

# SOFTWARE- ENTWICKLUNG UND PROJEKTMANAGEMENT

---

3.JG. | 2024/25

Kompetenz: Projektmanagement

Zusammengestellt: Prof. Ing. Bernhard W. Ratschiner, BSc, MSc



## 2 INHALT

---

- Projekt
- Projektarten
- Softwareprojekt
- Unterschiede zu allgemeinen Projekten
- Die 4 Phasen des Projekts

# 3 INHALT

---

- Projektmanagement
- Geschichte des SWP
- Werkzeuge des SWP
- Projektteam des SWP
- Dokumente im SWP
- Vorgehensmodell: Wasserfallmodell

## 4 PROJEKT

---

**Projekt** = Vorhaben, das sich auszeichnet durch die Einmaligkeit der Bedingungen in ihrer Gesamtheit (DIN 69901). Es ist ein Unternehmen auf Zeit.

- Zielvorgabe
- Zeitliche, finanzielle, personelle, usw. Bedingungen
- Abgrenzung gegenüber anderen Vorhaben
- Projektspezifische Organisation (Organigramm)
- Start- und Endtermin

# 5 PROJEKT

---

**Projekt** = Vorhaben, das sich auszeichnet durch die Einmaligkeit der Bedingungen in ihrer Gesamtheit (DIN 69901). Es ist ein Unternehmen auf Zeit.

- Bereichsübergreifend, interdisziplinär
- Einzigartigkeit der Aufgabe
- Ein gewisser Neuigkeitsgrad (Technik, Verfahren, ...)
- Gewisse Komplexität der technischen Lösung
- Personifizierte Verantwortung



# 6 PROJEKT

---

Die 3 Säulen (Ziele) des Projekts

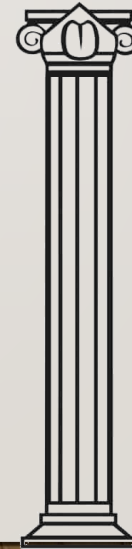
Qualität und  
Funktionalität



Kosten



Zeit



# 7 PROJEKTARTEN

---

## Auftrags-/Abwicklungsprojekt

- Beispiele: Bau-, Anlagen-, Maschinenprojekte
- Klar vorgegebene Lösung bzw. Ergebnis (z.B. Montage, Inbetriebnahme, Hausbau, ...),  
Ergebnisfokussiert
- Wichtig: Termin-, Kosten- und Vertragsmanagement

## 8 PROJEKTARTEN

---

### Organisationsprojekt

- Beispiele: DV-Projekt / IT-Projekt (z.B. Softwareeinführung), Change-Projekte
- Überführung von Zustand A in Zustand B, Mensch und Verhalten als Teil der Lösung
- Wichtig: Einfühlungsvermögen, Konsens



## 9 PROJEKTARTEN

---

### Innovations-/Entwicklungsprojekt

- Beispiele: Produktentwicklung, Softwareentwicklung
- Neuartigkeit/Innovation
- Lösung: unbekannt
- Wichtig: Kreativität

# 10 SOFTWAREPROJEKT

---

**Grundsätzlich:** Ein Projekt, in dem hauptsächlich oder ausschließlich Software entwickelt wird.

Sie unterscheiden sich durch:

**Anwendung, z.B.:**

- Kaufmännische Anwendung (z.B. SAP) oder Technische Anwendung (z.B. CAD-Programm)
- Echtzeitanwendung (z.B. Skype, Motorsteuerung)
- Internetapplikation (z.B. Webseiten, Cloudsysteme, ...)

# II SOFTWAREPROJEKT

---

**Grundsätzlich:** Ein Projekt, in dem hauptsächlich oder ausschließlich Software entwickelt wird.

Sie unterscheiden sich durch:

**Projektart, z.B.:**

- Neuentwicklung oder Weiterentwicklung
- Wartung oder Prozessanalyse
- Integration COTS („commercial off-the-shelf“ z.B. MS Office, Warenwirtschaftssysteme, ...)

# I2 SOFTWAREPROJEKT

---

## **Wesentlicher Einflussfaktor: Kunde**

Auftragsprojekt (Auftragsgebunden)

- Kunde: extern, spezifisch
- Vertraglich geregelte Entwicklung
- Beispiel: Steuerung einer Müllverbrennungsanlage, Webseite

# I3 SOFTWAREPROJEKT

---

## **Wesentlicher Einflussfaktor: Kunde**

Internes Projekt (Auftragsungebunden)

- Kunde: intern
- Bezahlung mit „Scheingeld“ (Buchung auf eine bestimmte Kostenstelle)
- Beispiel: Neue Kalkulationssoftware für Materialkosten, Homepage für die Firma



# 14 SOFTWAREPROJEKT

---

## **Wesentlicher Einflussfaktor: Kunde**

Entwicklungsprojekt für den anonymen Markt (nicht Auftragsgebunden)

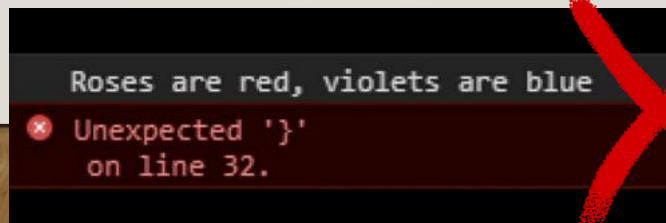
- Kunde: extern, anonym
- Repräsentant des Kunden ist Marketing (Marktforschung vorab)
- Beispiel: MS Office, SAP, PC oder Konsolenspiele

# 15 UNTERSCHIEDE ZU ALLGEMEINEN PROJEKTEN

---

Viele Gemeinsamkeiten mit anderen (Entwicklungs-) Projekten, aber

- Software ist **immateriell**: SW und ihre Eigenschaften sind schwer messbar/beurteilbar, Fortschrittskontrolle ist schwierig
- Ergebnisse/Zwischenergebnisse für IT-Laien oft **schwer beurteilbar**: Externe Fortschrittskontrolle ist schwer → erschwert Kommunikation. Analogie: Skizze für ein Haus = Softwarearchitektur
- Software ist **nicht „stetig“**: Geringe Änderungen haben oft gravierende Auswirkungen



# 16 UNTERSCHIEDE ZU ALLGEMEINEN PROJEKTEN

---

- **Umfeldabhängigkeiten** sind wenig erforscht: Analogie: Hausbau in Europa, Antarktis, Sahara, ... umfeldspezifische Anpassungen der Vorgehensweise sind notwendig
- Wenig gesichertes Wissen über **Vorgehensmodelle**: Welches Modell für welches Projekt in welchem Umfeld
- **Zusammenhang zwischen Anforderungen und Kosten** ist Anwendern schwer vermittelbar: Änderungen, die Anwender als „gering“ einschätzt, haben oft gravierende Konsequenzen (→ Architektur)

# 17 UNTERSCHIEDE ZU ALLGEMEINEN PROJEKTEN

---

- „**Unteilbarkeit**“ der Arbeit bei der Softwareentwicklung: Zeit kann nicht beliebig über mehr „Manpower“ kompensiert werden
- In der Regel mehr **Anforderungsänderungen**: Konsequenz aus schweren (Vorab-) Beurteilbarkeit von Software

# 18 DIE 4 PHASEN DES PROJEKTS

---

Der **Projektlebenszyklus** ist der „Lebensweg“ eines Projekts von dessen Start bis zu dessen Ende und umfasst klassischerweise vier (fünf) Phasen:

- Initiierung
- Planung
- Durchführung + Überwachung
- Abschluss



# 19 PROJEKTMANAGEMENT

---

**Management** = Leitung soziotechnischer Systeme in personen- und sachbezogener Hinsicht mit Hilfe von professionellen Methoden & Werkzeugen (lat.: leiten, zustande bringen, organisieren)

- Sachebene: Bewältigung der Aufgaben (aus den obersten Zielen abgeleitet)
- Personenbezogen: Richtiger Umgang mit Menschen, Führung

**Projektmanagement ist die Gesamtheit aller Führungsaufgaben, -techniken und -mittel für die Abwicklung eines Projekts (nach DIN 69901)**



# 20 PROJEKTMANAGEMENT

---

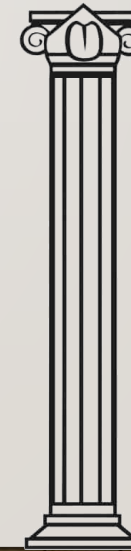
## Grundsätzliche Aufgaben

- Erforderliche Qualität sicherstellen
- Kostenrahmen einhalten
- Termine einhalten

Qualität und  
Funktionalität



Kosten



Zeit



## 21 PROJEKTMANAGEMENT

---

### Grundsätzliche Aufgaben

- Erforderliche Qualität sicherstellen
- Kostenrahmen einhalten
- Termine einhalten

Qualität und  
Funktionalität

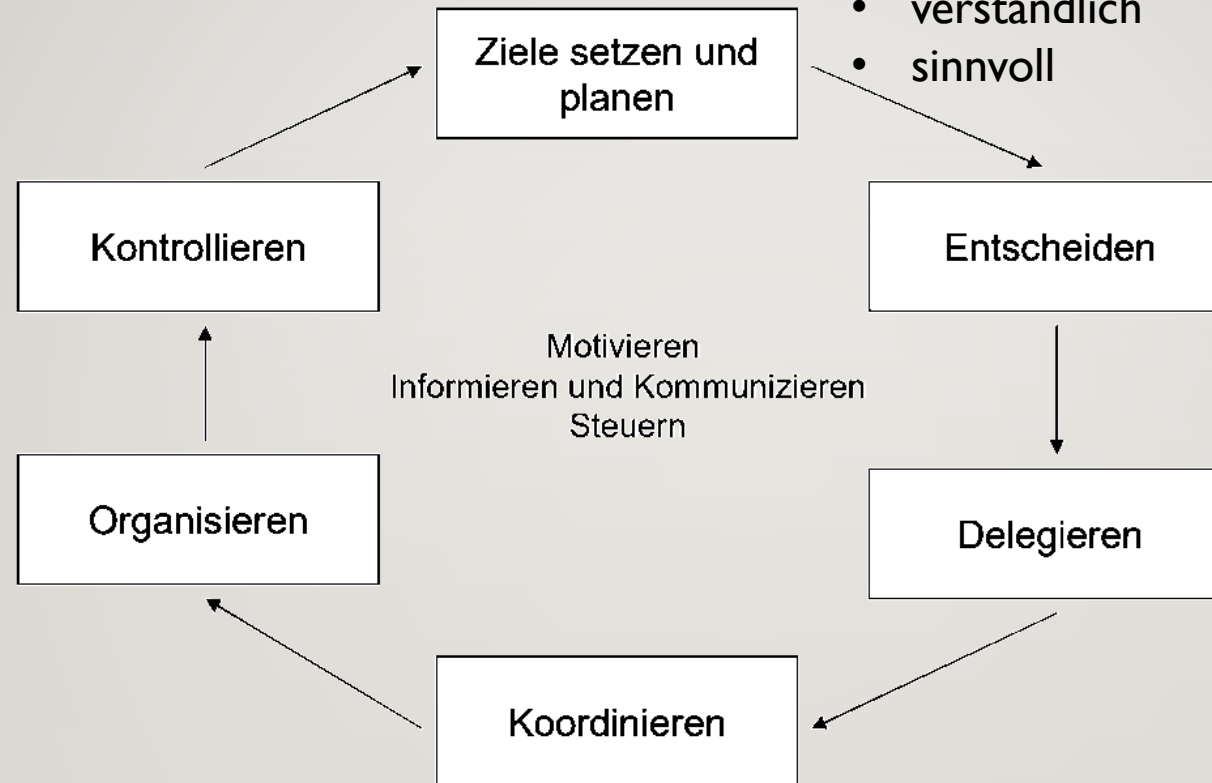
Kosten

Zeit

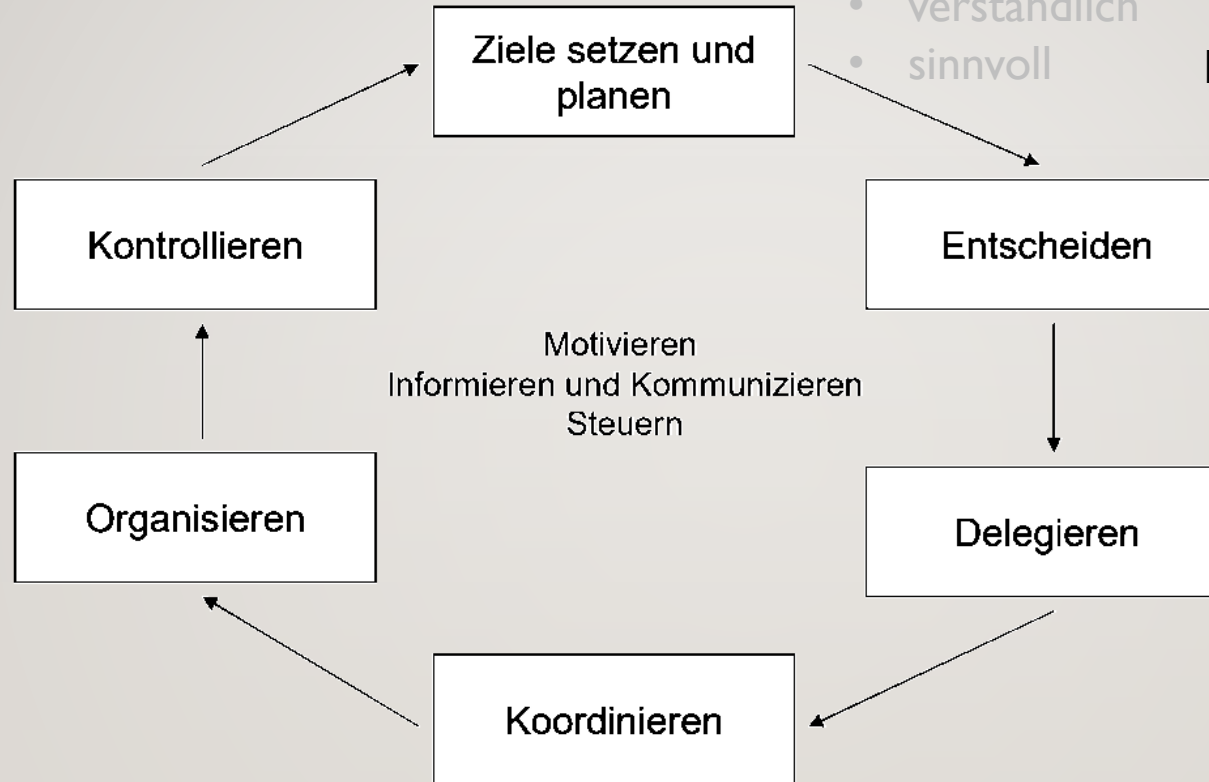


Ziele sind ...

- eindeutig
- widerspruchsfrei
- realistisch
- verständlich
- sinnvoll



- eindeutig
- widerspruchsfrei
- realistisch
- verständlich
- sinnvoll



- Entscheidungen werden nicht getroffen
- Entscheidungen werden zu spät getroffen
- Entscheidungen sind nicht nachvollziehbar

- Entscheidungen werden nicht getroffen
- Entscheidungen werden zu spät getroffen
- Entscheidungen sind nicht nachvollziehbar

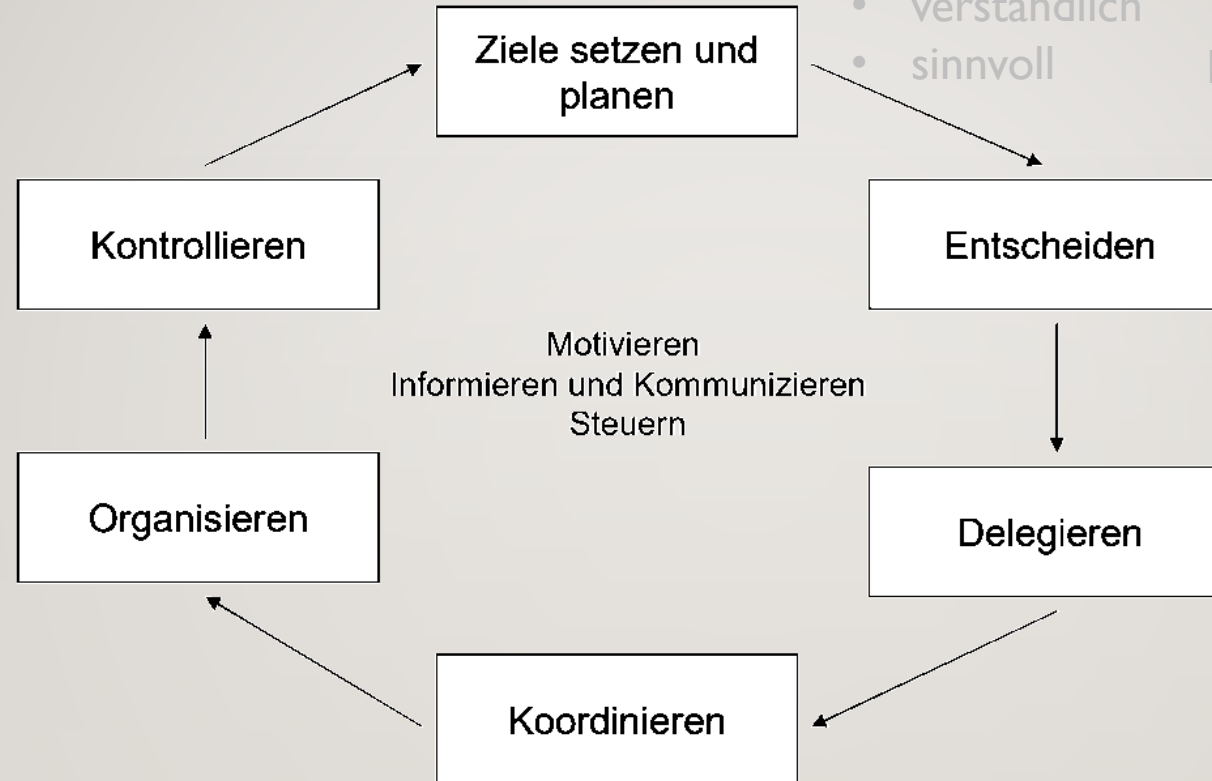


Ziele sind ...

- eindeutig
- widerspruchsfrei
- realistisch
- verständlich
- sinnvoll

Häufige Fehler

- Entscheidungen werden nicht getroffen
- Entscheidungen werden zu spät getroffen
- Entscheidungen sind nicht nachvollziehbar



Wichtig: Gleichgewicht von

- Funktion | Pflichten | Aufgaben
- Verantwortung
- Macht | Befugnisse | Rechte



Ziele sind ...

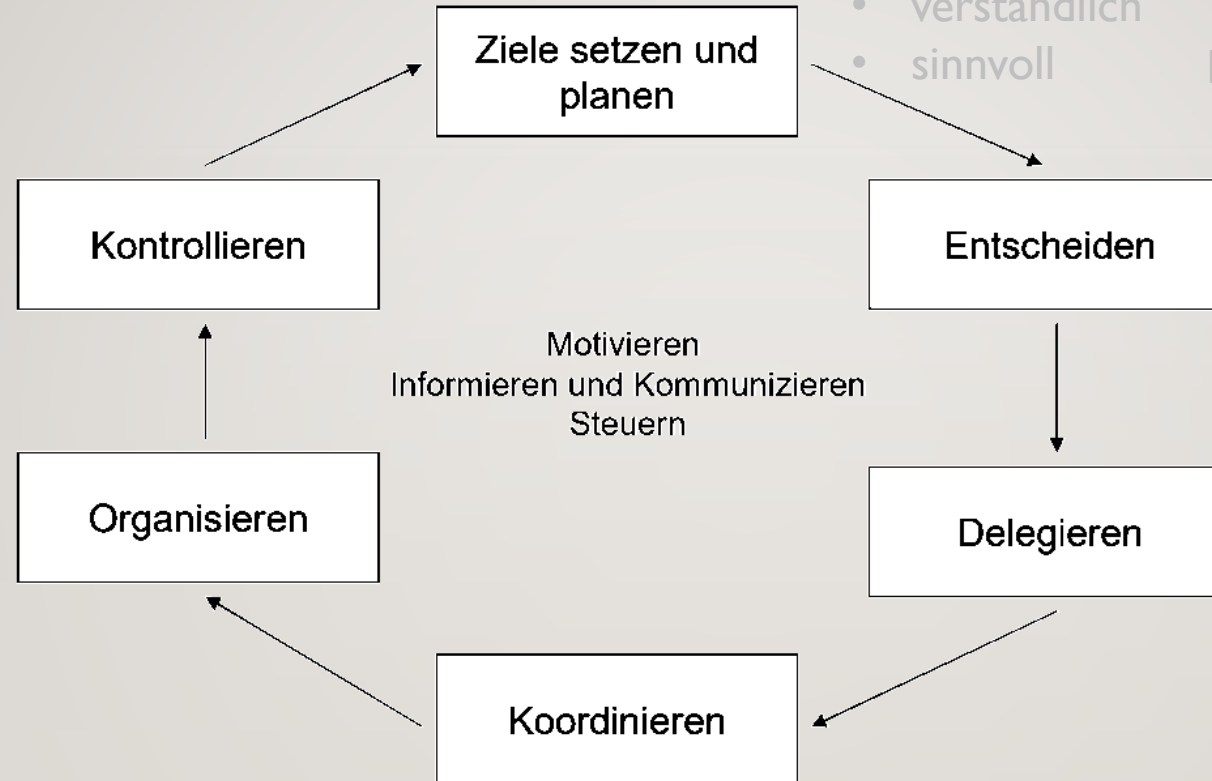
- eindeutig
- widerspruchsfrei
- realistisch
- verständlich
- sinnvoll

Häufige Fehler

- Entscheidungen werden nicht getroffen
- Entscheidungen werden zu spät getroffen
- Entscheidungen sind nicht nachvollziehbar

Wichtig: Gleichgewicht von

- Funktion | Pflichten | Aufgaben
- Verantwortung
- Macht | Befugnisse | Rechte



Abstimmen von Tätigkeiten  
Wichtig! Organisation und Mitarbeiterführung



Ziele sind ...

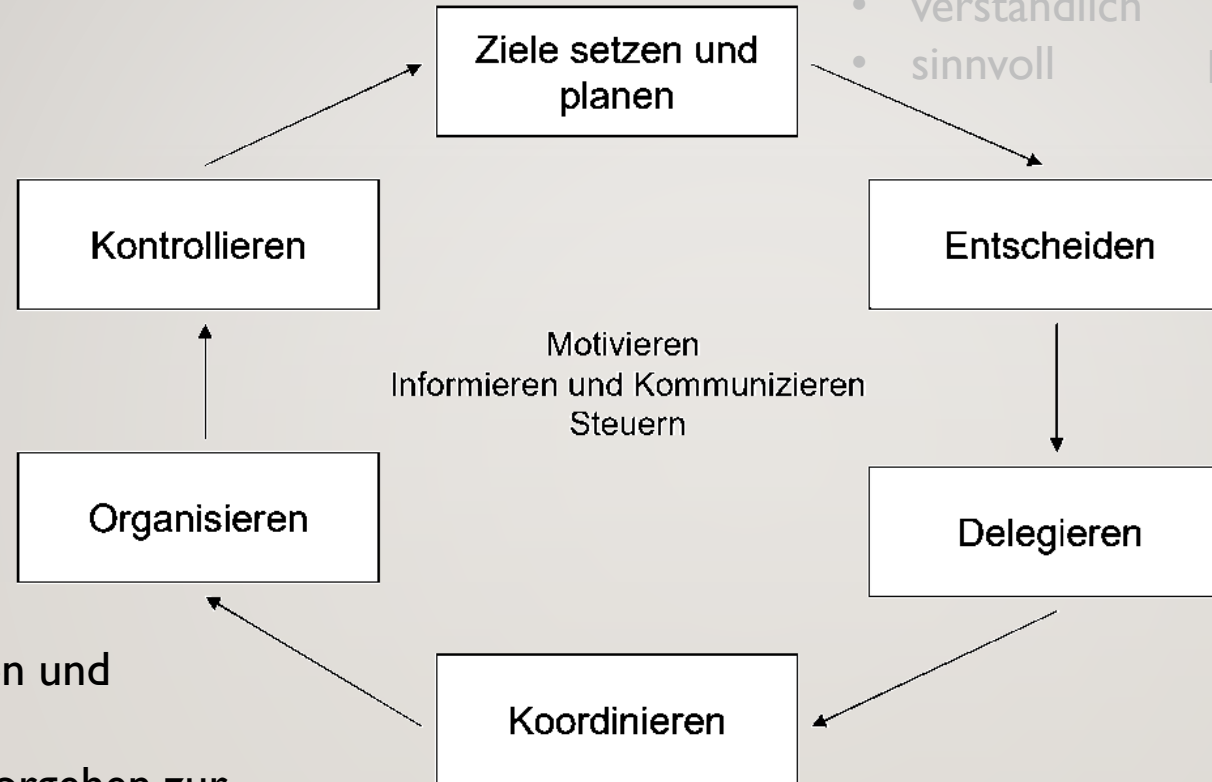
- eindeutig
- widerspruchsfrei
- realistisch
- verständlich
- sinnvoll

Häufige Fehler

- Entscheidungen werden nicht getroffen
- Entscheidungen werden zu spät getroffen
- Entscheidungen sind nicht nachvollziehbar

Wichtig: Gleichgewicht von

- Funktion | Pflichten | Aufgaben
- Verantwortung
- Macht | Befugnisse | Rechte

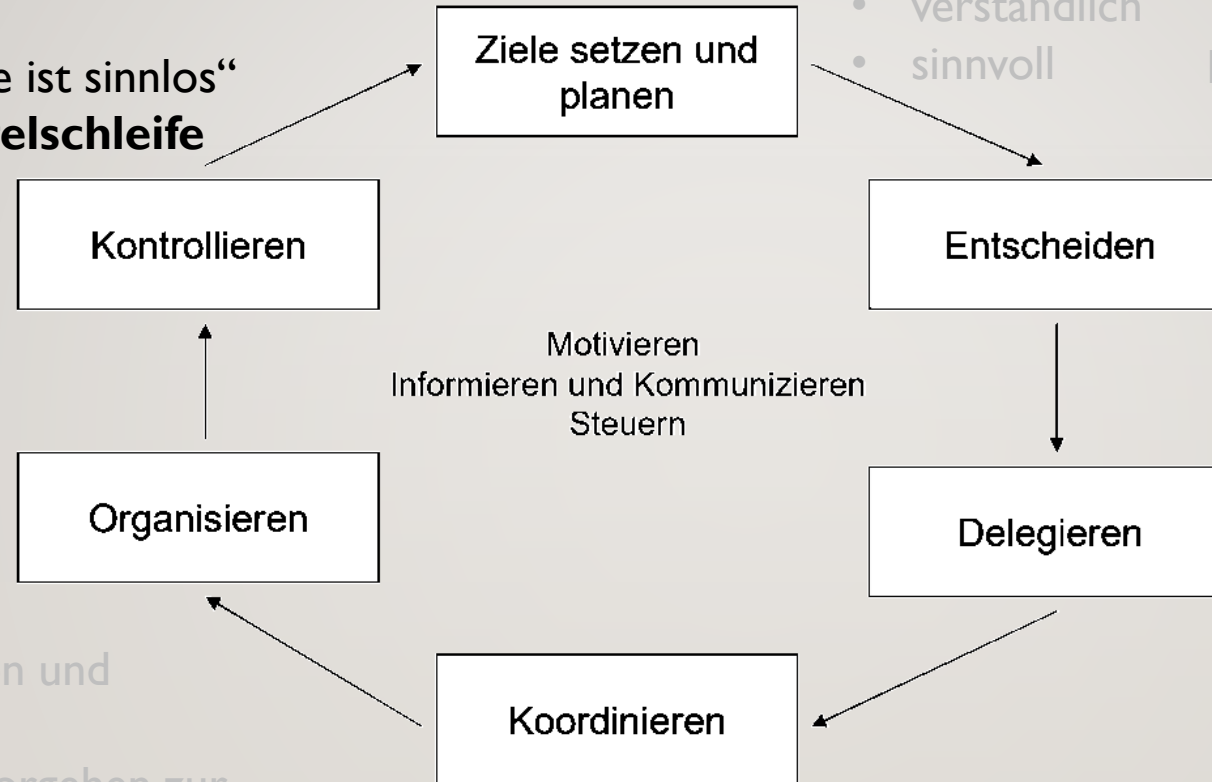


- Festlegen von Abläufen und Zuständigkeiten
- Ablauforganisation: Vorgehen zur Erfüllung bestimmter Funktionen
- Aufbauorganisation: Zuständigkeiten und Gliederung

Abstimmen von Tätigkeiten  
Wichtig! Organisation und Mitarbeiterführung



„Planung ohne Kontrolle ist sinnlos“  
Ein Projekt ist eine **Regelschleife**



Ziele sind ...

- eindeutig
- widerspruchsfrei
- realistisch
- verständlich
- sinnvoll

Häufige Fehler

- Entscheidungen werden nicht getroffen
- Entscheidungen werden zu spät getroffen
- Entscheidungen sind nicht nachvollziehbar

Wichtig: Gleichgewicht von

- Funktion | Pflichten | Aufgaben
- Verantwortung
- Macht | Befugnisse | Rechte

- Festlegen von Abläufen und Zuständigkeiten
- Ablauforganisation: Vorgehen zur Erfüllung bestimmter Funktionen
- Aufbauorganisation: Zuständigkeiten und Gliederung





## 28 PM FÜR SW-PROJEKTE – GESCHICHTE

---

Das **Projektmanagement** für Softwareprojekte hat sich im Laufe der Zeit entwickelt und ist eng mit der wachsenden Komplexität, Größe und Bedeutung von Softwareprojekten verbunden.

**Frühe Jahre (1950er - 1960er Jahre):** In den Anfängen der Computerprogrammierung waren Softwareprojekte oft klein und von begrenzter Komplexität. Zu dieser Zeit wurden Projekte in der Regel von Entwicklern selbst organisiert und geleitet. Es gab keine formalisierten Projektmanagementmethoden.





## 29 PM FÜR SW-PROJEKTE – GESCHICHTE

---

**1970er Jahre:** Mit der zunehmenden Komplexität von Software und der steigenden Nachfrage nach effektiveren Managementtechniken begannen Organisationen, Projektmanagementansätze aus anderen Branchen zu übernehmen und auf Softwareprojekte anzuwenden.

**1970er - 1980er Jahre:** In dieser Zeit wurden verschiedene Ansätze des Projektmanagements wie das Wasserfallmodell entwickelt. Das Wasserfallmodell war eine der ersten formalisierten Projektmanagementmethoden für Softwareentwicklung, bei der der Entwicklungsprozess in lineare, sequentielle Phasen unterteilt wurde



## 30 PM FÜR SW-PROJEKTE – GESCHICHTE

---

**1980er - 1990er Jahre:** Agile Methoden begannen in den 1980er Jahren an Popularität zu gewinnen, vor allem durch Ansätze wie Scrum und Extreme Programming (XP). Diese Methoden waren eine Reaktion auf die Schwächen des Wasserfallmodells und betonten iterative, inkrementelle Entwicklung, enge Zusammenarbeit mit Kunden und die Anpassung an sich ändernde Anforderungen.

## 3 | PM FÜR SW-PROJEKTE – GESCHICHTE

---

**2000er Jahre bis heute:** Agile Methoden haben sich weiterentwickelt und sind mittlerweile weit verbreitet. Es gibt verschiedene agile Ansätze wie Kanban, Lean, und weitere spezialisierte agile Methoden. Darüber hinaus sind hybride Ansätze entstanden, die Elemente aus agilen und traditionellen Projektmanagementmethoden integrieren.



## 32 PM FÜR SW-PROJEKTE – GESCHICHTE

---

Der Bedarf an Projektmanagement für Softwareprojekte ist aufgrund der

- **wachsenden Komplexität** von Softwareanwendungen,
- der Notwendigkeit, **Termine** und **Ressourcen** effizient zu **verwalten** sowie der
- Anforderungen an **Qualität** und **Kundenzufriedenheit**

entstanden. Die Erfahrungen und Lehren aus früheren Projekten führten zur Entwicklung und Verfeinerung von Projektmanagementansätzen, um effektive Softwareentwicklung sicherzustellen.





## 33 PRINZIPIEN UND TECHNIKEN

---

Das Projektmanagement für Softwareprojekte hat einige Prinzipien und Techniken aus dem allgemeinen Projektmanagement übernommen, aber es hat auch spezifische Aspekte entwickelt, um den Anforderungen und Herausforderungen der Softwareentwicklung gerecht zu werden.

Hier sind einige Bereiche, in denen das Projektmanagement für Softwareprojekte vom allgemeinen Projektmanagement abgeleitet oder beeinflusst wurde:





## 34 PRINZIPIEN UND TECHNIKEN

---

### **Planung und Strukturierung:**

- Übernahme von allgemeinen Projektmanagementprinzipien zur effektiven Planung von Aufgaben, Ressourcenallokation, Zeitmanagement und Budgetierung.
- Spezifische Anpassungen für die dynamische Natur der Softwareentwicklung, um flexiblere Planungs- und Anpassungsmechanismen zu ermöglichen.

## 35 PRINZIPIEN UND TECHNIKEN

---

### **Risikomanagement:**

- Integration von Risikomanagementpraktiken aus dem allgemeinen Projektmanagement, um Risiken in Bezug auf Budget, Zeitrahmen, Technologie, Anforderungen und externe Faktoren zu identifizieren, bewerten und behandeln.
- Anpassung an die spezifischen Risiken in der Softwareentwicklung, wie technologische Unsicherheiten, Anforderungsänderungen und externe Abhängigkeiten.

## 36 PRINZIPIEN UND TECHNIKEN

---

### **Kommunikation und Teamarbeit:**

- Anwendung bewährter allgemeiner Kommunikations- und Teamarbeitstechniken, um eine effektive Kommunikation und Zusammenarbeit im Projektteam zu fördern.
- Berücksichtigung von spezifischen Kommunikationsherausforderungen in der Softwareentwicklung, z.B. zwischen technisch versierten Entwicklern und nicht-technischen Stakeholdern.

## 37 PRINZIPIEN UND TECHNIKEN

---

### **Qualitätsmanagement:**

- Übernahme von Qualitätsmanagementansätzen und -techniken, um sicherzustellen, dass die Software den Qualitätsstandards und den Kundenerwartungen entspricht.
- Integration von speziellen Softwaretest- und Qualitätssicherungsverfahren zur Gewährleistung der Softwarequalität.

## 38 PRINZIPIEN UND TECHNIKEN

---

### **Change-Management und Agilität:**

- Einbeziehung von agilen Prinzipien und Praktiken, um auf sich ändernde Anforderungen, technologische Entwicklungen und Kundenfeedback flexibel zu reagieren.
- Implementierung von kontinuierlichen Verbesserungsprozessen, um die Effizienz und Effektivität des Entwicklungsprozesses kontinuierlich zu optimieren
- Betonung von Feedbackschleifen und ständiges Anpassen, um die Software zu verbessern.



## 39 PRINZIPIEN UND TECHNIKEN

---

### **Zusätzlich ...**

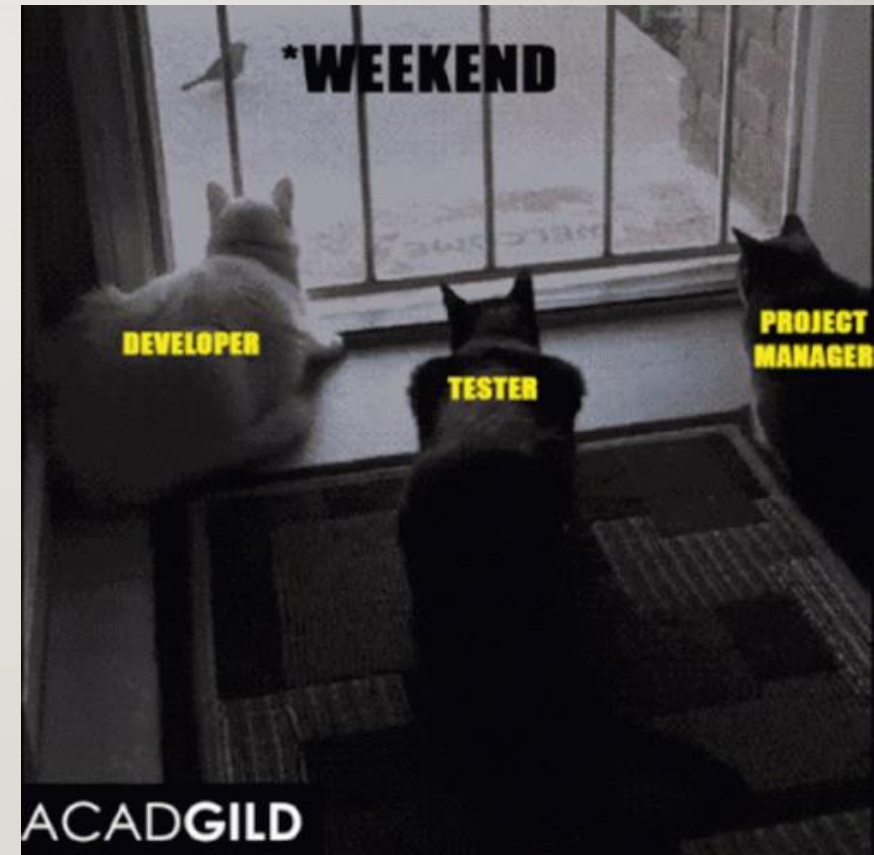
- Integration flexibler Anforderungsmanagementprozesse, die es ermöglichen, Änderungen während des Projekts zu akzeptieren und in den Entwicklungsprozess zu integrieren.
- Einsatz von Prototypen und Mockups, um frühzeitig ein besseres Verständnis der Anforderungen zu erlangen und Missverständnisse zu minimieren und um schließlich die Erwartungen des Kunden zu erfüllen
- Integration von Automatisierung für Tests, um eine effizientere Überprüfung von Code und Funktionen zu ermöglichen.
- Anpassung von Projektmanagementpraktiken an die Skalierbarkeit von Softwareprojekten, sei es ein kleines Team für ein Startup oder ein großes Team für ein umfassendes Unternehmensprojekt.

## 40 PROJEKTTEAM UND ROLLEN

---

Ein typisches Projektteam für ein Softwareprojekt besteht aus verschiedenen Rollen, die zusammenarbeiten, um das Projekt erfolgreich abzuschließen.

Hier sind einige häufige Rollen und ihre Aufgaben aufgezählt.



## 4 | PROJEKTTEAM UND ROLLEN

---

### **Projektleiter/Projektmanager:**

- Verantwortlich für die Gesamtleitung des Projekts.
- Plant, organisiert und überwacht den gesamten Projektverlauf.
- Koordiniert Ressourcen und stellt sicher, dass das Projektziel erreicht wird.



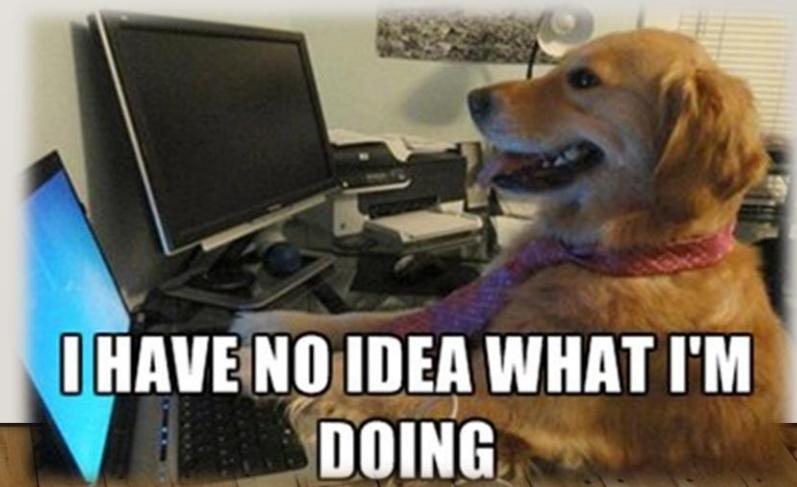


## 42 PROJEKTTEAM UND ROLLEN

---

### **Softwareentwickler/Programmierer:**

- Entwickeln und implementieren den Code gemäß den Anforderungen und Spezifikationen des Projekts.
- Arbeiten eng mit anderen Teammitgliedern zusammen, um Module zu integrieren und sicherzustellen, dass sie korrekt funktionieren

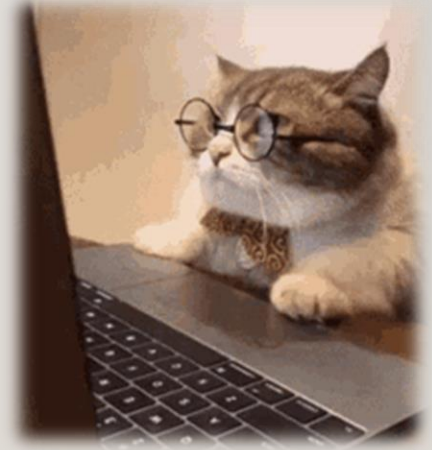


## 43 PROJEKTTEAM UND ROLLEN

---

### **Tester/QA (Qualitätskontrolle):**

- Testen die Software auf Fehler und Unregelmäßigkeiten.
- Erstellen Testfälle und führen manuelle und automatisierte Tests durch, um sicherzustellen, dass die Software den Anforderungen entspricht und fehlerfrei läuft.





## 44 PROJEKTTEAM UND ROLLEN

---

### **UX/UI-Designer:**

- Entwerfen die Benutzeroberfläche und das Benutzererlebnis, um sicherzustellen, dass die Software benutzerfreundlich und ästhetisch ansprechend ist.
- Berücksichtigen die Benutzeranforderungen und -erwartungen, um ein positives Nutzererlebnis zu gewährleisten

UX ... User eXperience



## 45 PROJEKTTEAM UND ROLLEN

---

### **Anforderungsanalyst/Anforderungsmanager/Business Analyst:**

- Sammeln, analysieren und dokumentieren die Anforderungen und spezifizieren die Funktionalitäten der Software.
- Kommunizieren zwischen den Stakeholdern und dem Entwicklungsteam, um sicherzustellen, dass die Anforderungen klar verstanden und umgesetzt werden.



## 46 PROJEKTTEAM UND ROLLEN

---

### Systemarchitekt:

- Entwerfen und definieren die Gesamtstruktur und -architektur der Software.
- Legen fest, wie die verschiedenen Komponenten (Frontend, Backend, Middleware, Datenbank, Module, ...) miteinander interagieren und koordinieren die technischen Aspekte des Projekts.





## 47 PROJEKTTEAM UND ROLLEN

---

### Datenbankadministrator:

- Entwerfen, implementieren und verwalten die Datenbankstruktur.
- Sorgen für die effiziente Speicherung und Verwaltung von Daten innerhalb des Systems



## 48 PROJEKTTEAM UND ROLLEN

---

### **Konfigurationsmanagement** (bei großen Projekten)

- Versionskontrolle (Verwaltung von Quellcode und anderen Entwicklungsartefakten)
- Änderungsmanagement (Verfolgung und Kontrolle von Änderungsanfragen und ihrer Doku)
- Build- und Freigabemanagement (Erstellung von Builds und Freigabe von Versionen)
- Dokumentationsmanagement (Sicherstellen, dass Dokumente mit Code synchron laufen)
- Änderungsverfolgbarkeit (Sicherstellen, dass Änderungen bis zum Ursprung verfolgbar bleiben)
- Sicherheits- und Compliance-Management (Softwarekomponenten entsprechen Compliance)







## 50 DOKUMENTE IM PM VON SWP

---

Im Projektmanagement von Softwareprojekten gibt es verschiedene wichtige Dokumente, die während des gesamten Projektlebenszyklus erstellt, verwaltet und genutzt werden.

Diese Dokumente dienen dazu, den Projektablauf zu planen, den Fortschritt zu überwachen, die Qualität zu sichern, Risiken zu managen und eine erfolgreiche Fertigstellung des Projekts zu gewährleisten.

**Stakeholder:** Person, für die es aufgrund ihrer Interessenlage von Belang ist, wie ein bestimmtes Unternehmen sich verhält (z. B. Aktionär, Mitarbeiter, Kunde, Lieferant)



## 51 DOKUMENTE IM PM VON SWP

---

### **Projektcharter:**

Einrichtungs- und Startdokument des Projekts, das den Projektauftrag, die Ziele, den Umfang, die Stakeholder, Hauptanforderungen, Budget, Zeitplan und Risiken zusammenfasst.

### **Projektplan:**

Detaillierter Plan, der den Projektablauf, die Arbeitspakete, Meilensteine, Zeitpläne, Ressourcenzuweisungen und Budgets umfasst

## 52 DOKUMENTE IM PM VON SWP



## 53 DOKUMENTE IM PM VON SWP

---

Der **Projektplan** enthält üblicherweise:

- Projektzusammenfassung (Übersicht, Projektname, Zielsetzung und Zweck)
- Projektziele und –umfang (Klare Definition der Anforderungen, alle Projektmitglieder benötigen ein gemeinsames Verständnis der Projektziele)
- **Projektstrukturplan** (Work Breakdown Structure, WBS)



## 54 DOKUMENTE IM PM VON SWP

---

### **Projektstrukturplan**

- Die Hauptziele des Projektstrukturplans sind, das Projekt in sinnvolle Einheiten aufzuteilen, die Ecktermine festzulegen und den erforderlichen Aufwand (Kosten) zu ermitteln.
- Der Projektstrukturplan gliedert das Projekt schrittweise über einzelne Ebenen in Strukturelemente. Diese Strukturelemente werden im Projektsystem als Projektstrukturplanelemente (PSP-Elemente) bezeichnet. Jedes PSP-Element ist das Ergebnis einer Aufteilung.

## 55 DOKUMENTE IM PM VON SWP

---

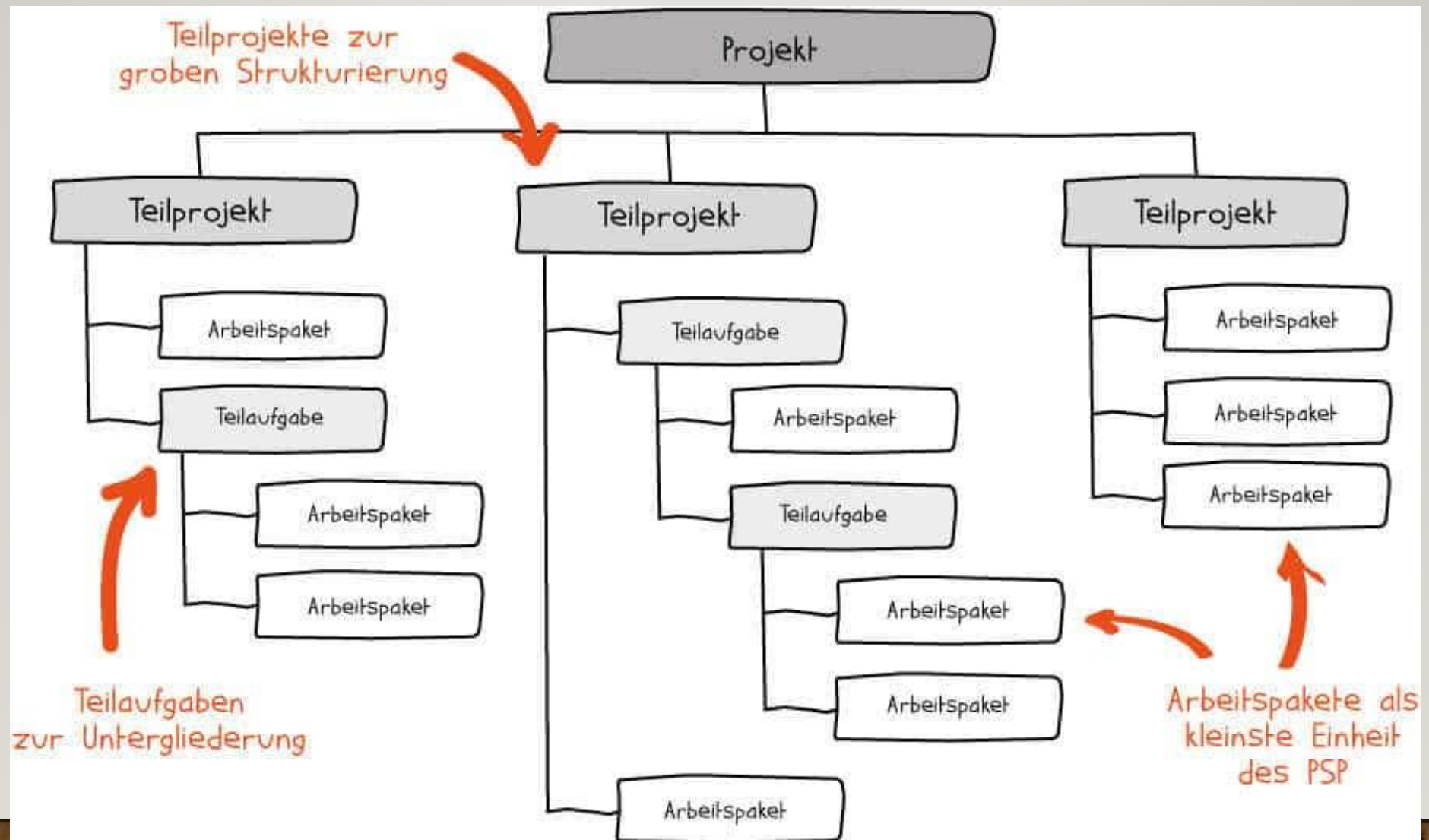
### **Projektstrukturplan (PSP)**

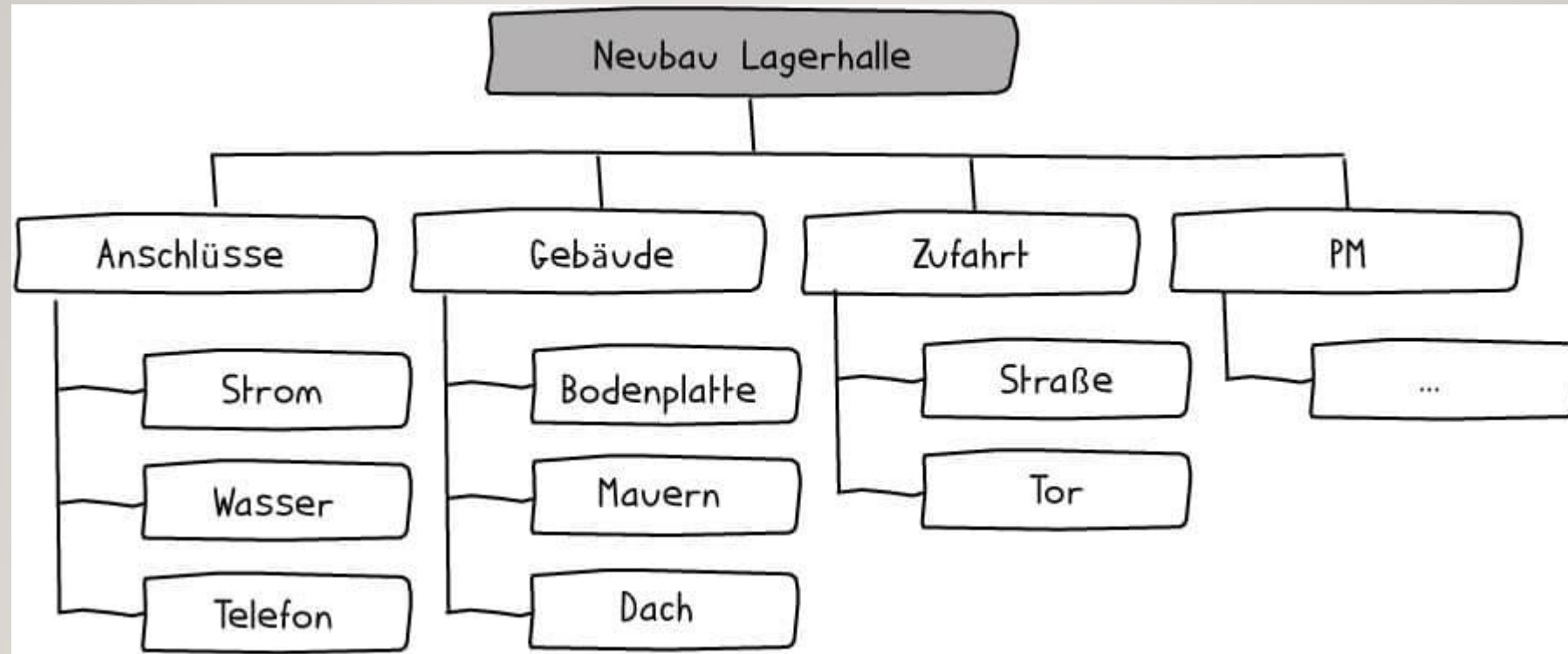
engl.: Work Breakdown Structure (WBS)

Man kann den Projektstrukturplan nach verschiedenen Gesichtspunkten strukturieren:

- nach Phasen (logik- bzw. ablauforientiert)
- nach Funktionen (funktionsorientiert)
- nach Objekten (objektorientiert)

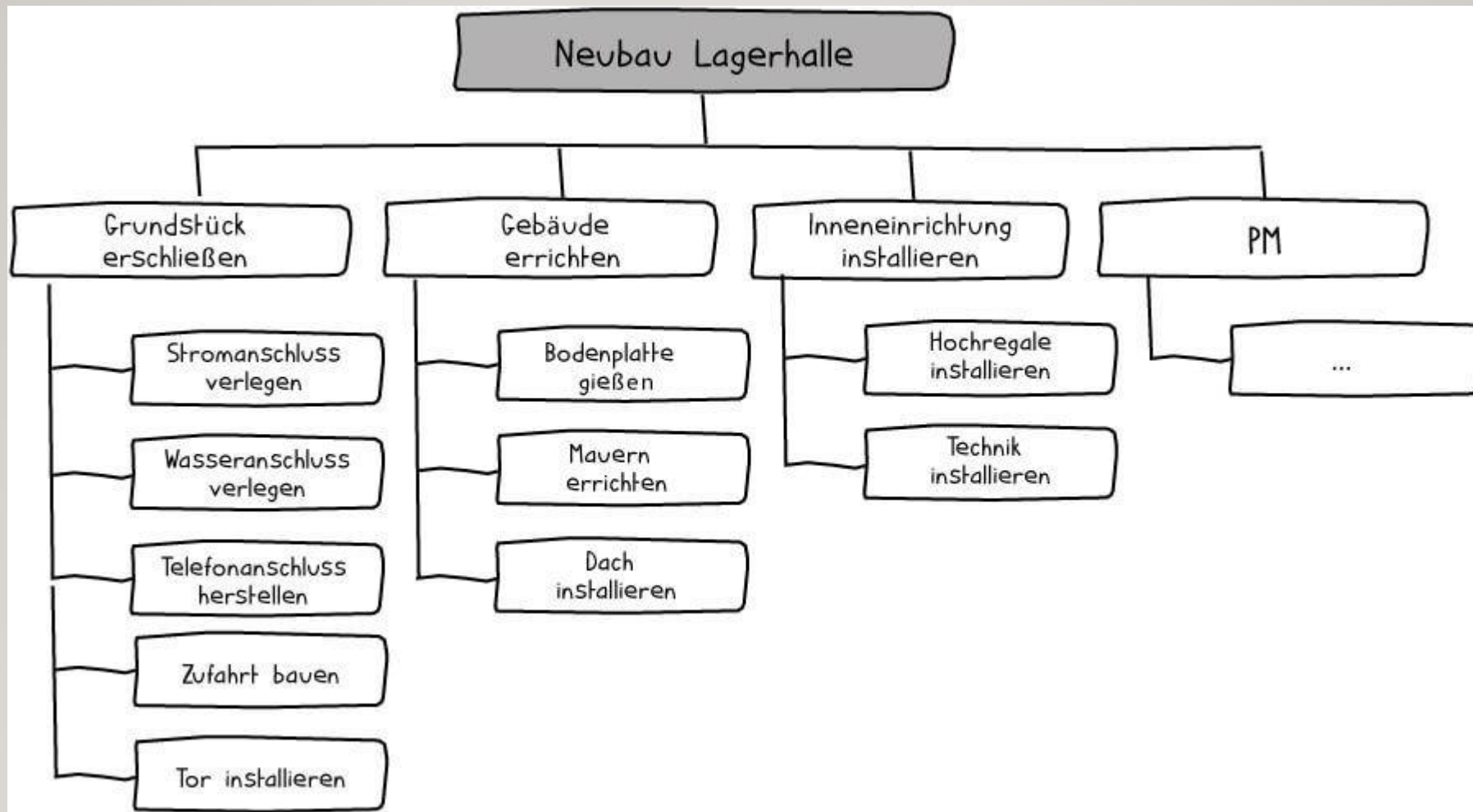
Laut DIN 69901 werden diejenigen Elemente im PSP, die nicht weiter aufgegliedert werden, Arbeitspakete genannt.





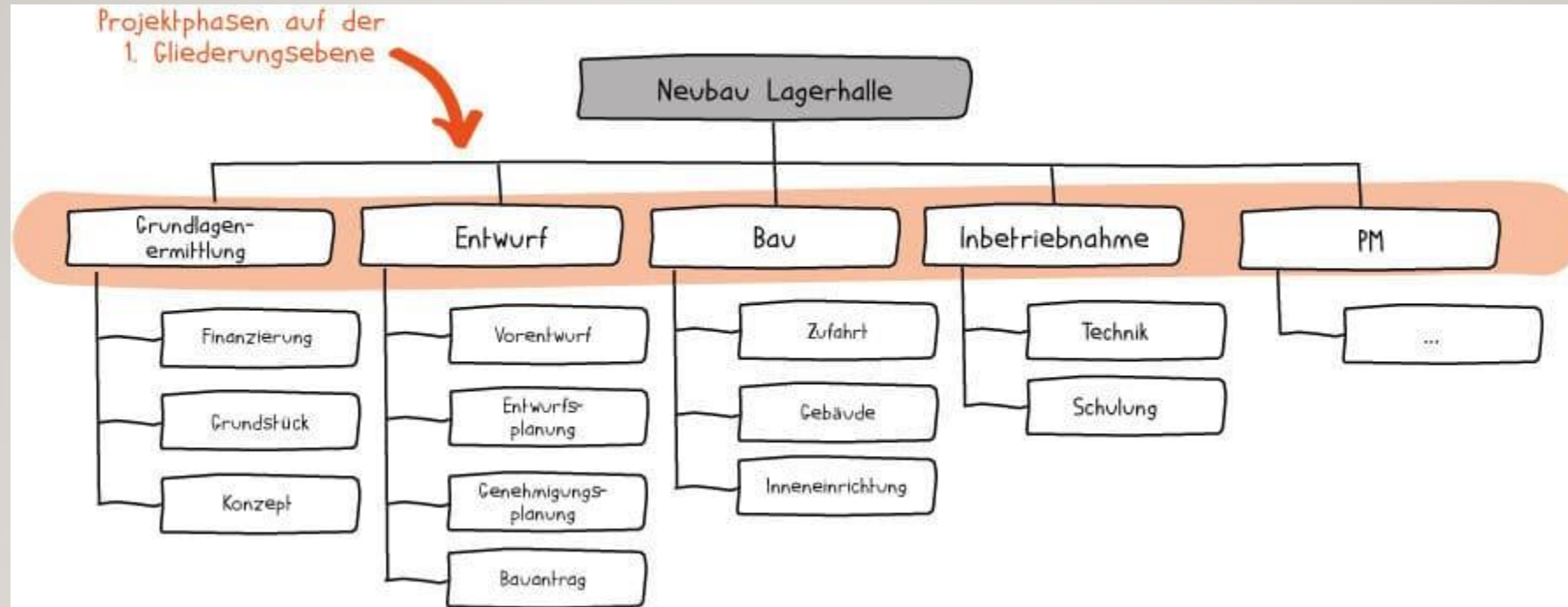
**Objektorientierung:** Ein objektorientierter PSP ist nach konkreten Elementen, Bauteilen oder Objekten gegliedert. Das folgende Beispiel zeigt einen groben PSP für den Neubau einer Lagerhalle.





**Aktivitätsorientierung (manchmal auch Funktionsorientierung):** Im Gegensatz zur Objektorientierung stehen hier die Aktivitäten und Tätigkeiten im Vordergrund.

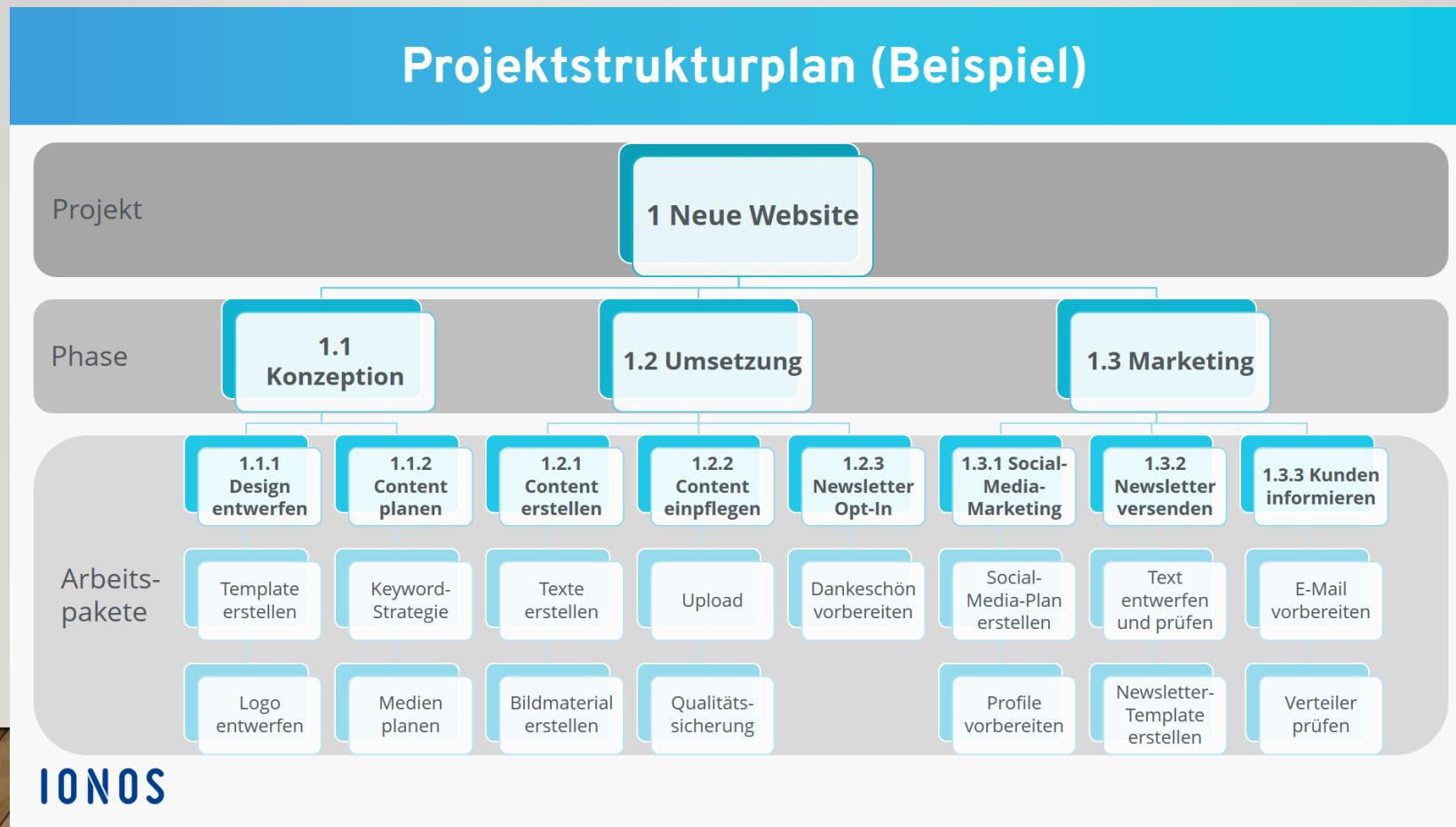




**Phasenorientierung:** Bei phasenorientierten Projektstrukturplänen werden die definierten Projektphasen auf der ersten Gliederungsebene aufgeführt. Alle weiteren Ebenen können dann entweder objekt- oder aktivitätsorientiert gegliedert werden.

# 60 DOKUMENTE IM PM VON SWP

## PSP - Webseite



## 6 | DOKUMENTE IM PM VON SWP

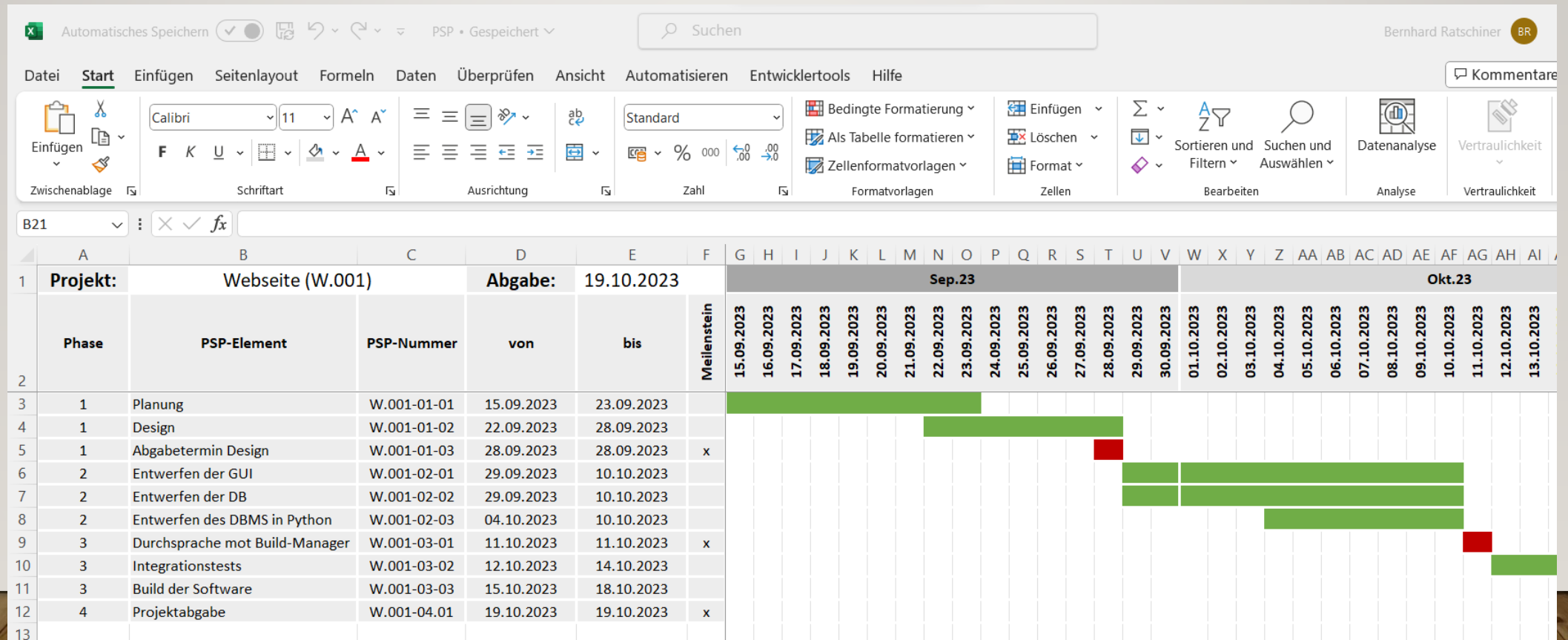
---

Der **Projektplan** enthält üblicherweise:

- Arbeitspakete und Aufgaben (Arbeitspakete in spezifische Aufgaben aufschlüsseln, klare Beschreibungen, Verantwortlichkeiten, Ressourcen und Abhängigkeiten)
- **Zeitplan und Meilensteine** (Start- und Endtermine für jedes Arbeitspaket und jede Aufgabe sowie Definition von Meilensteinen, um den Projektfortschritt zu verfolgen). Dieser wird üblicherweise als **Gantt** Diagramm ausgeführt.

# 62 DOKUMENTE IM PM VON SWP

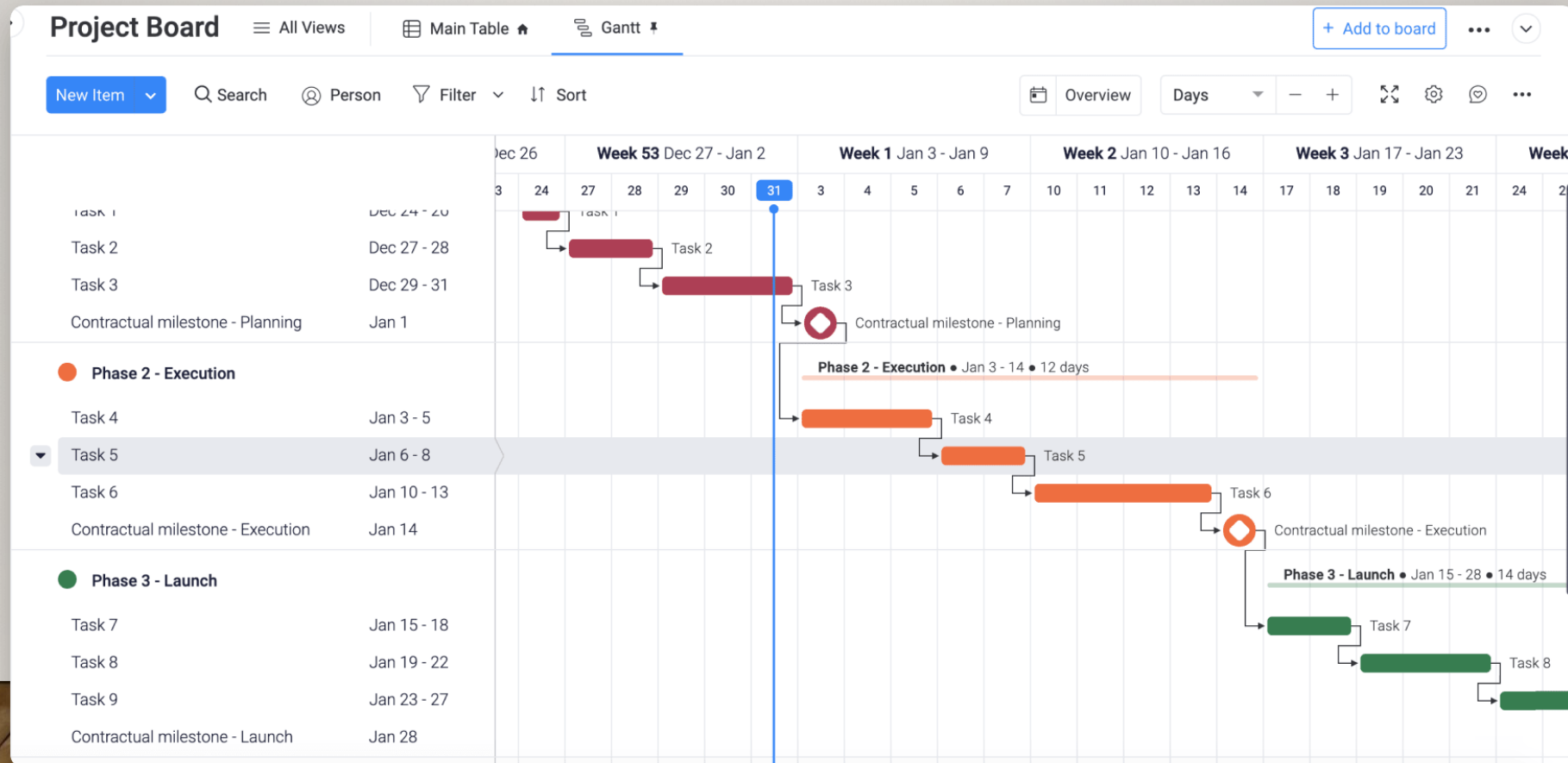
## Gantt Diagramm mit Meilensteinen in MS Excel





## 63 DOKUMENTE IM PM VON SWP

### Gantt Diagramm mit Meilensteinen mit Drittanbieter Software



## 64 DOKUMENTE IM PM VON SWP

---

- **Ressourcenmanagement** (Zuweisung von Ressourcen, also Mitarbeiter, Werkzeuge, Ausrüstung, zu den Aufgaben, um sicherzustellen, dass genügend Kapazitäten für die erfolgreiche Projektdurchführung vorhanden sind)
- **Risikomanagement** (Identifikation, Bewertung, Management und Überwachung der Risiken im Projekt sowie Festlegung von Strategien zur Risikobewältigung)
- **Qualitätsmanagement** (Beschreibung der Qualitätsstandards, -richtlinien und -prozesse, um sicherzustellen, dass die Qualitätsziele des Projekts erreicht werden)

## 65 DOKUMENTE IM PM VON SWP

---

- **Kommunikationsplan** (Festlegung der Kommunikationsstrategie, -methoden und -frequenzen innerhalb des Projektteams und mit den Stakeholdern)
- **Budget und Kostenmanagement** (Kostenschätzung, Budgetierung und Kontrolle der Projektkosten sowie Überwachung von Ausgaben und Budgetabweichungen)
- **Lieferung und Bereitstellung** (Planung der Auslieferung und Bereitstellung der Software, einschließlich Phasen, Rollout-Strategien und Schulungsplänen)
- **Änderungsmanagement** (Dokumentation und Kontrolle von Änderungsanfragen, Änderungen am Projektumfang und deren Auswirkungen auf Zeit, Kosten und Qualität)

## 66 DOKUMENTE IM PM VON SWP

---

- **Abnahmekriterien und Akzeptanzprüfung** (Definition der Kriterien für die Abnahme der Software und die Zustimmung der Stakeholder, um sicherzustellen, dass die Ziele des Projekts erreicht wurden) bzw.
- **Veröffentlichungsprüfung** (wurde Lizenzpflichtige Software verwendet, kostenpflichtige Software von Drittanbietern, Markenrechte, ...)
- **Wichtige Kontakte und Verantwortlichkeiten** (Auflistung der Schlüsselkontakte im Projekt und ihrer jeweiligen Verantwortlichkeiten)



## 67 VORGEHENSMODELL – WASSERFALL MODELL

---

- Ein Vorgehensmodell stellt Methoden und Elemente des Projektmanagements zu Prozessen und Phasen eines standardisierten Projektablaufes zusammen.
- In diesem Sinne ist ein Vorgehensmodell als Projektmanagementsystem nach DIN 69904 und 69905 anzusehen.

## 68 VORGEHENSMODELL – WASSERFALL MODELL

---

Was ist das?

- Das Wasserfallmodell (englisch: waterfall model) ist ein lineares Vorgehensmodell, das Entwicklungsprozesse in aufeinanderfolgende Projektphasen unterteilt.
- Im Gegensatz zu iterativen (dynamischen) Modellen wird jede Phase nur einmal durchlaufen.
- Die Ergebnisse einer jeden Vorgängerphase gehen als Vorannahmen in die Folgephase ein.
- Zur Anwendung kommt das Wasserfallmodell insbesondere in der Software-Entwicklung.



## 69 VORGEHENSMODELL – WASSERFALL MODELL

---

Wer hat das erfunden?


- Die Entwicklung des klassischen Wasserfallmodells wird dem Computerwissenschaftler Winston W. Royce zugeschrieben.
- Er ist aber nicht der Erfinder, sondern hat die Schwächen von linearen Vorgehensmodellen aufgezeigt. Er schlägt iterative Modelle vor
- Wasserfall folgt seiner Idee, sieht aber nur eine Iteration vor



## 70 VORGEHENSMODELL – WASSERFALL MODELL

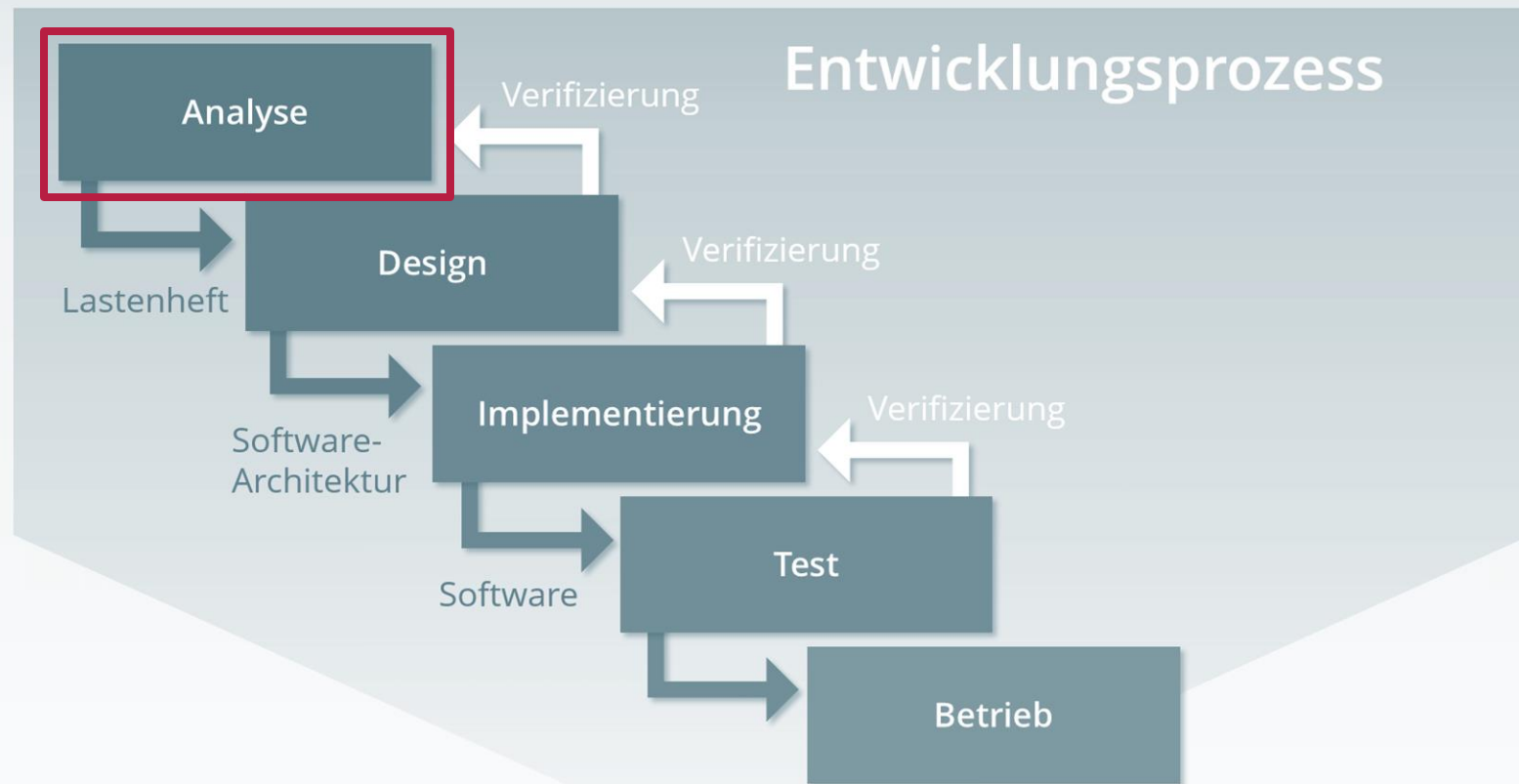
---

Wie funktioniert das?

- Analyse (Planung, Anforderungsanalyse und -spezifikationen)
  - Design (Systemdesign und -spezifikationen)
  - Implementierung (Programmieren und Modultests)
  - Test (Systemintegration, System- und Integrationstests)
  - Betrieb (Auslieferung, Wartung und Verbesserungen)
- 
- Phasen des Softwareprojekts



# 71 VORGEHENSMODELL – WASSERFALL MODELL



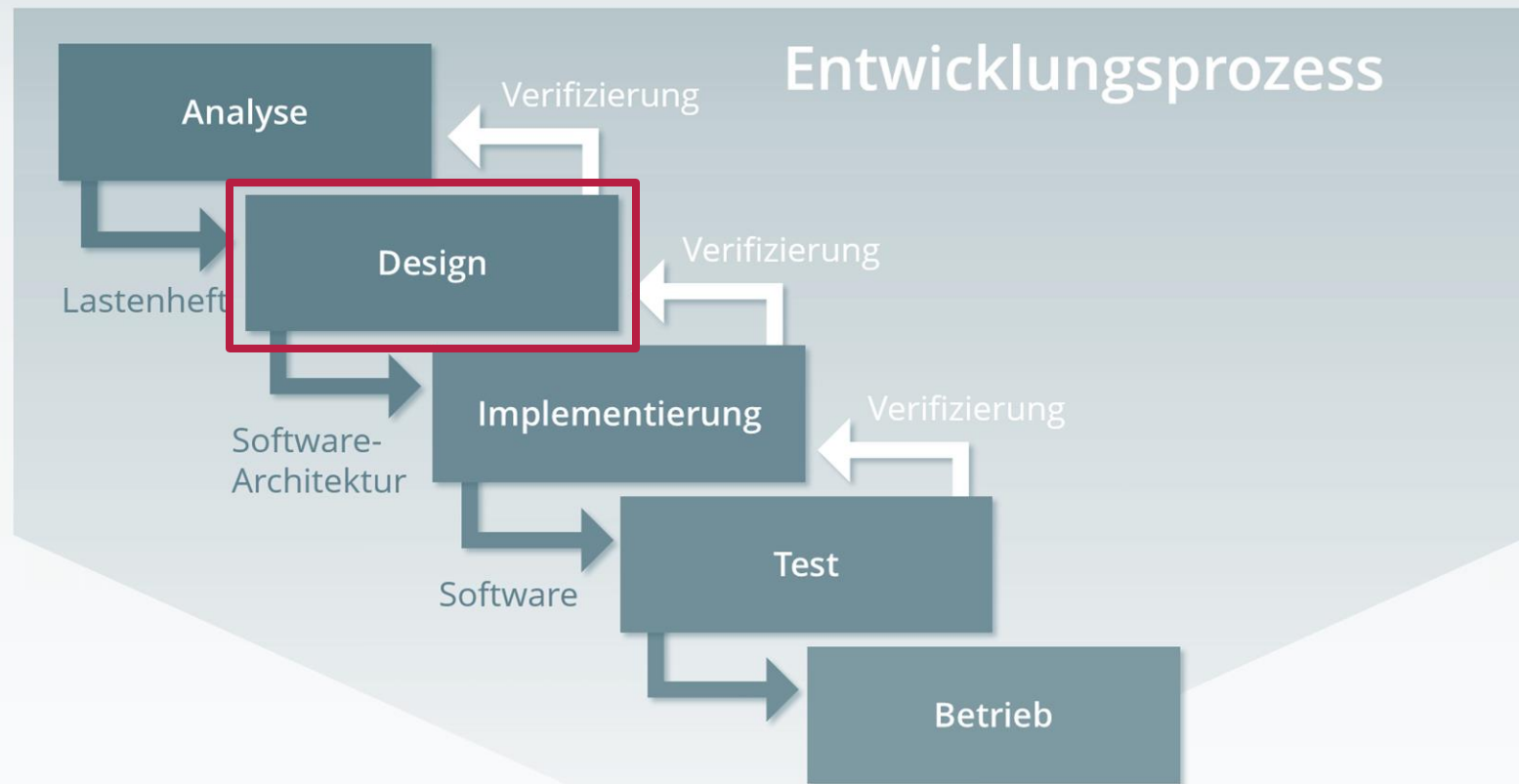
## 72 VORGEHENSMODELL – WASSERFALL MODELL

---

### Analyse

- **Machbarkeitsstudie:** Software-Projekt bezüglich Kosten, Ertrag und Realisierbarkeit einschätzen → Lastenheft (grobe Beschreibung der Anforderungen), Projektplan, Projektkalkulation und eventuell Angebot für den Kunden
- **Anforderungsdefinition:** Ist-Analyse (= Problembereich) und Soll-Konzept (= Funktionen und Eigenschaften der Software, um Anforderungen zu erfüllen)
- **Analyse der Anforderungsdefinition:** definieren von Teilaufgaben und deren Lösung

## 73 VORGEHENSMODELL – WASSERFALL MODELL



## 74 VORGEHENSMODELL – WASSERFALL MODELL

---

### Design

- **Software-Architektur:** wie wird die Software strukturiert (Module, Backend, Middleware, Frontend, Datenbank, DBMS, ...)
- **Bauplan der Software:** Komponenten wie Schnittstellen, Frameworks oder Bibliotheken (z.B. mittels PAP nach DIN 66001)

Ergebnis: Entwurfsdokument mit Software-Bauplan sowie Testplänen für einzelne Komponenten.





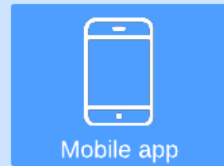
75

## Current state

### Free user



Free courses



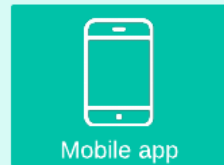
View  
courses  
only

### Paid user



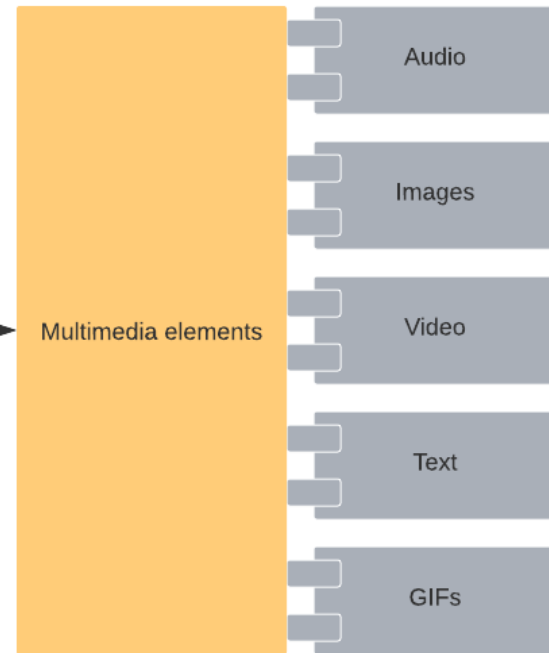
10,000+ courses

Educational  
resources

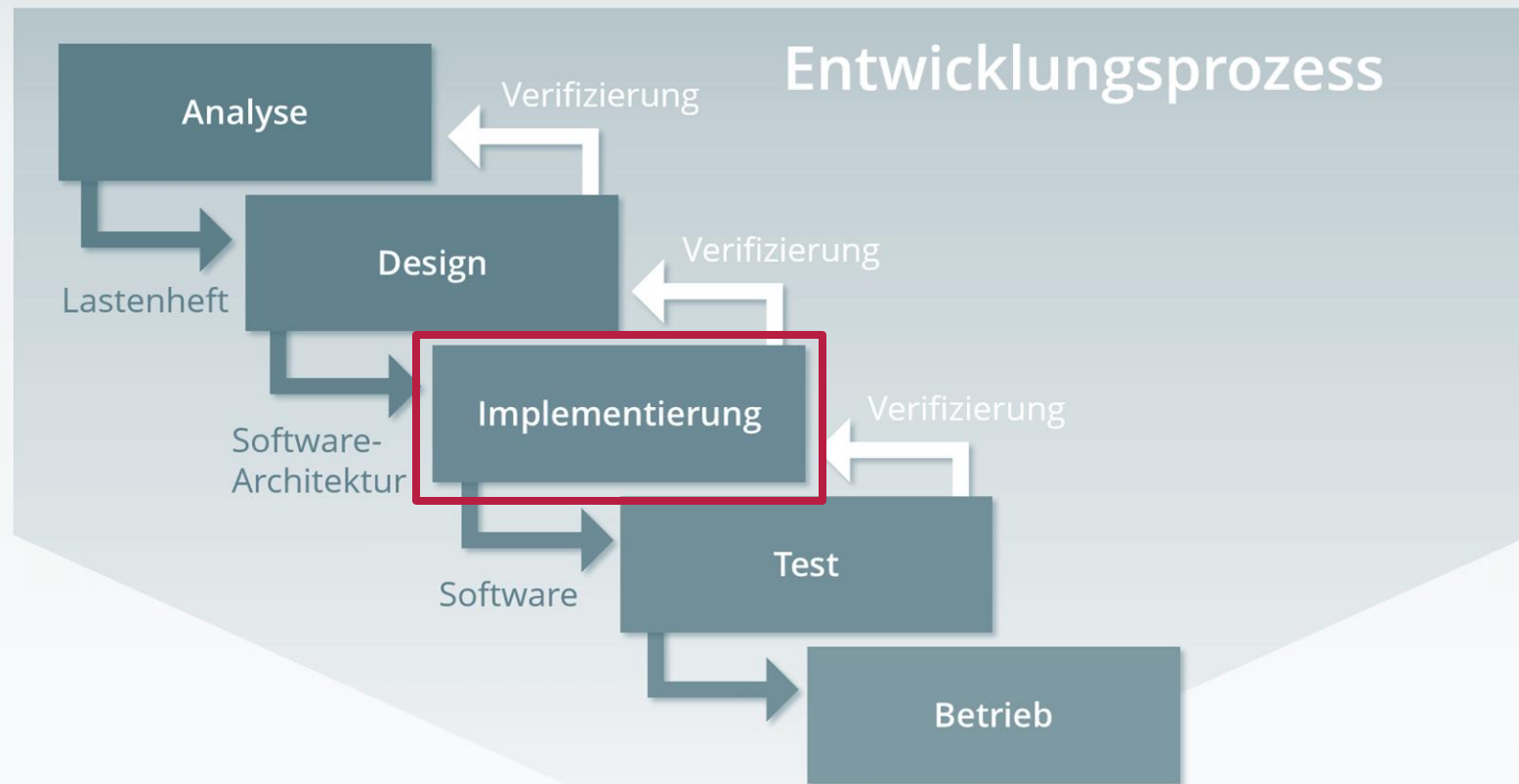


Limited number  
of courses

## Software-Architektur



## 76 VORGEHENSMODELL – WASSERFALL MODELL



# 77 VORGEHENSMODELL – WASSERFALL MODELL

---

## **Implementierung**

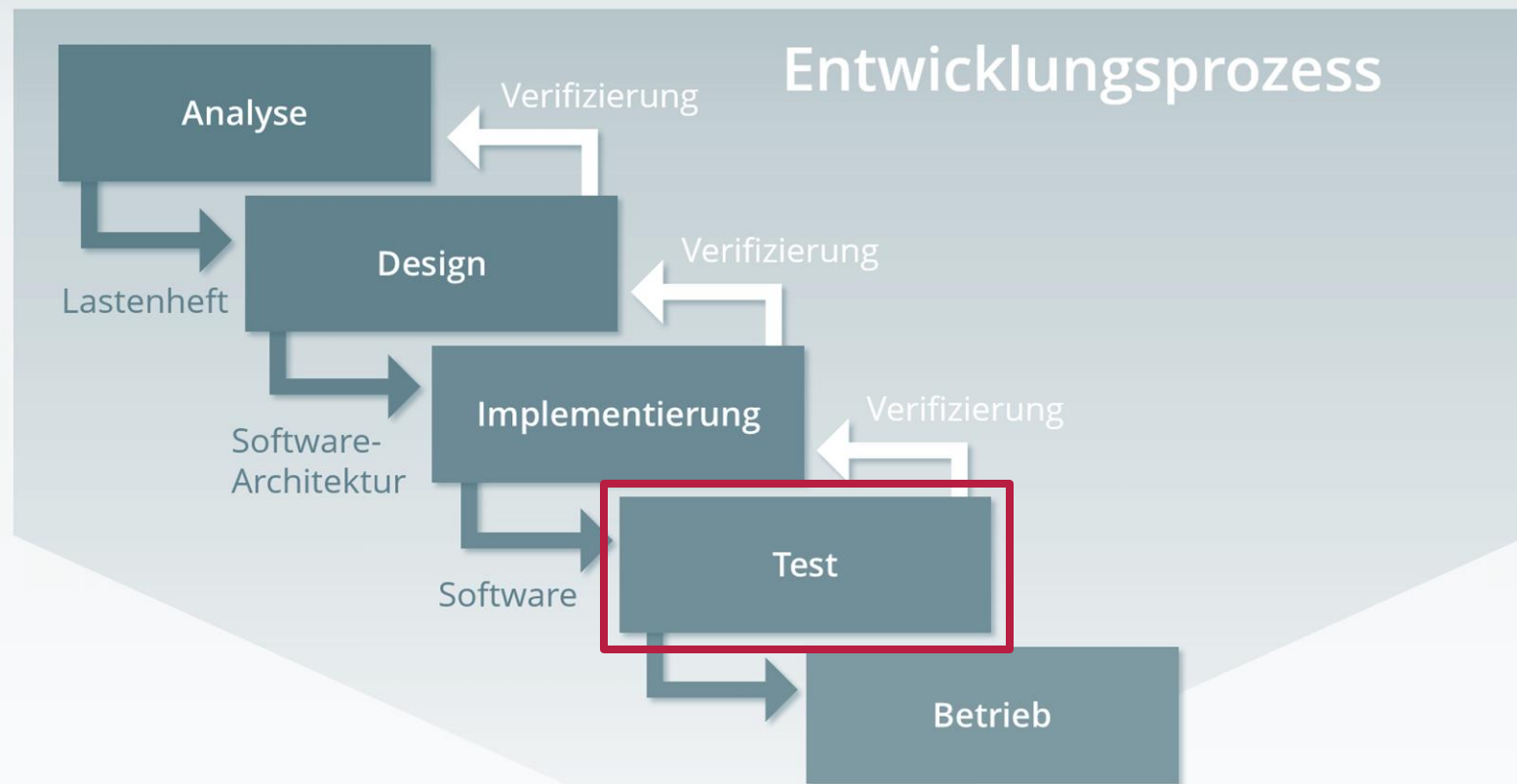
= Implementierungsphase ist die Realisierung der Software-Architektur der Design-Phase

- Software-Programmierung = Software-Entwurf in der gewünschten Programmiersprache, Komponenten separat entwickeln
- Fehlersuche
- Modultests = Module Schritt für Schritt ins Gesamtprogramm integrieren und testen

Ergebnis: Software-Produkt, das in der nachfolgenden Phase zum ersten Mal als Gesamtprodukt getestet wird (Alpha-Test).



## 78 VORGEHENSMODELL – WASSERFALL MODELL





## 79 VORGEHENSMODELL – WASSERFALL MODELL

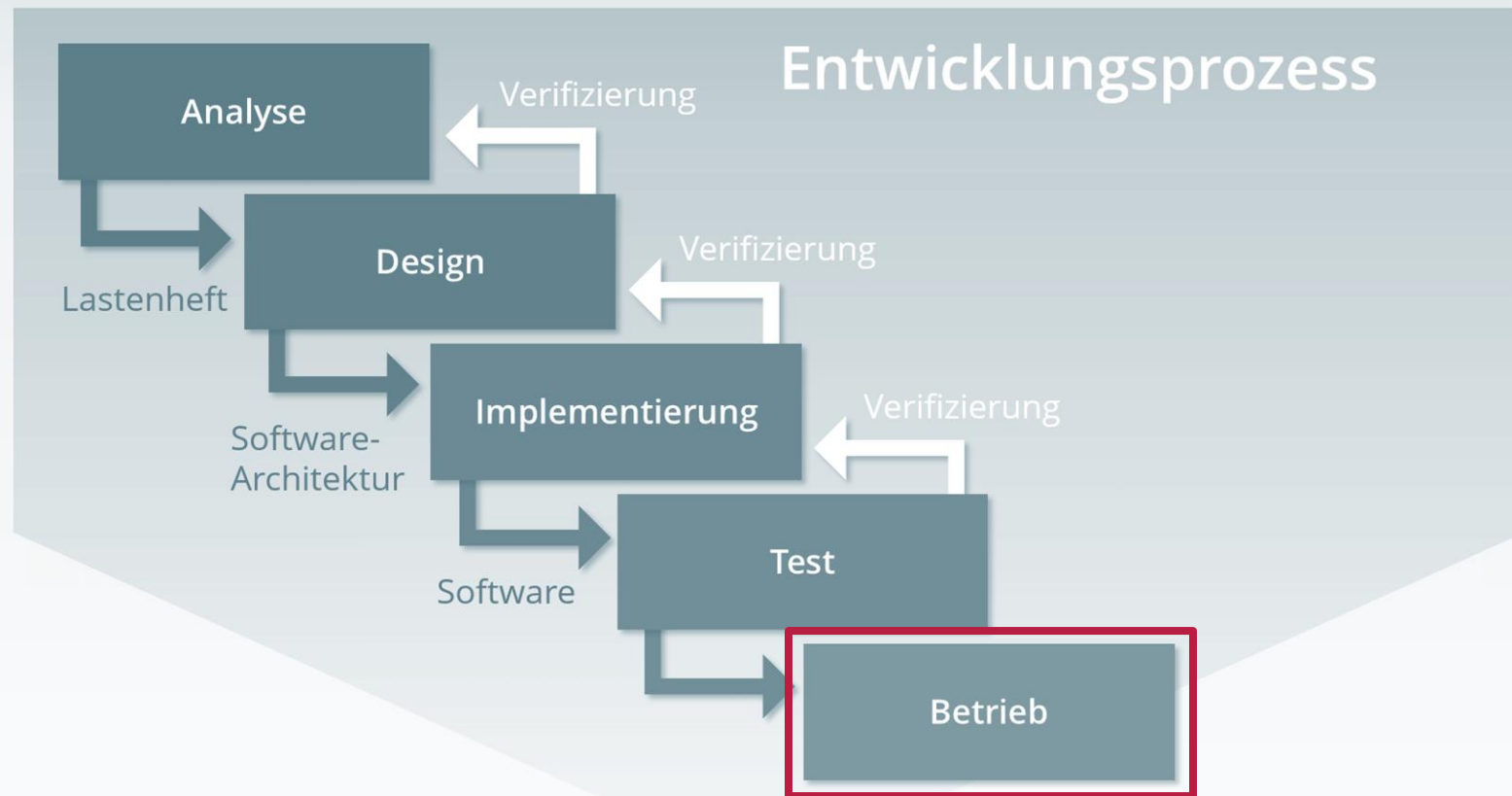
---

### Test

= Integration der Software in die gewünschte Zielumgebung

- Zunächst als Beta-Version an den Endbenutzer ausgeliefert (Beta-Test)
- **Akzeptanztest** = Testen ob alle Anforderungen erfüllt werden
- Hat das Software-Produkt den Akzeptanztest überstanden, ist es bereit für den **Release**

## 80 VORGEHENSMODELL – WASSERFALL MODELL



## 8 | VORGEHENSMODELL – WASSERFALL MODELL

---

### **Betrieb**

- Nach erfolgreichem Abschluss der Testphase wird die Software für den Betrieb im Produktiveinsatz freigegeben.
- Die letzte Phase des Wasserfallmodells schließt Auslieferung, Wartung und Verbesserung der Software ein.

## 82 VORGEHENSMODELL – WASSERFALL MODELL

---

### Vorteile

✓ Einfache Struktur durch klar abgegrenzte Projektphasen.

✓ Gute Dokumentation des Entwicklungsprozesses durch klar definierte Meilensteine.

✓ Kosten und Arbeitsaufwand lassen sich bereits bei Projektbeginn abschätzen.

✓ Projekte die nach dem Wasserfallmodell strukturiert werden, lassen sich auf der Zeitachse gut abbilden.

### Nachteile

✗ Komplexe oder mehrschichtige Projekte lassen sich nur selten in klar abgegrenzte Projektphasen unterteilen.

✗ Geringer Spielraum für Anpassungen des Projektablaufs aufgrund veränderter Anforderungen.

✗ Der Endanwender wird erst nach der Programmierung in den Produktionsprozess eingebunden.

✗ Fehler werden mitunter erst am Ende des Entwicklungsprozesses erkannt.



## 83 VORGEHENSMODELL – WASSERFALL MODELL

---

Wenn's dein Vorgehensmodell  
nutzen, aber keine Ahnung haben,  
dass es nicht agil ist.



## 84 QUELLEN

---

- <https://projekte-leicht-gemacht.de/blog/projektmanagement/klassisch/projektplanung/projektstrukturplan/#Vorteile-eines-Projektstrukturplans>