

Neu?

App Mock-up-Version 1.0 – Start

**Mein Profil**

Weitere Seite 1 (TODO)

Weitere Seite 2 (TODO)

App Mock-up-Version 1.0 – Profil

Name:

E-Mail:

Anschrift:

**ÄNDERN**

### Aufgabe 3:

Ordnen Sie die folgenden Begriffe den vier Phasen des **User Centered Design (UCD)** zu:

|          |                |                |         |
|----------|----------------|----------------|---------|
| Umfragen | Wireframes     | Kundenfeedback |         |
| Mock-ups | Usability-Test | Interviews     | Skizzen |