

## Das erwartet Sie:

- Daten und Informationen unterscheiden
- Prozess der Softwareentwicklung



# Software zur Verwaltung von Daten anpassen



**Lernfeld 5:        Software zur Verwaltung von Daten  
                         anpassen**

**1. Ausbildungsjahr  
Zeitrichtwert: 80 Stunden**

**Die Schülerinnen und Schüler verfügen über die Kompetenz, Informationen mittels Daten abzubilden, diese Daten zu verwalten und dazu Software anzupassen.**

Die Schülerinnen und Schüler informieren sich innerhalb eines Projektes über die Abbildung von Informationen mittels Daten. Dabei **analysieren** sie Daten hinsichtlich Herkunft, Art, Verfügbarkeit, Datenschutz, Datensicherheit und Speicheranforderung und berücksichtigen Datenformate und Speicherlösungen.

Sie **planen** die Anpassung einer Anwendung zur Verwaltung der Datenbestände und entwickeln Testfälle. Dabei **entscheiden** sie sich für ein Vorgehen.

Die Schülerinnen und Schüler **implementieren** die Anpassung der Anwendung, auch im Team und erstellen eine Softwaredokumentation.

Sie **testen** die Funktion der Anwendung und **beurteilen** deren Eignung zur Bewältigung der gestellten Anforderungen.

Sie **evaluieren** den Prozess der Softwareentwicklung.

# Die Themen und Lernziele

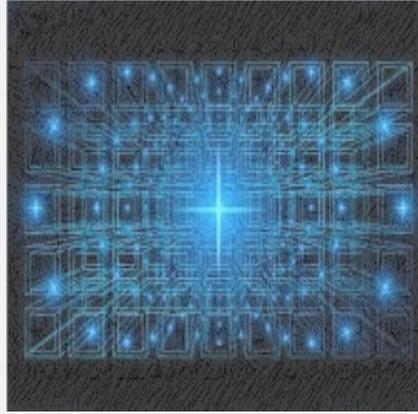


Das Umfeld der Softwareentwicklung analysieren

---

## Lernziel

Aufgaben und Kompetenzen in der SE kennenlernen

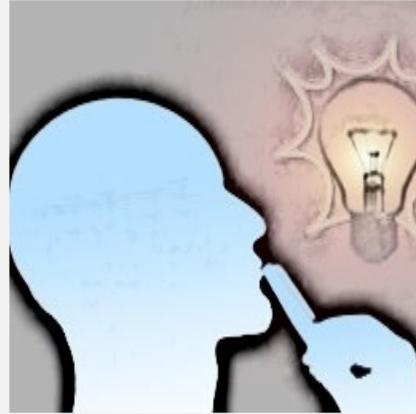


Grundlagen zur Verwaltung von Daten

---

## Lernziel

Informationen versus Daten



Den Prozess der Softwareentwicklung analysieren

---

## Lernziel

Prozessphasen sowie Vorgehensmodelle kennenlernen



Den Prozess der Anforderungsspezifikation beschreiben

---

## Lernziel

Anforderungen an die zukünftige Software spezifizieren können



Einfache Anwendungen in Python schreiben

---

## Lernziel

Programmiersprachen und –werkzeuge unterscheiden lernen

# Die Themen und Lernziele



Auf Dateien in  
Anwendungen  
zugreifen

---

## Lernziel

Daten speichern und  
einlesen lernen



Verwaltung der  
Daten mithilfe von  
Datenbanken

---

## Lernziel

Grundlagen von  
relationalen Datenbanken



Software testen  
und dokumentieren

---

## Lernziel

Qualitätsbewusstsein  
entwickeln



Prozess der  
Softwareentwicklung  
evaluieren

---

## Lernziel

Reflexion



## Das Umfeld der Softwareentwicklung analysieren

### Lernziel

Aufgaben und  
Kompetenzen in der  
Softwareentwicklung  
kennenlernen

# Der heutige Tag

Softwareentwicklung

Aufgaben u.  
Kompetenzen  
von Software-  
entwicklern

**Software  
unterscheiden**

Neu-  
entwicklung  
oder  
Anpassung

# Was antworten Sie auf die Frage nach Ihrer Ausbildung

1. Programmierer
2. Software-Entwickler
3. Informatiker
4. Irgendetwas mit Computern

» *People I am trying to impress: **Software Engineer** (actual title).*

*People who are familiar with technology: **Developer***

*People less familiar with technology: **Programmer***

*My Grandma who still uses CompuServe 2000: **I work on computers.** «*

» *It depends whether the job is more an issue of design or implementation.* «

Quelle: <https://jaxenter.de/sind-sie-programmierer-oder-software-entwickler-1545>

# Programmierer/in

## Programmierer/-in

sind im Bereich IT tätig. Sie entwickeln und konzipieren Softwareprogramme für unterschiedliche Einsatzgebiete

Diese reichen von Unterhaltungssoftware bis hin zu Office-Applikationen und medizinischen Anwendungen

**Programmierer** 

- Aufgaben**
  - Entwicklung und Konzeption von Software
  - Fehlersuche und Verbesserung
  - Identifizierung von Kundenwünschen und Marktbedürfnissen
- Voraussetzungen**
  - Abgeschlossenes Studium z.B. im Bereich: Informatik, Mathematik, Ingenieurwesen
  - Abgeschlossene Ausbildung zum Fachinformatiker
  - Quereinsteiger
- Skills**
  - Umgang mit spezialisierter Software
  - Programmiersprachen
  - Teamfähigkeit
  - Kommunikationsstärke
- Gehalt**

Einstieg:	40.000 €
Durchschnitt:	55.000 €
Top:	80.000 €

<https://www.freelancermap.de/blog/was-macht-ein-programmierer/>

# Softwareentwickler/in

## Softwareentwickler/-in

konzipieren und entwickeln Programme oder Teile von Programmen und Applikationen für Computer, Tablets, Smartphones und weitere Hardware

**Softwareentwickler** 

**Aufgaben**

- Entwicklung von Software
- Fehlersuche in Programmcodes
- Laufende Optimierung der Software
- Identifizierung von Marktbedürfnissen

**Voraussetzungen**

- Abgeschlossenes Informatik, Wirtschaftsinformatik, Gamedesign oder UX-Design Studium
- Ausbildung im IT- oder Informatikbereich

**Skills**

- Umgang mit spezifischer Software
- Programmiersprachen
- Schreiben von Codes
- Teamfähigkeit

**Gehalt**

Einstieg:	40.000 €
Durchschnitt:	55.000 €
Top:	80.000 €

<https://www.freelancermap.de/blog/was-macht-ein-programmierer/>

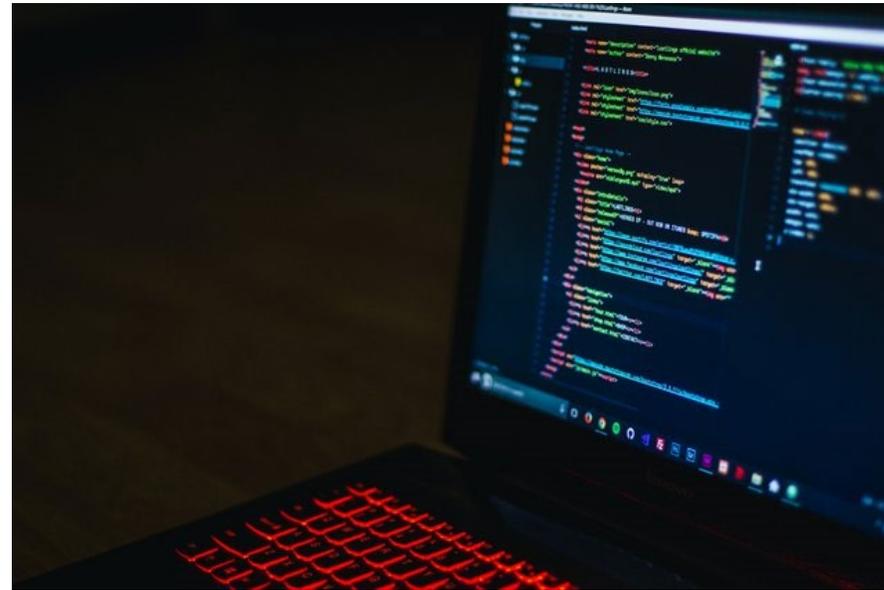
# 5.1.1 Aufgaben des Fachinformatikers/-in

## Systemintegrator/-in



Aber ! nicht nur Umgang mit Hardware

## Anwendungsentwickler/-in



Aber ! nicht nur Programmierung

# Anforderungen Fachinformatiker/-in Anwendungsentwicklung

- Anwendungsentwicklung, Softwareprojekte konzipieren (planen)
- Realisieren (Umsetzung)
- Software für Kunden, um bspw. Geschäftsprozesse effektiver zu gestalten, unternehmensabhängig
- IT-Sicherheitsregeln und Datenschutz
- Testfälle definieren, Testergebnisse protokollieren
- Software warten, wenn fertig, Fehler beseitigen
- Dokumentieren, Benutzerhandbuch
- Kunden oder Kollegen im Umgang mit der Software schulen
- Was für Technologien nutzt man? Abh. vom Betrieb, Schule



# Anforderungen Fachinformatiker/-in Systemintegration

- Auf Hardware und Netzwerke spezialisiert
- Kernaufgabe: Netzwerke in Unternehmen einrichten und warten
- Hardwarekomponente/n beschaffen und diese einrichten
- Sorgt dafür, dass die IT am Arbeitsplatz immer funktioniert
- Probleme an Computern beheben
- Neue Computer einrichten – alles, was zum Arbeiten benötigt wird, darauf installieren
- Kunden oder Kollegen im Umgang mit dem Computer schulen
- u. v. m.



# Programmiersprachen

## Die 6 wichtigsten

- Java
- C/C++
- Java Script
- PHP
- Python
- C#



HTML, CSS

SQL

## Weitere Programmiersprachen

- Swift
- Visual Basic
- Ruby
- Go
- Rust
- Object Pascal

# Programmiersprachen

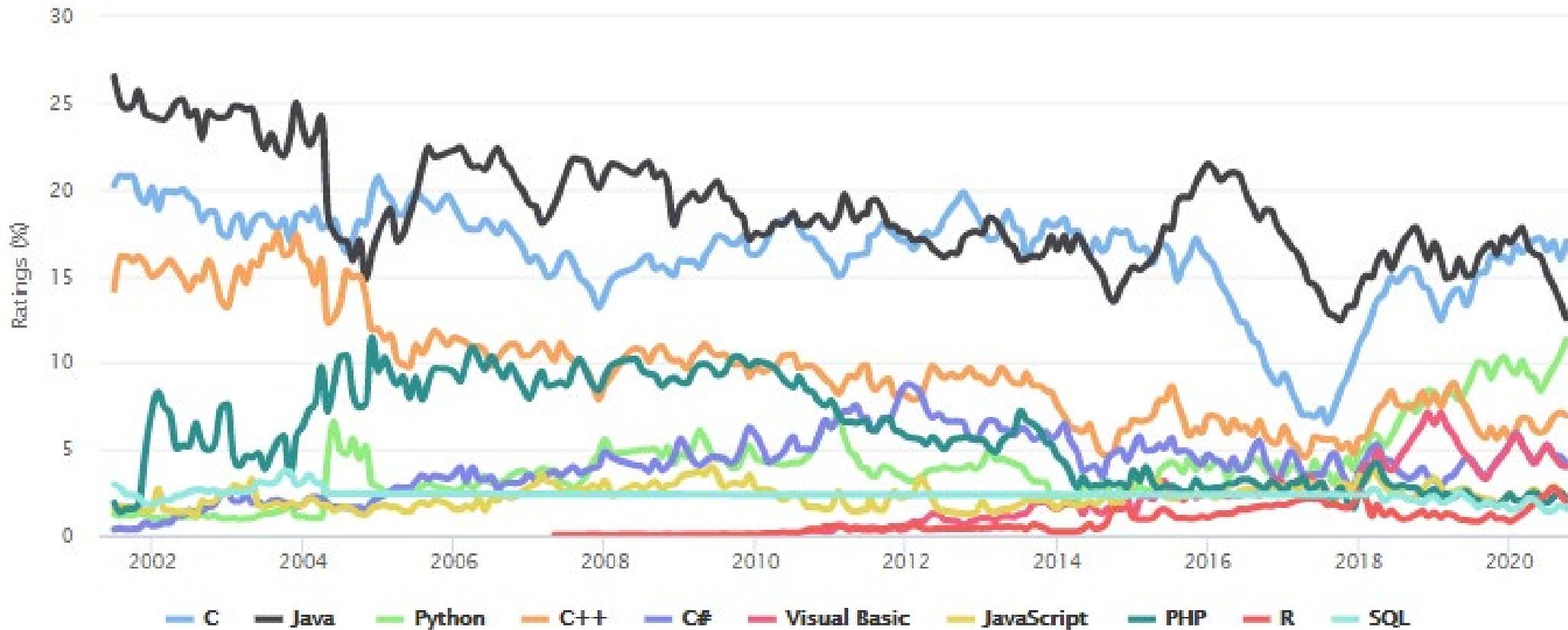
<https://www.tiobe.com/tiobe-index/>

Oct 2020	Oct 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.95%	+0.77%
2	1	▼	Java	12.56%	-4.32%
3	3		Python	11.28%	+2.19%
4	4		C++	6.94%	+0.71%
5	5		C#	4.16%	+0.30%
6	6		Visual Basic	3.97%	+0.23%
7	7		JavaScript	2.14%	+0.06%
8	9	▲	PHP	2.09%	+0.18%
9	15	▲▲	R	1.99%	+0.73%
10	8	▼	SQL	1.57%	-0.37%
11	19	▲▲	Perl	1.43%	+0.40%
12	11	▼	Groovy	1.23%	-0.16%
13	13		Ruby	1.16%	-0.16%
14	17	▲	Go	1.16%	+0.06%
15	20	▲▲	MATLAB	1.12%	+0.19%
16	12	▼▼	Swift	1.09%	-0.28%
17	14	▼	Assembly language	1.08%	-0.23%
18	10	▼▼	Objective-C	0.86%	-0.64%
19	16	▼	Classic Visual Basic	0.77%	-0.46%
20	22	▲	PL/SQL	0.77%	-0.06%

# Programmiersprachen

TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



<https://www.tiobe.com/tiobe-index/>

# Softwareentwickler (SEs) - Varianten

Frontend- und Backend-Entwickler

Mobile-Entwickler

Algorithmen-Ingenieur

Spiele-Entwickler

DevOps-Entwickler

System-Entwickler

Test-Ingenieur

Web-Entwickler

Compiler Ingenieur

Full-Stack-Entwickler

Datenbankentwickler

CRM-, ERP- und CMS-Entwickler

Betriebssystem-Entwickler

Datenanalyst

Embedded-System-Entwickler

Junior-/Senior-Entwickler



# Junior- /Senior-Entwickler

## Junior

- Weniger als 2 bis 3 Jahre Erfahrung in einer gegebenen Technologie
- Die meisten Aufgaben werden überwacht bzw. kontrolliert
- Zuweisung der am wenigsten komplexen Aufträge

## Senior

- Experte für einen bestehenden Technologie-Stack
- Zuständig für das Mentoring der nächsten Entwicklergeneration (Junior Teammitglieder)
- U. U. Entwicklung von Lösungen
- Führungsqualität (Projektleiter)
- Kundenkontakt
- Sammeln von Produkthanforderungen
- Innerhalb vorgegebener Deadlines ein funktionierendes Produkt abgeben

## DevOps

- Ist ein Akronym, das sich aus **Development**, also Entwicklung, und **Operations** herleitet
- Soll durch gemeinsame Anreize, Prozesse und Software-Werkzeuge (englisch: *tools*) eine effektivere und effizientere Zusammenarbeit der Bereiche *Dev*, *Ops* und Qualitätssicherung (QS) ermöglichen
- Mit *DevOps* sollen die Qualität der Software, die Geschwindigkeit der Entwicklung und der Auslieferung sowie das Miteinander der beteiligten Teams verbessert werden
- Der **DevOps**-Manager kann Entwicklungstools entwickeln oder begutachten und in die Praxis umsetzen

# Test-Ingenieur

## Bug

- Im Englischen ist "**bug**" eine Wanze, im Amerikanischen allgemein ein Insekt
- In der **Computersprache** ist damit ein Fehler gemeint, der beim Programmieren entstanden ist

regelmäßig · haben

 **debuggen**

**debuggt** · **debuggte**  
*hat **gedebuggt***

Netzverb ([www.verbformen.de](http://www.verbformen.de)) · CC BY-SA 4.0



# 5.1.1 Aufgaben und Kompetenzen

- Kernkompetenzen
  - Analysieren von Prozessen und Anforderungen in der IT
  - Beraten und Betreuen von Kunden
  - Planen, Durchführen und Auswerten von Tests
  - Implementieren (Programmieren mit einer oder mehreren Programmiersprachen)
  - Integrieren (Systemlandschaft)
  - Pflege, Wartung und Dokumentation von Software und Prozessen
  - Fachkenntnisse
- Soft-Skills
  - Technisches und mathematisches Verständnis
  - Analytisches Denkvermögen
  - Teamfähigkeit und Organisationstalent
  - Aufgeschlossenheit und Umsichtigkeit
  - Kommunikation
  - Sehr gute Englischkenntnisse (Wort, Schrift)
  - Abstraktionsvermögen
  - Denken über den Tellerrand hinaus
  - Interesse an innovativen Entwicklungen
  - Geduld und Spaß an komplexen Fragestellungen



Welche Aussagen sind richtig?

- a) Softwareentwickler beschäftigen sich nur mit Programmieren von Software.
- b) Die Aufgabenbereiche von Softwareentwicklern sind sehr vielfältig.
- c) Neben fundierten fachlichen Kenntnissen benötigen Softwareentwickler auch noch andere Kompetenzen.
- d) Java, C# und Python sind heutzutage wichtige Programmiersprachen.
- e) Kenntnisse in mehreren Programmiersprachen sind für Softwareentwickler von Vorteil.
- f) Entwicklerdokumentationen werden nicht von Softwareentwicklern verfasst.
- g) Softwareentwickler müssen auch Fachkenntnisse im Testen und Debuggen von Software besitzen.
- h) Heutzutage müssen Softwareentwickler nicht mehr teamfähig sein.

# Kompetenzcheck

Wählen Sie drei nicht fachliche Kompetenzen für Softwareentwickler aus, welche Ihnen am wichtigsten erscheinen und begründen Sie kurz Ihre Entscheidung.

Stellen Sie in Partnerarbeit Ihre Überlegungen einem anderen Teilnehmenden vor.



# Aufgabe – Ergebnisse (Vorschlag)

## Wichtige, nicht-fachliche Kompetenzen

- Offenheit
- Neugier
- Teamfähigkeit
  - Kommunikation
  - Wissen weitergeben, ohne Arroganz
- Fähigkeit zur Problemlösung

# Aufgabe – Ergebnisse (Vorschlag)

## Wichtige, nicht-fachliche Kompetenzen

- Soziale Kompetenz
  - Umgänglichkeit
  - Lernfähigkeit, Bereitschaft, sich kontinuierlich zu verbessern
  - Ausdauer
  - Flexibilität
  - Kreativität
- Technisch/methodisch an Aufgaben gehen können
- Zeitmanagement
- Entscheidung treffen können, nach Kenntnis aller Konsequenzen
- Kundenorientiert/empathisch

## 5.1.2 Software aus der Sicht von SEs beschreiben

Was ist Software?

- Sehr teuer oder auch kostenfrei
- Nicht greifbar, immateriell
- Verschleißt nicht, altert aber relativ zur Umgebung
- Ist relativ leicht änderbar, aber es gibt keine Ersatzteile
- Qualität ist schwer zu definieren und zu messen

## 5.1.2 Software aus der Sicht von SEs beschreiben

Was ist Software?

Software besteht aus in elektronischer Form durch die Zahlen 0 und 1 dargestellte und abgesicherte Informationen, die im Einzelnen keinen Sinn ergeben, aber im Ergebnis durch aufwendige Mechanismen der Fehlerkorrektur im Hintergrund etwas auf einem Bildschirm bzw. zuerst in einem Speicher darstellen



„[...] alle nicht-physischen Komponenten eines computergestützten Systems.“

Quelle: Chip.de

**Software** ['spɪf(t)wɛ:] (dt. = weiche Ware [von] soft = leicht veränderbare Komponenten)

Quelle: de.wikipedia.org

## 5.1.2 Software aus der Sicht von SEs beschreiben

Warum braucht man Software?

- Software kann Prozesse
  - darstellen
  - elektronisch ablaufen lassen
  - simulieren
  - ordnen
  - berechnen
  - verbinden
- und sogar aufbauen

## 5.1.2 Software aus der Sicht von SEs beschreiben

Wirtschaftliche Bedeutung von Software?

Link:

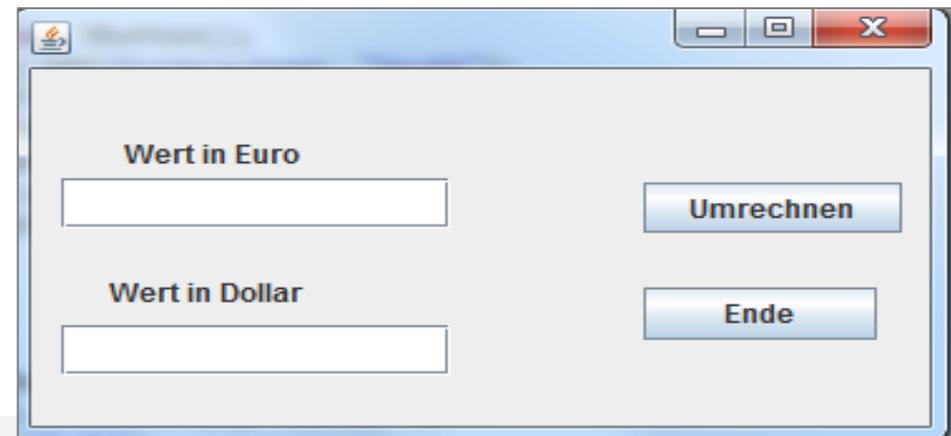
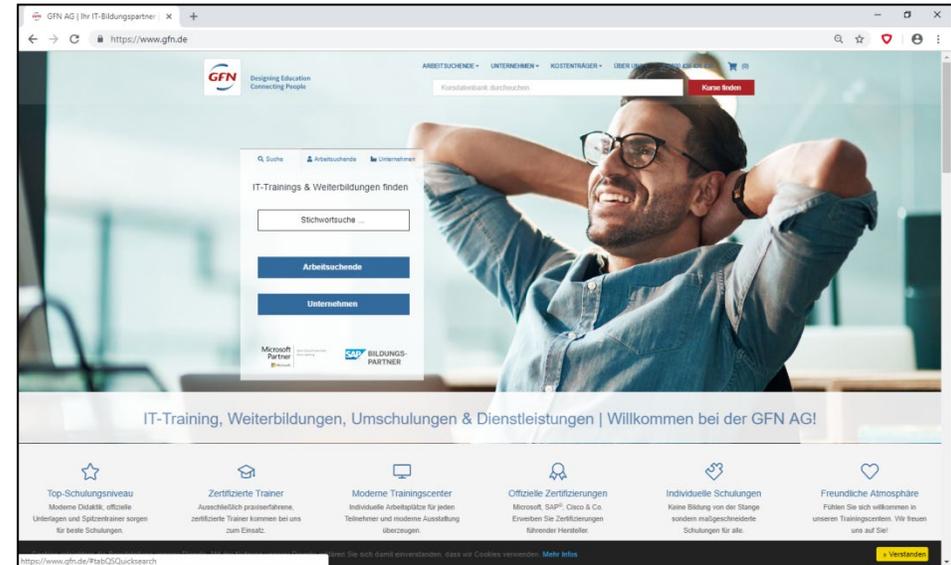
<https://www.spektrum.de/news/wirtschaftliche-bedeutung-der-software-industrie-in-deutschland/349994>

[https://softwareimpact.bsa.org/eu/pdf/factsheets/EU\\_Germany\\_Software\\_Economic\\_Impact\\_FACTSHEET\\_de.pdf](https://softwareimpact.bsa.org/eu/pdf/factsheets/EU_Germany_Software_Economic_Impact_FACTSHEET_de.pdf)

# Anwendungsbereiche 1

Webseiten im Inter- bzw. Intranet als Java-Applet

Plattformunabhängige Desktopanwendungen



# Anwendungsbereiche 2

Anwendungen auf  
Chipkarten

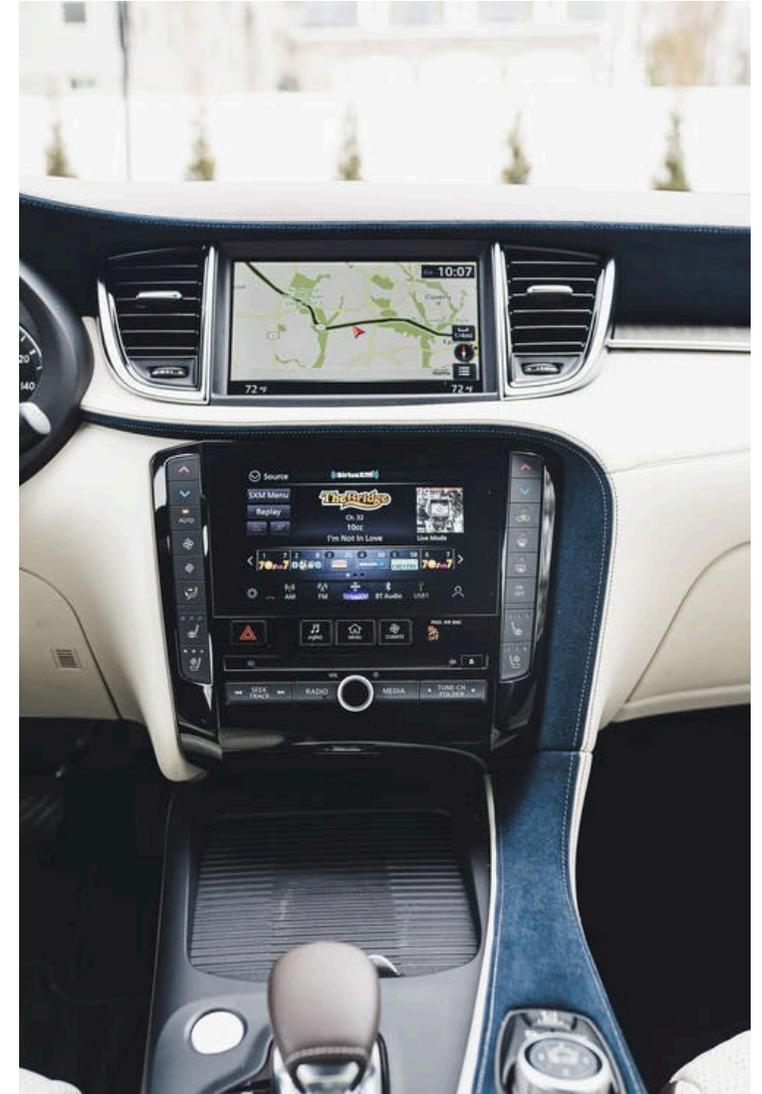


Anwendungen (Apps) auf  
Mobiltelefonen/  
Smartphones, Tablets



# Anwendungsbereiche 3

Embedded Java in  
Eingebetteten Umgebungen  
(Internet Browser,  
Mobiltelefone,  
Fahrzeuge etc.)



# Anwendungsbereiche 4

## Echtzeitsysteme

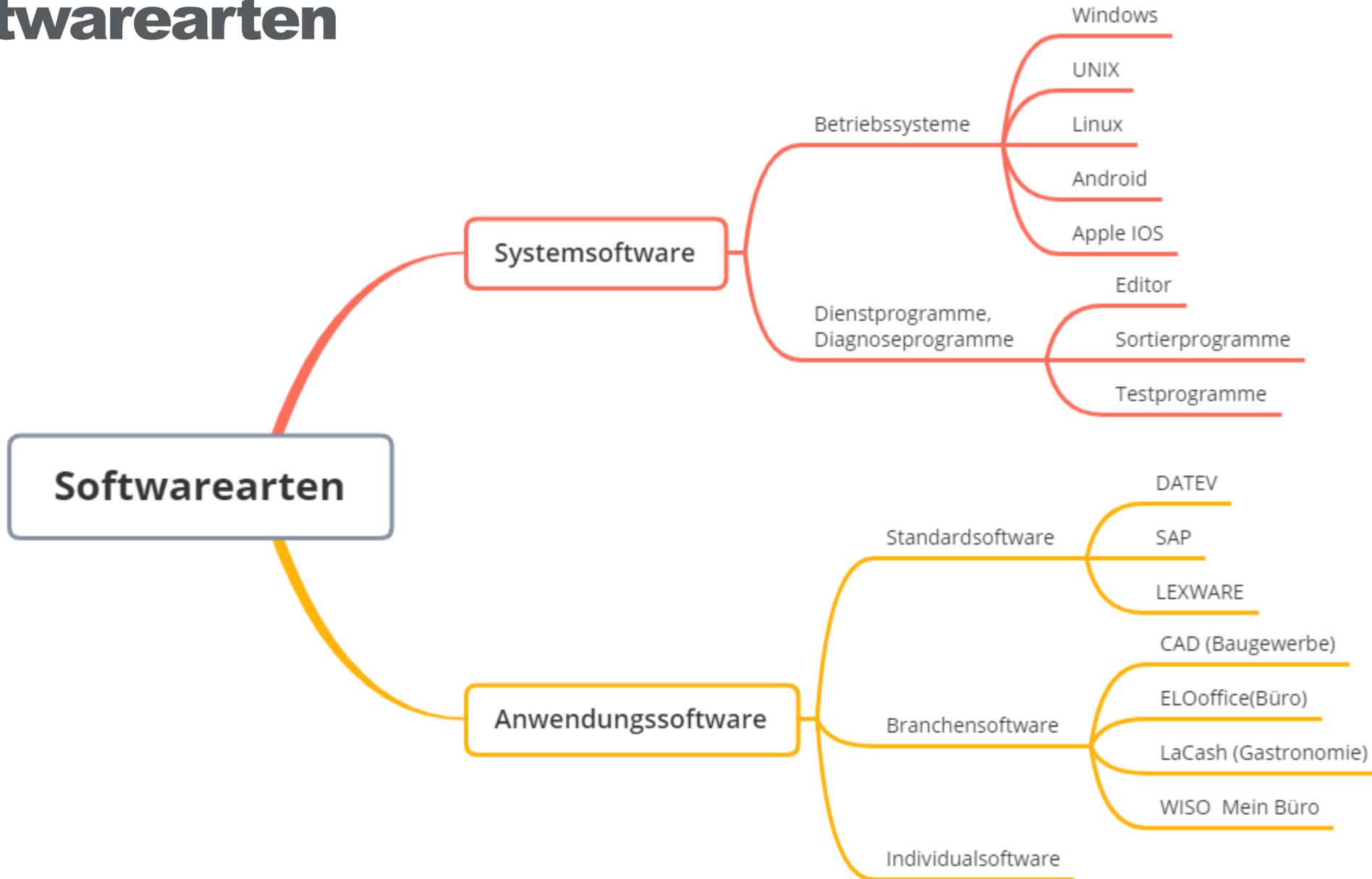


1.200 Prozessoren, 4 Milliarden Lines of Codes (LoC)



1 Prozessor, 500.000 Lines of Codes (LoC)

# Softwarearten



## 5.1.2 Software aus der Sicht von SEs beschreiben

### Aufgabenbezogen

#### **Systemsoftware**

unterstützt und ermöglicht das Ausführen von  
Anwendungssoftware  
(Betriebssysteme, Treiber usw.)

#### **Unterstützungssoftware**

Programme, welche bei der Entwicklung helfen oder  
nicht anwendungsspezifische Leistungen erbringen  
(Compiler, Editoren usw.) unterstützen und  
ermöglichen

#### **Anwendersoftware**

Umfasst alle Programme, die betriebswirtschaftliche, technische oder branchenbezogene Anwendungen  
unterstützen  
(Lern- und Unterhaltungssoftware, Buchhaltungsprogramme usw.)

## 5.1.2 Software aus der Sicht von SEs unterscheiden

### Nach Art des Auftraggebers

- Standardsoftware
  - funktionsbezogene Standardsoftware (CAD, Buchhaltungssoftware, PPS)
  - funktionsübergreifende Standardsoftware (ERP, Office)
  - Branchensoftware
- Individualsoftware

### Nach Art der Einbettung

- nicht eingebettete Software (wird installiert)
- fest in einem Gerät zu dessen Steuerung untergebrachte Software (z. B. ROM)

### Nach Nutzungsrecht

- Proprietär (kommerziell, Copyright, Quellcode nicht öffentlich)
  - kostenpflichtig
  - ohne finanziellen Aufwand
- Public Domain (freie kostenlose, ungeschützte Software)
- Free Software (freie Software, Open Source geschützt)
- Shareware (bezahlbar nach einer Testphase)

## 5.1.2 Software aus der Sicht von SEs beschreiben

Nach Art des Auftraggebers

### Standardsoftware

- Funktionsbezogene Standardsoftware (CAD, Buchhaltungssoftware, PPS)
- Funktionsübergreifende Standardsoftware (ERP, Office)
- Branchensoftware

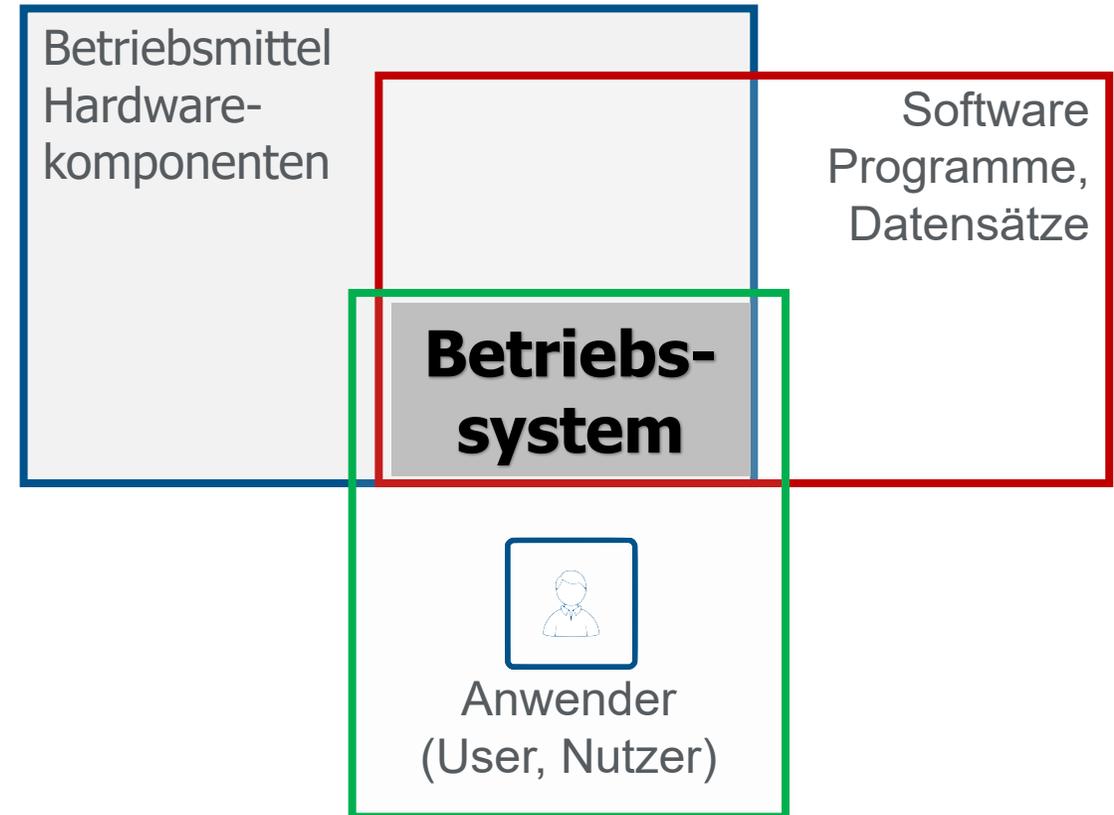
### Individualsoftware

Für einen bestimmten Anwender gefertigte Anwendungssoftware, die sich dadurch auszeichnet, dass sie maßgeschneidert gemäß den Anforderungen eines einzelnen Auftraggebers erstellt wird. Sie ist leicht anpassbar und integrierbar

## 5.1.2 Software aus der Sicht von SEs beschreiben

### Betriebssystem

Das Betriebssystem als Vermittler zwischen Hardware, Software und Anwender



# Kompetenzcheck



Welche Aussagen sind richtig?

- a) Neben Programmen gehören auch Daten zur Software.
- b) Windows 10 fällt in den Bereich der Anwendersoftware.
- c) Betriebssysteme sind Software.
- d) Linux ist ein Betriebssystem.
- e) Computerspiele sind Bestandteil der Systemsoftware.
- f) Standardsoftware ist beim Kauf schon speziell an die Bedürfnisse des Kunden angepasst.
- g) Für die Erstellung von Software muss der Entwickler auch grundlegende fachliche Kenntnisse in dem Bereich besitzen, für den die Software entwickelt wird.
- h) Die Erstellung von Individualsoftware ist mit höheren Kosten verbunden als der Kauf von Standardsoftware.

# Kompetenzcheck

Finden und diskutieren Sie jeweils 3 konkrete Beispiele für System- und Anwendungssoftware.

Überlegen und notieren Sie sich mindestens vier Beispiele für Standardsoftware, welche Sie persönlich einsetzen.



# System-/Anwendungssoftware

## Systemsoftware

Allgemeine Gebrauchssoftware, die grundlegende Systemressourcen und –prozesse verwaltet

In Low-Level-Assembler-Sprache oder Maschinencode geschrieben

Muss speziellen Hardwareanforderungen abdecken: interagiert eng mit der Hardware

Wird gleichzeitig mit dem Betriebssystem installiert, in der Regel vom Hersteller

Läuft immer, wenn der Computer angeschaltet ist

Arbeitet im Hintergrund und Benutzer haben in der Regel keinen Zugriff darauf

Läuft unabhängig

Ist notwendig, damit das System funktioniert

## Anwendungssoftware

Software, die spezielle Aufgaben erfüllt, um Nutzeranforderungen abzudecken

In High-Level-Sprachen geschrieben, zum Beispiel Python oder Java-Script

Berücksichtigt nicht direkt die Hardware und interagiert nicht direkt mit der Hardware

Benutzer oder Admin installiert die Software, wenn sie benötigt wird

Benutzer startet und beendet das Programm

Läuft im Vordergrund, Benutzer arbeiten direkt mit der Software, um spezifische Aufgaben zu lösen

Benötigt die Systemsoftware, um laufen zu können

Wird nicht für das Funktionieren des Systems benötigt

## 5.1.2 Softwarearten unterscheiden

Aufgabe

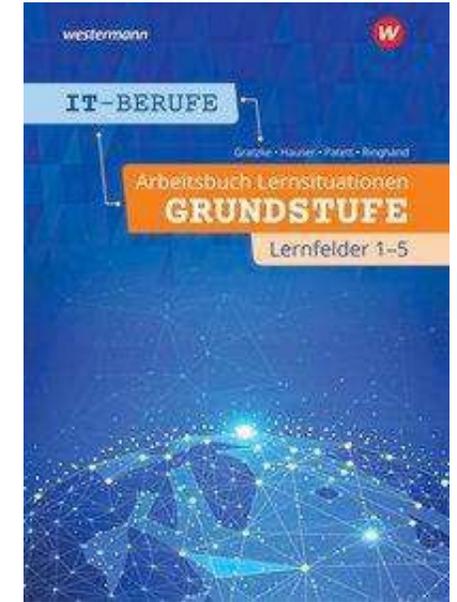


### Aufgabe:

- Bearbeiten Sie im Arbeitsbuch Lernsituation 1, Aufgabe 2

### Optional:

- Unterscheiden Sie Hardware von Software.





## Gruppenarbeit:

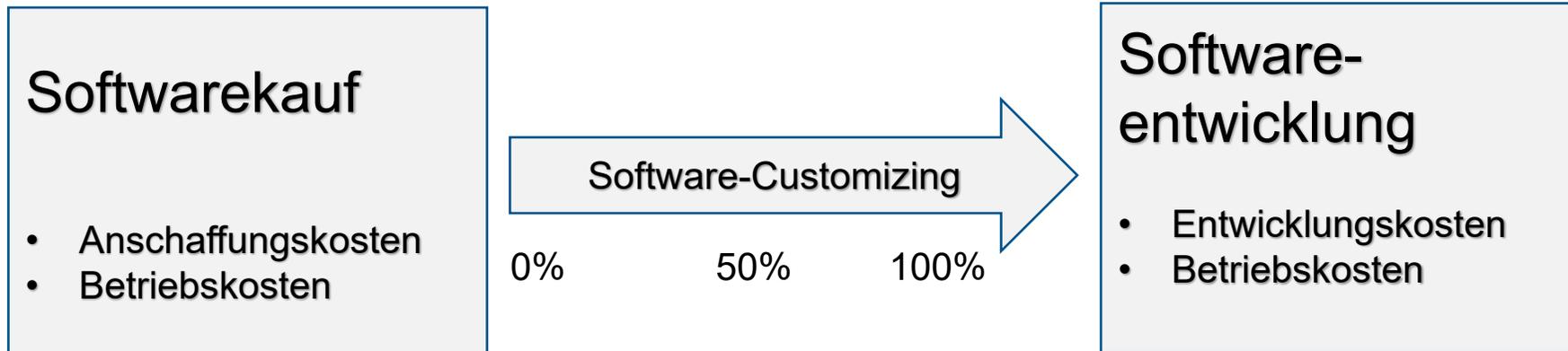
- Individualsoftware vs. Standardsoftware
- Wie kann aus Individualsoftware eine Branchensoftware werden?
- Welche Grundfunktion muss ein Betriebssystem erfüllen?
- Was sind Dienstprogramme und bei welchen der folgenden Beispiele handelt es sich um Dienstprogramme?
  - Textverarbeitung
  - Taskmanager
  - Registrierungs-Editor
  - Freecell

# 5.1.3 Neuentwicklung und Anpassung von Software unterscheiden

- Customizing

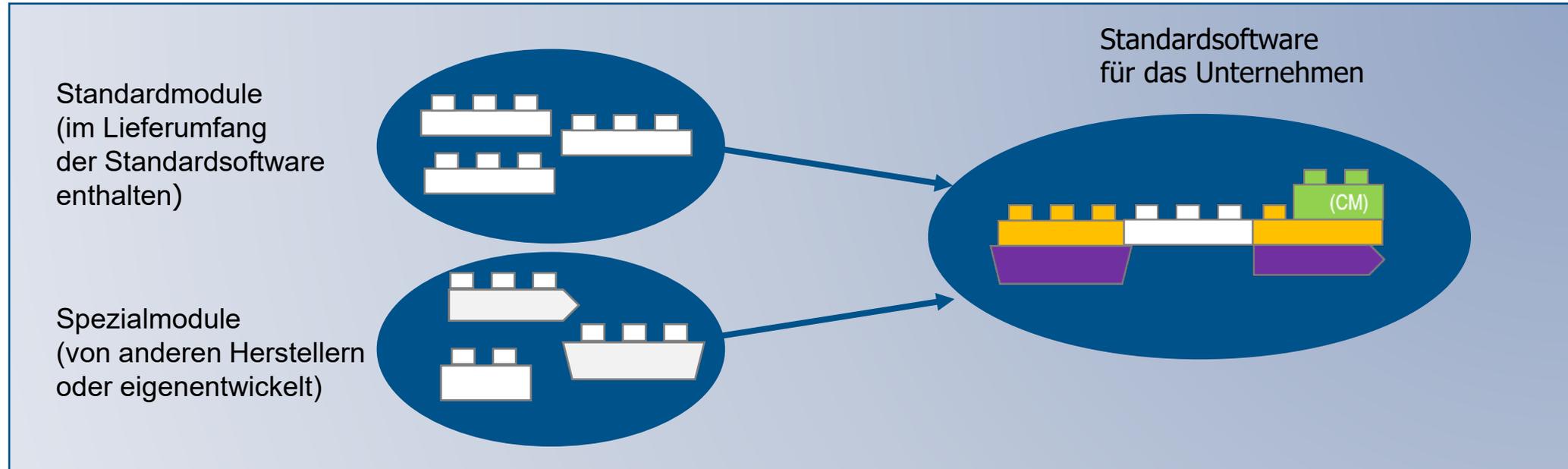
- Konfiguration

- Einstellen von Stammdaten, länderspezifischen Einstellungen (Sprache, Währung), Integration des Corporate Design in Listen und Formulare
- Einstellung der Oberflächen (Farben, Menüs)
- Einrichtung von Schnittstellen und Anpassung und Anlegen von Datenbanken
- Freischalten der vom Hersteller entwickelten Module



# 5.1.3 Neuentwicklung und Anpassung von Software unterscheiden

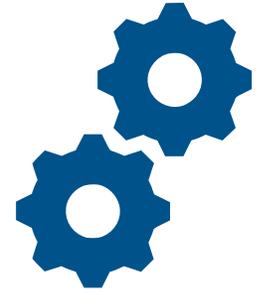
- Customizing
  - Konfiguration



# 5.1.3 Neuentwicklung und Anpassung von Software unterscheiden

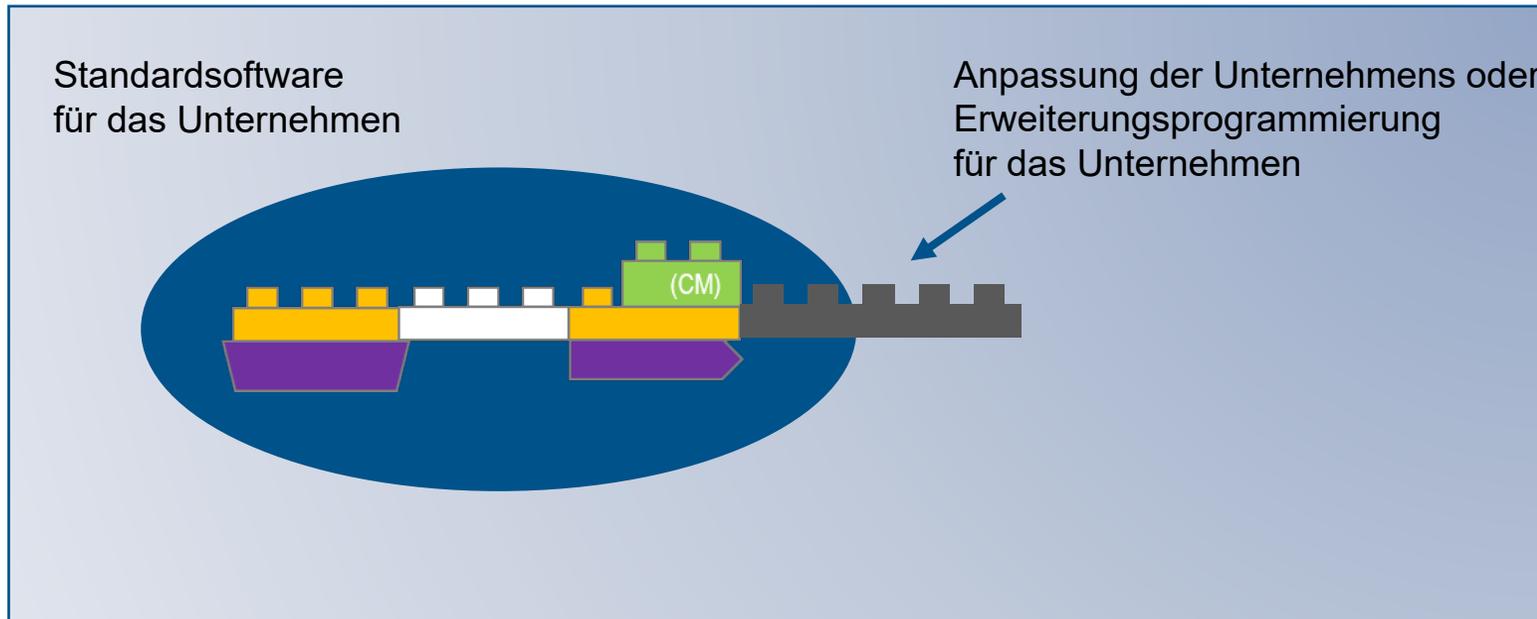
## ○ Customizing

- Parametrisierung (Anpassung)
  - Anpassung von Funktionen (Parameter einstellen, Berechnungen, Abläufe) beispielsweise mit Makrosprachen
  - Rollen und Berechtigungen definieren
  - Benutzer anlegen und Rollen zuweisen
  - Schnittstellen konfigurieren
  - Warnungen festlegen und Alarmgrenzen bestimmen
  - Datenbank-Parameter einschließlich Passwörter eingeben
  - Aktivierung bzw. Deaktivierung von Teilen der Standardsoftware



# 5.1.3 Neuentwicklung und Anpassung von Software unterscheiden

- Customizing
  - Konfiguration
  - Parametrisierung
- Erweiterungsprogrammierung



## 5.1.3 Neuentwicklung und Anpassung von Software unterscheiden

- Customizing
  - Konfiguration
  - Parametrisierung
- Erweiterungsprogrammierung
- **Neuentwicklung**
  - Komplet neu als Individualsoftware entwickeln
- **Datenmigration**
  - (aus lateinisch *migratio*, „Übersiedlung“) steht für den Prozess einer grundlegenden Veränderung einer bestehenden Systemlandschaft oder eines wesentlichen Teils dieser Landschaft

Die Migration kann sich sowohl auf Hard- als auch auf Software beziehen



## 5.1.3 Neuentwicklung und Anpassung von Software unterscheiden

- Customizing zu unterschiedlichen Zeiten notwendig
  - Anpassung der Software vor und nach der Installation (im Installationsprozess)
  - Pflege und Wartung der Software durch Datenkorrekturen, Datenanpassungen, Einhaltung der Datenintegrität, Funktionsanpassungen, Einrichtung neuer Arbeitsplätze, ... während der Produktivphase der Software
  - Schnittstellen-Programmierung, Datenaufbereitung und Datenübernahme bei einer möglichen Ablösung der Software

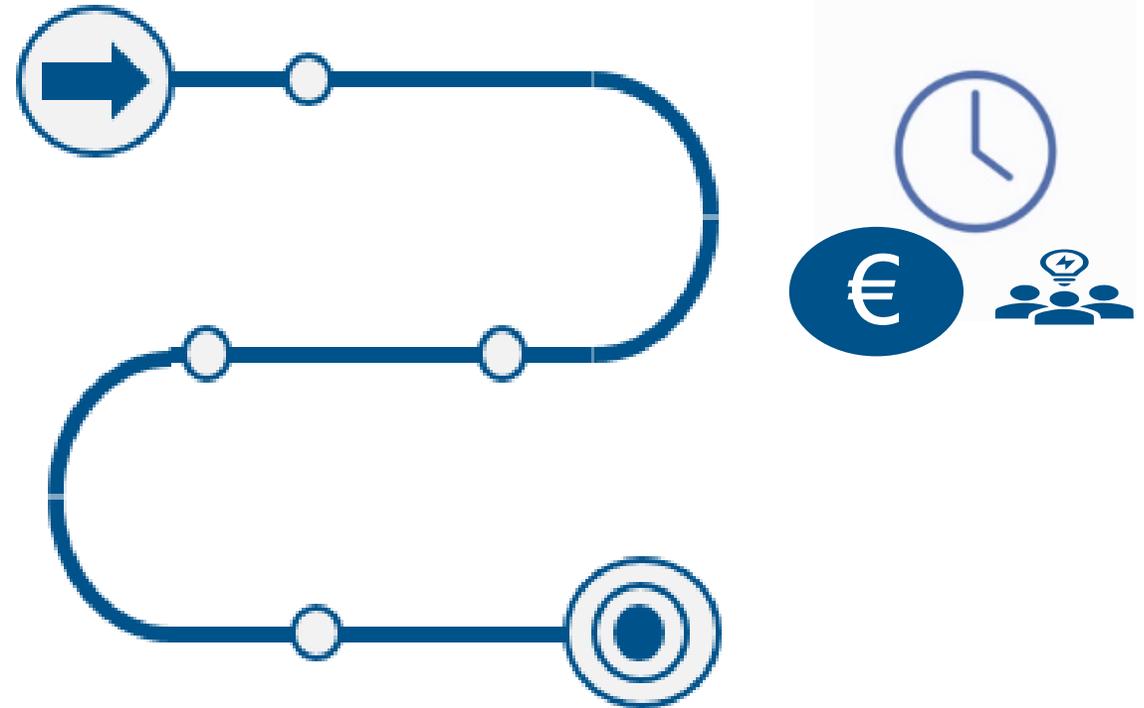


## 5.1.4 In Softwareprojekten mitarbeiten

Merkmale eines Projekts:

Solange man „einfach mal an etwas arbeitet“, ist es noch lange kein **Projekt**.

- Zielvorgabe
- Zeitliche Begrenzung
- Begrenzte Ressourcen
- Projektspezifische Organisationsform
- Neuartigkeit & Einmaligkeit
- Komplexität

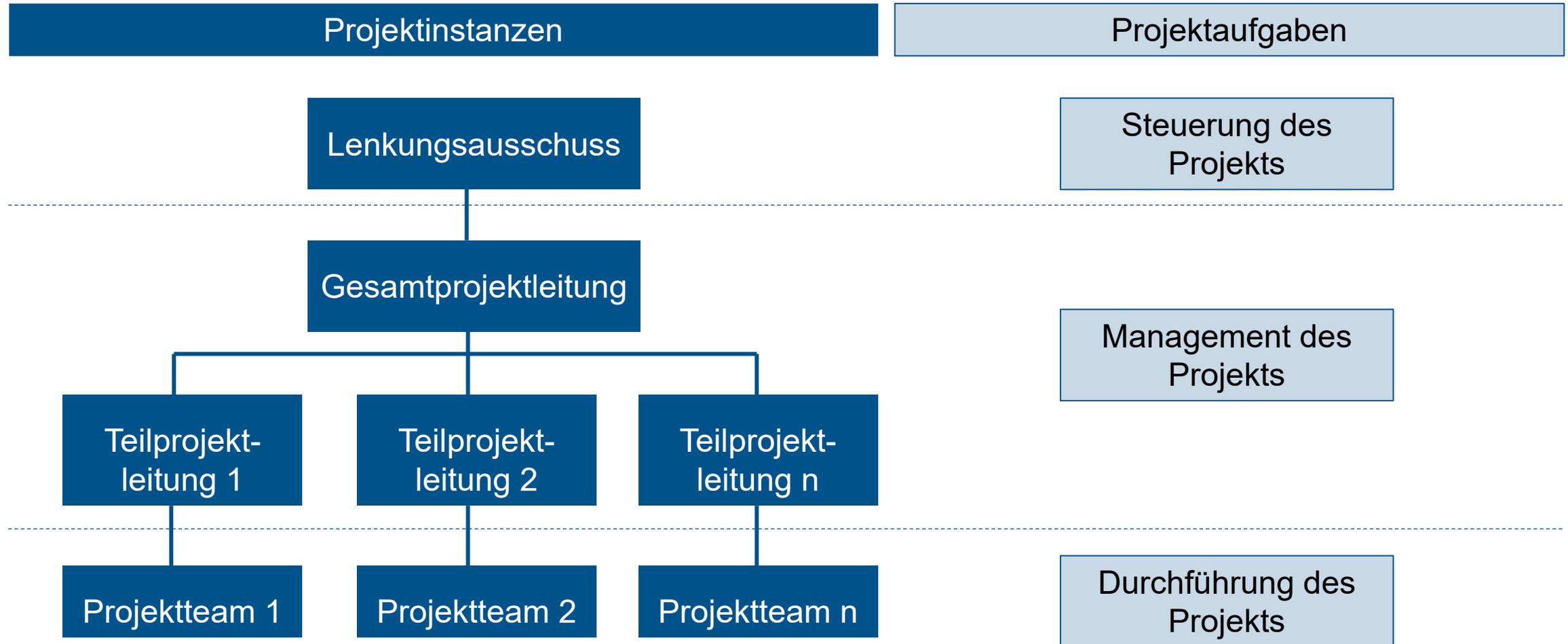


(Lernfeld Projektmanagement – LF-F4)

## 5.1.4 Einteilungen von Softwareprojekten

Einteilung nach	Beispiele
Projektträger	Unternehmensprojekte, Projekte öffentlicher Einrichtungen usw.
Zu erstellende Leistung	Datenbankprojekte, Spieleprojekte usw.
Auftraggeber	Innerhalb des Unternehmens (interne Projekte), externe Projekte
Komplexität	Kleinprojekte, mittlere Projekte, Großprojekte
Anlass	Neuentwicklung, Anpassungs-, Migrationsprojekte

# 5.1.4 In Softwareprojekten mitarbeiten





## 5.1.4 In Softwareprojekten mitarbeiten

### Projektstart:

- Entscheider bewerten den Bedarf und Nutzen
- Initiierung des Projekts
- Benennung der Projektleiter
- Benennung des Projektteams
- Projektauftrag liegt vor
  - Ziele und Inhalte sind beschrieben
- Start mit dem Kick-off-Meeting



# 5.1.4 In Softwareprojekten mitarbeiten

## Projektplanung:

- Planung beginnt teilweise schon vor dem eigentlichen Projektstart
- Danach wird die Planung intensiviert
- Planungsgegenstände:
  - Inhalt
  - Zeit
  - Budget
  - Benötigte Ressourcen
  - Qualität
  - Kommunikation
  - Risiko
  - Stakeholder
  - Teilprojekte

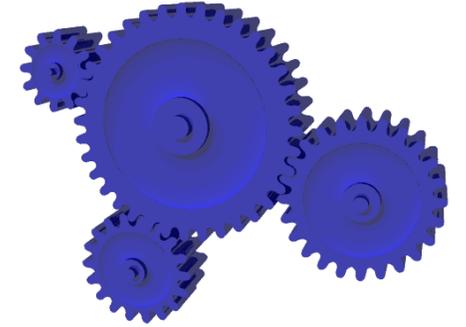


Der Projektleiter und sein Team planen alle Arbeiten, die notwendig sind, um das Projektziel zu erreichen.

## 5.1.4 In Softwareprojekten mitarbeiten

### Projektdurchführung:

- Planung ist weitestgehend abgeschlossen
- Realisierung in kleinen Teams
- Anpassungswünsche
  - Machbarkeitsprüfung, Integrationsmöglichkeit ins laufende Projekt
  - Planung
  - Freigabe
- Kommunikation
  - zwischen den Teams
  - zur Projektleitung
- Regelmäßige Meetings



## 5.1.4 In Softwareprojekten mitarbeiten

### Projektkontrolle:

- Erfolgt parallel zur Durchführungsphase
- Bildet aber gleichzeitig auch eine eigene Phase
- Projektleiter steuert und überwacht alle relevanten Projektaktivitäten
- Projektleiter
  - kontrolliert u. a. die Budgeteinhaltung
  - achtet auf das termingerechte Abliefern
  - priorisiert Aufgaben
  - aktualisiert den Projektplan
  - löst Ressourcenkonflikte
- Bei umfangreichen Projekten werden auch autonome Kontrollteams gebildet



## 5.1.4 In Softwareprojekten mitarbeiten

### Projektabschluss:

- Übergabe inkl. Unterlagen (bspw. Kundendokumentation)
- Abnahme durch den Kunden → Projektende
- Interner Projektabschluss:
  - Erstellen eines Abschlussberichtes
  - Evaluieren des Projektablaufes
  - Archivieren der Projektunterlagen
  - Wertschätzen der Arbeit des Projektteams und dessen Auflösung



# Kompetenzcheck



Welche Aussagen sind richtig?

- a) Das Arbeiten in Projekten ist der Arbeit an Routineaufgaben sehr ähnlich.
- b) Projekte besitzen einen definierten Anfangs- und Endpunkt.
- c) Projekte lassen sich u. a. nach der Größe und dem Anlass einteilen.
- d) Ein Projekt kann aus mehreren Teilprojekten bestehen.
- e) Das Arbeiten an Projekten kann mehrere Tätigkeiten umfassen.
- f) Für das Projektmanagement ist in der Regel die Projektleitung zuständig.
- g) Für das Projektteam startet das Projekt mit dem Kick-Off-Meeting.



Was ist richtig, was ist falsch?

- h) In größeren Projekten übernehmen die Softwareentwickler spezielle Teilaufgaben, wie die Programmierung der Schnittstellen.
- i) Neben der fachlichen Planung werden u. a. auch auftretende Kosten kalkuliert und das Risiko abgewogen.
- j) Die Kontrolle und Überwachung des Projekts können von speziellen Kontrollteams übernommen werden.
- k) Das Wertschätzen der Arbeit des Projektteams ist nicht besonders wichtig.

# Zusammenfassung – Das Umfeld der Softwareentwicklung analysieren



- Aufgabe und Kompetenzen von Softwareentwicklern beschreiben
  - Analyse von Prozessen, Entwicklung Front-/Backend, Programmierung, Kundenberatung und -betreuung, Planen, Durchführen und Auswertung von Tests, Pflege, Wartung und Dokumentation
  - Soft Skills (persönliche, soziale und methodische Kompetenzen), Geduld, Teamfähigkeit
- Software aus Sicht der Softwareentwickler unterscheiden
  - Systemsoftware, Unterstützungssoftware, Anwendersoftware
  - Standard- und Individualsoftware

# Zusammenfassung – Das Umfeld der Softwareentwicklung analysieren



- Neuentwicklung und Anpassung unterscheiden
  - Customizing
    - Konfiguration
    - Parametrisierung
  - Erweiterungsprogrammierung
  - Neuentwicklung
  - Datenmigration

# Zusammenfassung – Das Umfeld der Softwareentwicklung analysieren



- In Softwareprojekten mitarbeiten
  - Projektinstanzen/Projektaufgaben
  - Projektphasen
    - Projektstart
    - Projektplanung
    - Projektdurchführung
    - Projektkontrolle
    - Projektabschluss

# Das Umfeld der Softwareentwicklung analysieren



IT-Berufe  
Grundstufe 1 - 5

Westermann  
**Kapitel 5.1**