

## Projektentwicklung

### **1. EDV-Projekt**

#### **1.1 Definition**

- abgegrenzte Entwicklungsaufgabe
- von begrenzter Dauer
- mit begrenztem Rahmen von Hilfsmitteln
- mindestens 3 Monate
- Schaffung einer individuellen Organisation

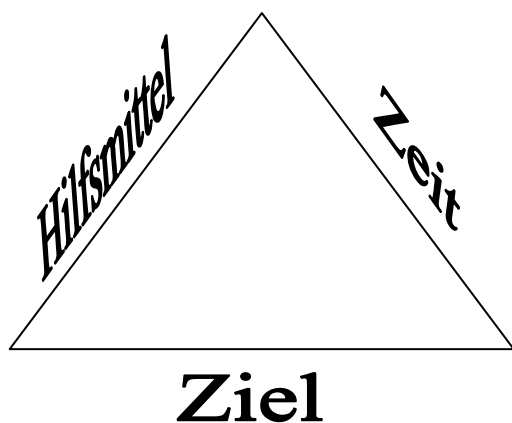
#### **1.2 Kriterien für ein EDV Projekt**

- erstmalige Realisierung einer Idee
- von erheblicher Bedeutung für das Unternehmen (Weiterbestand, Geschäftserfolg, Ruf...)
- Improvisation ist nicht möglich
- individuelle Organisation ist notwendig
- Problemlösung mit EDV-Unterstützung

#### **1.3 Projektumfang**

- Dauer (min. 3 Monate, max. 2 Jahre)
- Mitarbeiter (min. 3, max. 15)
- Definition von Teilprojekten (jeweils max. 20 Mannjahre je Teilprojekt)

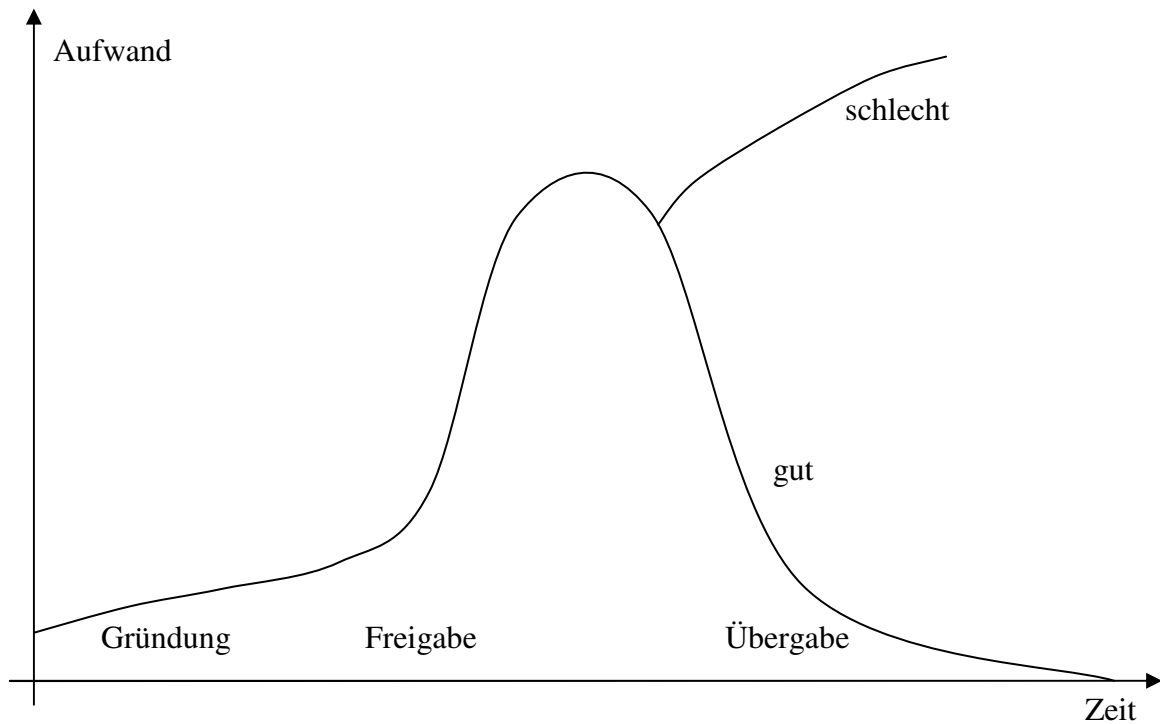
#### **1.4 Bestimmungsgrößen**



## 1.5 Phasenschema

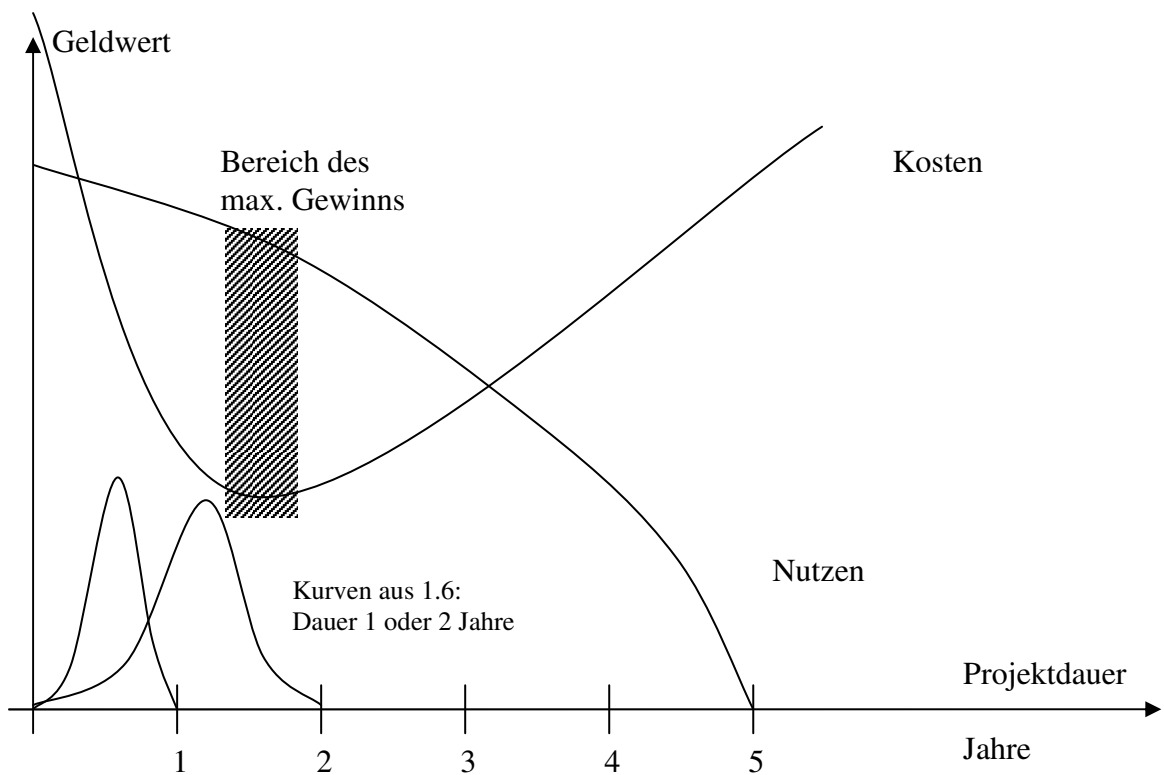
PROJEKTMANAGEMENT										
DOKUMENTATION										
I D E E	V O R U N T E R S U C H U N G	V O R S T U D I E	P R O J E K T P L A N U N G	S Y S T E M E N T W I C K L U N G	S Y S T E M I N F R A S T R U K T U R	Detail - entwicklung	S Y S T E M T E S T	I N S T A L L A T I O N	W A R T U N G	Kontrolle
						Codierung mit Einzeltest				Nachbesserung
						Datenerfassung / -übernahme				Erweiterung
Projektgründung ^      ^ Projektfreigabe      ^ Projektübergabe										

## 1.6 Projektkosten je Tag



## 1.7 Einfluss der Projektdauer auf Gesamtkosten und Gesamtnutzen

- Ausgehen von einer festgehaltenen Problemstellung wird die Projektdauer variiert.



## 1.8 Tabellarische Aufstellung von Zeit- und Aufwandsverteilung

Projektphase	Zeit	Aufwand
Bis Projektgründung	5 %	4%
Projektplanung und Systementwicklung <sup>1)</sup>	20 %	13%
Detaillentwicklung <sup>2)</sup>	30%	20%
Codierung und Einzeltest	30%	50%
Systemtest	15%	13%

1) Technische Planung (Module, Schnittstellen)

2) Theoretische Planung (genauer)

## 1.9 Alternatives Phasenschema

- Initialisierung
- Ist-Analyse
- Ist-Kritik
- Sollkonzept
- Durchführbarkeitsstudie
- Alternativensuche
- Entscheidung
- Realisierung
- Implementierung
- Test
- Wartung

## 1.10 Projektdurchführung (Wer führt das Projekt durch?)

a) EDV-Abteilung	b) Fachabteilung	c) Externer Berater
- mangelnde Fachkenntnisse	- keine EDV-Erfahrung	- teuer
- Ablenkung durch Routineaufgabe	- keine Projekterfahrung	- Kommunikation und Wartung problematisch
- Konflikte zwischen Abteilungen	- Risikofurcht	- Keine Betriebserfahrung
+ gute EDV-Kenntnisse	+ praktische Erfahrung	+ Erfahrung
+ Erfahrung	+ Problemkenntnisse	+ Standardlösungen

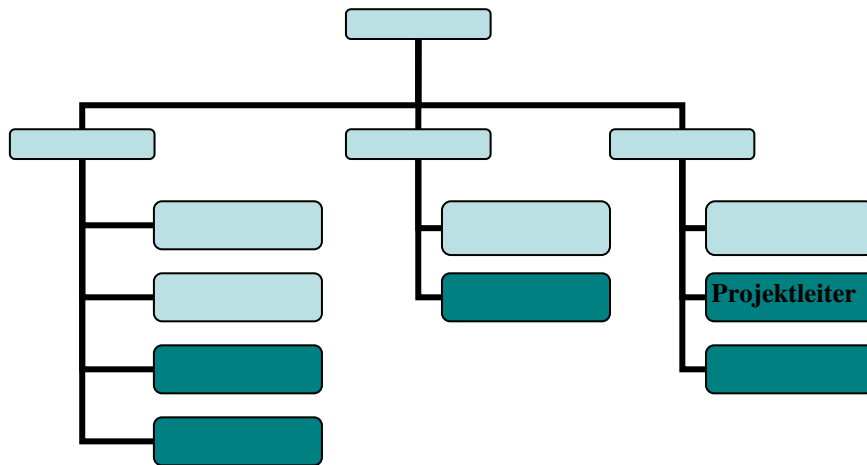
### d) Projektteam:

Zusammenarbeit von Mitarbeitern aus verschiedenen Abteilungen/Firmen, um die jeweiligen Vorteile zu nützen:

- Organisation des Projektes schwierig
- Projektleiter ist gefordert

## 1.11 Organisationsformen für das Projektteam

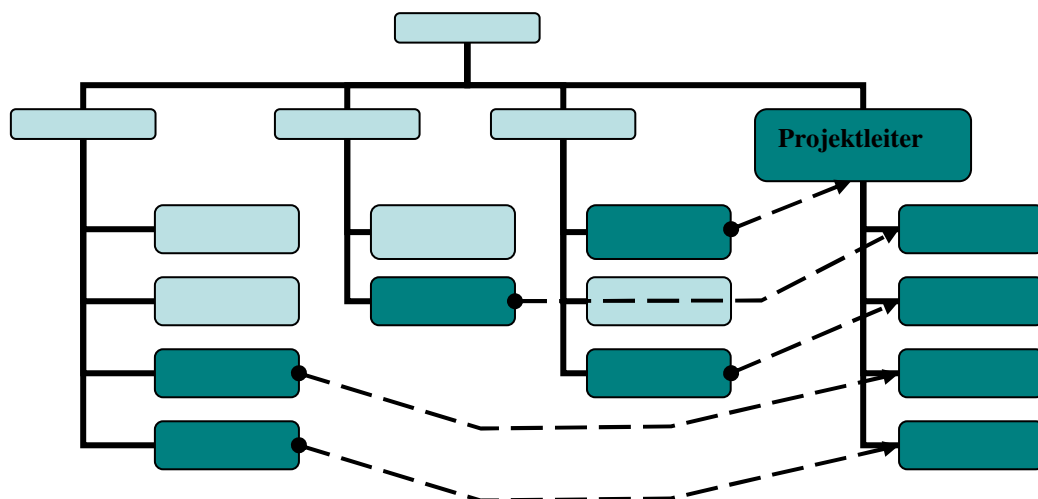
- **Projektausschuss:**



Testmitglieder bleiben am alten Arbeitsplatz in der Abteilung und arbeiten von dort aus am Projekt mit:

- Kommunikation schwierig
- Überbelastung durch Routinetätigkeiten
- Kompetenzschwierigkeiten
- Gefahr von Verzögerung
- + Aufbau leicht möglich, billig
- + Zusammenarbeit in der Abteilung weiterhin möglich
- + Leichter Abbau der Organisation

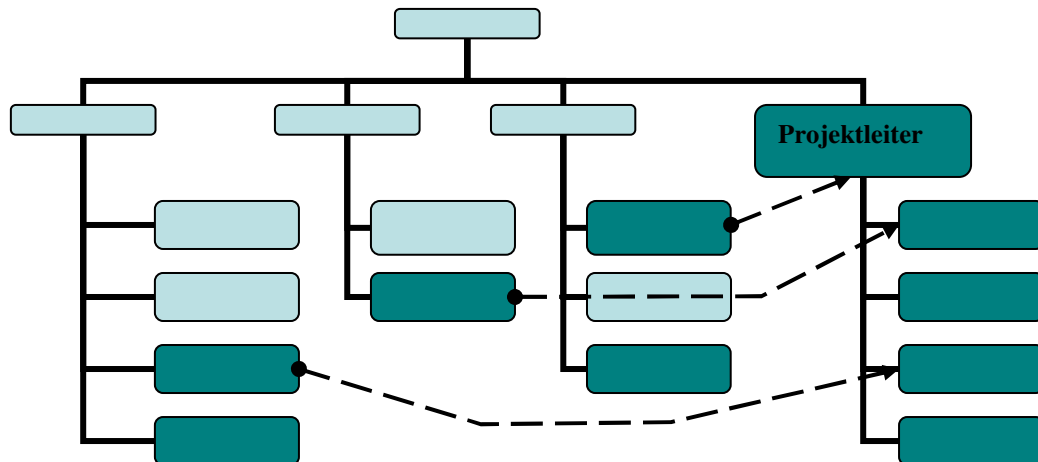
- **Reine Projektorganisation**



Projekt ist neue Abteilung, Projektleiter ist Abteilungsleiter. Mitarbeiter erhalten neuen Arbeitsplatz:

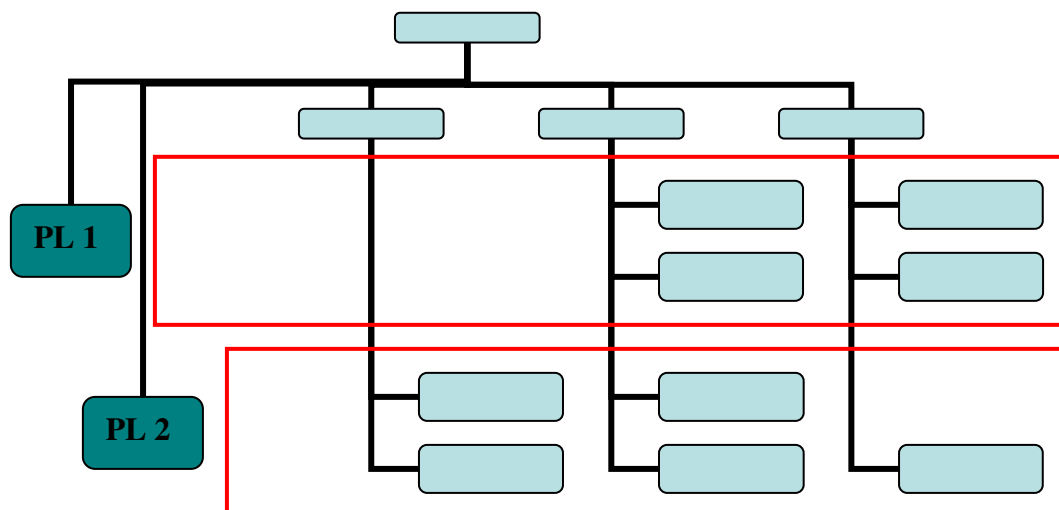
- Neuaufnahme von Personal für Abteilungen notwendig
- Aufbau der Abteilung (= Projekt) ist schwierig und teuer
- Neid des Nichtausgewählten
- Abbau der Organisation ist schwierig
- + Volle Konzentration auf das Projekt
- + Rasche Projektdurchführung
- + Kurze Kommunikationswege

### • Gemischte Projektorganisation



Mitarbeiter die dauernd für das Projekt wichtig sind bilden eine neue Abteilung.  
Andere arbeiten (zeitweise) vom alten Arbeitsplatz aus mit.  
Ziel: Kombination der Vorteile der beiden anderen Varianten.

### • Matrixorganisation



Jeder Mitarbeiter arbeitet in einer Abteilung und in einem Projekt.  
Projektleiter entscheidet innerhalb des Projektes- Abteilungsleiter ist zuständig zwischen den Projekten und projektübergreifend (Schulungen, Firmenstandards).  
Nur möglich in Unternehmen, die vom Projektgeschäft leben.

## 2. Voruntersuchung

### 2.1 Anlass und Ursache für ein EDV - Projekt

- Beschleunigung von Abläufen
- Fehler in der Organisation
- bessere Transparenz von Betriebsabläufen
- bessere Kontrolle
- Automatisierter Workflow
- Ausweitung des Unternehmens  
(neue Filialen, neue Produkte, neue Verkaufsgebiete, Umsatzsteigerung)
- juristische und logistische Gründe
- Modernisierungsbedürfnis
- Prestige
- Personalschwierigkeiten  
(mangelnde Leistungsbereitschaft, Genauigkeit, Qualität, schwankende Qualität, gesetzliche Begrenzung der Arbeitszeit, Mangel an geeigneten Arbeitskräften)

### 2.2 Durchführung der Voruntersuchung

Formulierung des Projektziels	
Ermittlung der Benutzeranforderungen	
	Ermittlung möglicher DV Verfahren
bis keine weiteren Verfahren vorhanden	
	Ermittlung der Kosten
	Ermittlung des Nutzens
	Ermittlung des Risikos
	Absprache von Terminen
bis keine weiteren Verfahren vorhanden	
Auswahl eines DV Verfahrens	
Fixierung der Annahmen	
Information der Beteiligten	
Erstellung der Durchführbarkeitsstudie	

- **Projektziel:**  
Befragung der Geschäftsleitung, Abteilungsleiter, Manager
- **Benutzeranforderungen:**  
Erforderliche Genauigkeit und Aktualität der Daten  
Fehlerkorrektur, Sicherheitsvorkehrungen  
Einteilung in NEED TO HAVE und NICE TO HAVE

- **Suche nach möglichen Lösungen (DV-Verfahren):**  
Literaturstudium (Fachbücher, Prospekte, Veröffentlichungen)  
Erfahrung aus Vorprojekten  
Konkurrenzbeobachtung, persönliche Gespräche
- **Kosten:**  
Kosten für die Projektdurchführung  
Anschaffung von Hardware/Software  
Betriebskosten
- **Nutzen:**  
Einnahmen  
Direkte Einsparungen (Personal, Zeit, Material)  
Indirekte Einsparungen (Fehlerbehebungskosten geringer, bessere Qualität)  
Umwegrentabilität (nicht qualifizierbarer Nutzen)
- **Risiko:**  
Bewertung wie Schulnoten, oder niedrig – mittel – hoch  
Neue Technologien → hohes Risiko  
Spezialsoftware riskanter als Standardsoftware  
Entwicklung riskanter als Kauf  
Bekanntes Personal → geringerem Risiko
- **Auswahl:**  
Alle Parameter berücksichtigen
- **Fixierung der Annahmen:**  
Schätzungen für das gewählte Verfahren verfeinern
- **Information der Betroffenen:**  
Sowohl der Benutzer als auch die Teammitglieder
- **Durchführbarkeitsstudie:**  
Zusammenfassung aller Anforderungen, Angaben und Annahmen  
Darstellung der technischen Durchführbarkeit  
der praktischen Anwendbarkeit  
der Rentabilität

### 3. Vorstudie

- Optional (kann entfallen)
- Behandelt schwierige Teilaufgabe der Voruntersuchung mit wissenschaftlichen Methoden
- Wird durchgeführt von Externen (Hochschule, Universität)  
z.B.: technologische Neuigkeiten / Probleme
- Marktforschung
- Ergebnis der Vorstudie beeinflusst die Durchführbarkeitsstudie



## 4. Projektgründung

- Formaler Beginn des Projektes
- Bisherige informelle Informationen, Diskussionen, Aktivitäten, Dokumente werden in einen formalen Rahmen gestellt:
  - Ernennung des Projektleiters
  - Aufstellung des Teams
  - Erstellen des Projektkontraktes
  - Abhaltung von Projektgründungsreviews und des Kick-Off-Meetings

### 4.1 Projektleiter

- Verantwortlich für Erfolg oder Misserfolg des Projektes
- **Aufgaben:**
  - Planung, Kontrolle und Steuerung des Projektes
  - Erstellung des Projektkontraktes
  - Schätzung von Aufwand, Zeit, Kosten
  - Information der Geschäftsleitung
  - Information des Teams
  - Sicherstellung eines guten Arbeitsklimas
- **Rechte:**
  - Mitsprache bei der Teamaufstellung
  - Weisungsbefugnis an das Team
  - Entscheidungsrecht innerhalb des Projektrahmens
  - Information über wichtige betriebliche Ereignisse
  - Budgetplanung und Kontrolle
  - Rechtzeitige Information (Mitwirkung möglichst ab der Voruntersuchung)
- **Anforderungen an einen Projektleiter:**
  - Führungsverhalten
  - Fachliche Kenntnisse
  - EDV Kenntnisse
  - Planungs- und Kontrolltechniken
  - Kommunikationstechniken
  - Motivationstechniken
  - Techniken der Problemlösung
  - Techniken der Konfliktlösung
  - Fähigkeit zu delegieren (nicht überfordern/unterfordern)
  - Mitarbeit in (min. zwei) Projekten als Teammitglied

## 4.2 Projektteam

- Verantwortlich für die übertragenen Teilaufgaben.
- Aufnahme von neuen Mitarbeitern (mit geringer Praxis) kann auch Vorteile haben: neue Ideen, neue Technologien, sind nicht betriebsblind.
- **Vertretene Abteilungen:**
  - EDV-Abteilung
  - Fachabteilung
  - Andere (teilweise) betroffene Abteilungen
  - Revisionsabteilung
- **Auswahlkriterien:**
  - Fachliche Qualifikation
  - Kontaktfähigkeit
  - Fähigkeit zur Teamarbeit

## 4.3 Projektvertrag

- Vertrag über das Projekt, wichtigste Inhalte:
  - Produkt
  - Übergabe
  - Hilfsmittel
  - Rechte und Pflichten
- **Produkt:**
  - Funktion (welche Aufgaben werden gelöst?)
  - Leistung: Mengenangaben zu den Funktionen  
z.B.: Laufzeit, Datemengen, Datendurchsatz, Anzahlangaben
  - Hardware/Software
  - Interfaces (Schnittstellen):
    - Technische Schnittstellen (zu anderen Programmen wie z.B. Word)
    - Mensch-Maschine-Schnittstellen (Benutzeroberflächen)
  - Wirtschaftlichkeit
  - Prioritäten
- **Übergabe:**
  - Programme: Sources (Quelltext) oder ausführbare Programme?
  - Dateien:
    - Format
    - Datenträger
    - Aktualität
    - Dokumentation der Daten
  - Formulare
  - Dokumentation:
    - Systemdokumentation

- Programmdokumentation
- Benutzer- und Bedienerdokumentation
- Format
- Abnahmetest:
  - Messbare Kriterien für die Erfüllung der Anforderungen.
  - Wer führt den Test durch?
  - Wer trägt die Kosten für Wiederholungen des Tests?
- Ausbildung:
  - Definition von Kursen (Art, Inhalt, Dauer, Ort, Häufigkeit)
  - Bezahlung der Kurse
- Wartung:
  - Wann und wie erfolgt die Wartung (bei Bedarf, periodisch)
  - Wer trägt die Kosten?

- **Hilfsmittel**

- Wer stellt wann was für welche Zeit zur Verfügung:
  - Büroraum
  - Hardware
  - Material
- Regelung für Ausgaben:
  - Bewilligung?
  - Personalausgaben (Überstunden, Prämien, Dienstreisen)
- Termine:
  - Fixtermine: Zwingend einzuhalten
  - Plantermine: Ergeben sich aus Studien, nicht bindend.
  - Sonstige Termine: Betreffen Wirtschaftlichkeit

- **Rechte und Pflichten**

- Geschäftsbedingungen
- Zahlungsbedingungen (Teilzahlungen)
- Eigentumsvorbehalt
- Garantien
- Schadenersatz
- Pönale (Vertragsstrafen)
- Rechte an technischen Daten
- Berichtswesen:
  - Wer bekommt wann Berichte über den Projektfortschritt?
  - Planung von Besprechungen

- **Abschließende Bemerkungen**

- Projektkontrakt soll vor Auffassungsdifferenzen und einer unkontrollierten Ausweitung des Projektes schützen.
- Der Projektkontrakt enthält also auch Einschränkungen, d.h. welche Aufgaben vom Produkt nicht erfüllt werden.
- Projektleiter soll keine zu optimistischen Versprechungen machen.
- Benutzerpflichten nicht vergessen!
- Sehr wichtig ist verständliche Formulierung

#### **4.4 Projektgründungsreview**

- Rückblick auf vorhergehende Phase(n)
- Kritische Untersuchung des Projektkontraktes in einer Diskussionsrunde
- Prüfung der Voraussetzungen für das Projekt:
  - Kompetenter Auftraggeber
  - Zusammensetzung des Teams
  - sachliche Voraussetzungen (Zielsetzung, Qualitätskriterien)

#### **4.5 Kick-off-Meeting**

- Erstes Zusammentreffen aller Teammitglieder
- Ziel ist gegenseitiges Kennenlernen
- Gruppendynamische Effekte
- Austausch von Erfahrungen
- Diskussion über die fundamentalen Parameter des Projektes (Zeit, Hilfsmittel, Termine, Geld)

#### **4.6 Projektmarketing**

- Dient dazu das Projekt bekannt zu machen
- Fördert das Zusammengehörigkeitsgefühl
- Projektnamen festlegen
- Logo
- Briefpapier
- Homepage
- Firmenzeitung

### **5. Projektplanung**

- In dieser Phase werden:
  - die Aufgaben bestimmt, die im Verlauf des Projektes durchzuführen sind
  - Hilfsmittel zu diesen Aufgaben zugeordnet
  - Kontrollmechanismen festgelegt
- Es werden folgende Dokumente erstellt:
  - Systemanforderungskatalog
  - Projektplan
- Grundlage ist die Erfassung des Ist-Zustandes

#### **5.1 Ist-Zustands-Aufnahme**

- Interview
- Fragebogen
- Beobachtung

- **Interview**
  - flexibel
  - billig
  - rückfragen möglich
  - größere Motivation beim Befragten
- **Tipps für ein Interview**
  - Fragen schriftlich vorbereiten
  - Fragen sortieren und strukturieren
  - Termin vereinbaren und einhalten
  - Ungefähre Dauer bekannt geben
  - Positives Gesprächsklima schaffen
  - Organisatorische Fehler notieren aber nicht werten oder kommentieren
  - Verständliche Sprache
  - Interesse und Verständnis zeigen
  - Trennung von Tatsachen und Meinungen
  - Informationen ordnen und gegenseitig überprüfen
  - Protokoll führen
- **Fragebogen**
  - Geeignet bei großer Anzahl von Informationsträgern
  - Sehr wichtig ist die exakte und genaue Formulierung
  - Problem bei geringer Rücklaufquote
  - Kontrollfragen
  - Dokumentiert sich selbst
- **Beobachtung**
  - sehr teuer
  - geeignet um organisatorische Fehler zu erkennen

## 5.2 Systemanforderungskatalog

- Ausgehend von der Ist-Zustandsaufnahme formuliert ein Systemanalytiker die Problemstellung
- detaillierter als der Projektkontrakt (technisch exakter)
- technisch orientiert
- verbale Formulierung
- möglichst allgemein verständlich

## 5.3 Projektplan

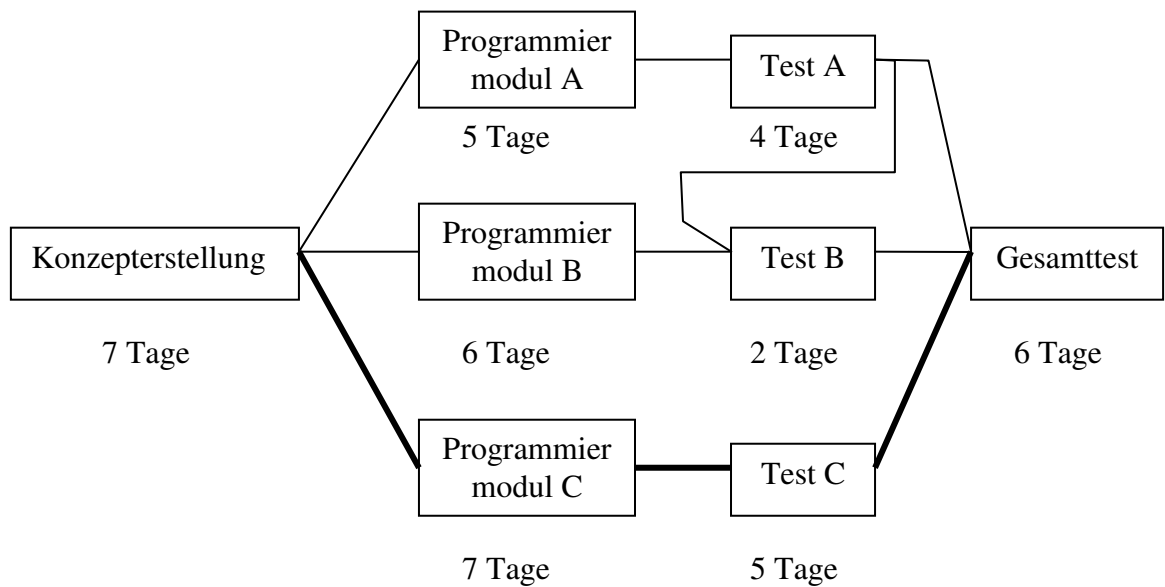
- Organisatorische Vorbereitung der Projektdurchführung
- **Dokumentationsplan**
  - Idealform:

- automatisch
- mitwachsend
- Inhalt:
  - Zahl der Kopien, Verteiler
  - Hilfskräfte
  - Verantwortung
  - Form
  - Index
- **Kommunikation- und Berichtsplan**
  - Hauptursache für Misserfolg ist mangelnde Kommunikation!
  - Verständnis hebt Motivation
  - Projektleiter ist Ansprechpartner für alle Teammitglieder
  - Kommunikationsplan:
    - Regelung: Wer informiert wen
    - Telefon-, Adressverzeichnis
    - Literatursammelstelle
  - Berichtsplan:
    - Verteiler
    - Detaillierungsgrad
    - Formulare
    - Termine
- **Reviewplan**
  - kritischer Rückblick auf vorige Phase(n)
  - Diskussion mit dem Autor eines Deliverable  
(z.B.: Projektdokumente, Sources, ausführbares Programme, etc.)
  - Kurzreferate des Autors
  - projektfremde Zuhörer
  - freie Meinungsäußerung
  - Ziel ist Erfahrungs- und Informationsaustausch
  - Kritik an der Sache, nicht am Autor
  - Qualitätssicherung im Vordergrund
  - Entscheidung über Beginn der Folgephase
- **Terminisierung von Reviews**
  - Meilensteinreviews (mögliche Zeitpunkte):
    - Ende der Voruntersuchung
    - Systemanforderungskatalog fertig
    - Projekt freigegeben
    - Programme formal richtig
    - alle Programme integriert
    - Systemtest positiv
    - Systemübergabe
    - bei Gefährdung des Projektes

- Periodische Reviews: etwa monatlich

- **Kontrollplan**

- Kontrolle des Projektverlaufs
- bei großen Projekten: Netzpläne:
  - graphische Darstellung der Vorgänge durch Rechtecke
  - Dauer der Vorgänge wird eingetragen
  - logische Abfolge der Vorgänge wird eingezeichnet
  - Übersichtlichkeit
  - Auswirkung von Änderungen lässt sich leicht erkennen
  - sinnvoll nur mit Softwareunterstützung
  - Anzahl von Teilaktivitäten (15 - 200)
  - z.B.:



- Budgetierungsprogramme
- Aufwandsschätzverfahren

- **Ausbildungsplan**

- für das Team:
  - Sachinformation
  - Technische Ausbildung
- für den Benutzer:
  - allgemeines EDV-Wissen
  - Hardwarebedienung
  - Systembedienung
- Festzulegen ist:
  - Art der Ausbildung (Selbststudium, Kurse)
  - Kurse: Ort, Dauer, Inhalt, Voraussetzungen
  - Wer trägt die Kosten?

- Wichtig: Schulung für später beginnende Teammitglieder

- **Testplan**

- Voraussetzungen für Testläufe schaffen (z.B.):
  - Hardware (Terminals)
  - Maschinenzeit
- Genaue Beschreibung der Testaktivitäten laut Projektkontrakt

- **Übergabeplan**

- Detaillierung des betreffenden Teils des Projektkontrakts (z.B.):
  - Änderung von Arbeitsabläufen
  - Änderung von Daten und Programmen
  - Verantwortliche bestimmen für die Umstellungsarbeiten und für die Kontrolle
  - Starthilfe für Benutzer
  - Installation vorbereiten (Strom, Klimatechnik)
- Wer ist für die Wartung zuständig?

- **Durchführungsplan**

- Bereitstellung der Arbeitspläne für das Projektteam wird geregelt:
  - Wo gearbeitet wird?
  - Arbeitsplätze ergonomisch gestalten
  - Infrastruktur bereitstellen (Telefon, Netzwerk, Kopierer,...)
  - Platz für Ablagen

- **Personalplan**

- Zuordnung von Aufgaben an Mitarbeiter
- Einteilung in Arbeitsgruppen ( $\leq 8$ )
- Planung von Neuaufnahmen

- **Sicherheitsplan**

<b>Gesundheitliche Gefährdung</b>	<b>Abhilfe</b>
Schirmauflösung, Flimmern, Blendung	gute Bildschirme
Haltungsschäden	ergonomische Möbel
Strahlung	Abschirmung
Gefährliche Maschinen	räumliche Distanz, Simulation
<b>Verlust von Daten und Programmen</b>	<b>Abhilfe</b>
Betrug, Diebstahl	Schlüssel, Zutrittskontrolle, biometrische Methoden, Passwort, Logfiles
Fehler, Fahrlässigkeit	Datensicherung, Notfallszenario üben, Schulung, zusätzliche Versicherung



## 6. Systementwicklung (Designphase)

- **Ziele dieser Phase**
  - Detaillierung der Punkte aus dem Systemanforderungskatalog  
(Beschreibung von Wegen, wie die gesteckten Ziele zu erreichen sind)
  - Auswahl eines DV Verfahrens
  - Beobachtung ob sich die Annahmen in der Durchführbarkeitsstudie als realistisch erweisen.
  - Verfeinerung des Projektplans
- **Meist 2 Stufen:**
  - Erstellung der funktionellen Spezifikation (Grobkonzept)
  - Erstellung der Detailspezifikation (Feinkonzept)
- **Aktivitäten in der Designphase**
  - Entwurf aller Systemfunktionen als genereller Ablauf
  - Entwurf aller Ein- und Ausgabeformate
  - Entwurf aller Datenbereiche, Tabellen, Feldbeschreibungen
  - Entwurf aller Datenbestandsbeschreibungen

### 6.1 Designteam

- Sollte so klein wie möglich sein
- Bei mehr als einem Designer muss ein Teamleiter ernannt werden
- **Auswahlkriterien für Designer:**
  - Programmiererfahrung  
(um die Auswirkungen von Designerentscheidungen zu erkennen)
  - Kreativität
  - Problemlösungsfähigkeit
  - sprachliche Ausdrucksfähigkeit

### 6.2 Designrichtlinien

- **Verwendung des Vorhandenen**
  - billiger
  - schneller verfügbar
  - getestet
  - Voraussetzung: Vorhandensein einer Dokumentation
- **Strukturierung (Top – Down – Design)**
  - Zerlegung der Gesamtaufgabe:
    - Hauptprogramme
    - Unterprogramme
    - Blöcke
    - Einzelanweisungen

- **Modularisierung**
  - Modul = Unterprogramm
  - Gliederung nach logischen Abgrenzungen (je Modul eine Teilaufgabe)
  - Je Modul nur einen Eingangspunkt und einen Ausgangspunkt
  - Resultate am Ende eines Moduls hängen nur von den Werten ab die am Beginn des Moduls übergeben werden (keinen globalen Daten)
  - Ein separates Modul für Ein-/Ausgabe
  - Baumstruktur der Modulkonstruktion
  - Max. Zeilenzahl je Modul nicht überschreiten (z.B. 40 Zeilen)
- **Möglichkeiten für Programmiererweiterungen**
  - Inklusion: Einfügen neuer Programmzeilen oder Datenelemente.
  - Antizipation: Vorbereitung für spätere Inklusionen.
  - Expandibilität: Punkte im Programm werden für späteren Einbau von Funktionsaufrufen vorbereitet.
  - Exklusion: Das Programm enthält alle denkbaren Funktionen, der Benutzer kann aber nur die aufrufen die er braucht und die er gekauft hat.
  - Parametrisierung: Statt Konstanten werden Werte aus übergeordneten Routinen übergeben.
  - Makros: Erlauben Änderungen im Sourcecode.
  - Externe Kontrolle: Steuerung des Programmablaufes durch Werte von außerhalb. (z.B. Registry, ini-Files, Parameterfiles).
  - Interpretative Systeme: Der Benutzer kann aus Bausteinen neue Funktionen selbst aufbauen.

### 6.3 Designhilfsmittel

- Sind Kommunikationsmittel, um die Ideen des Designers den Programmierern mitzuteilen.
- **Struktogramme**
- **Programmablaufplan (Flussdiagramme):**
  - veraltet
  - nicht strukturierte Programme sind möglich

- **Entscheidungstabellen:**

Dienen zur Darstellung von komplexen, von mehreren Einflussfaktoren gesteuerten Programmabläufen.

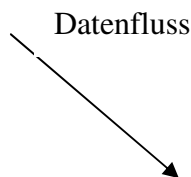
Bedingungsbeschreibungen	Bedingungseintragungen
Aktionsbeschreibungen	Aktionseintragungen

Beispiel: Flugticketbuchung

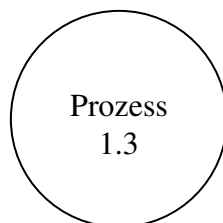
verlangt	1.Klasse	J	J	N	N
verfügbar	1.Klasse	J	N	-	-
verfügbar	Economy	-	-	J	N
ausgeben	1.Klasse	J	-	-	-
ausgeben	Economy	-	-	J	-
auf Warteliste setzen		-	J	-	J

- **Datenflussdiagramme (Data Flow Diagrams, DFD's)**

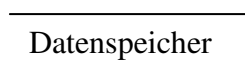
- Gerichtete Linie stellt Fluss von Informationen oder Objekten dar



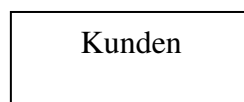
- Kreis stellt eine Aufgabe oder einen Prozess dar, bezeichnet durch einen Namen und eine Referenznummer



- Zwei parallele Linien stellen einen Speicher für Daten oder Objekte dar, bedeutet zeitliche Verzögerung für die Inhalte



- Rechteck stellt einen Punkt dar, wo Daten entstehen oder enden aus der Sicht des betrachteten Systems, bedeutet die Grenze des Systems (Schnittstelle anderen).



- Zeichnungen...

- Weitere Verfahrensschritte:

- Verfeinerung der Prozesse in weiteren Diagrammen, solange es möglich ist einen Prozess durch einfachere Teilprozesse zu definieren/zu erklären.
- Erstellung des Data Dictionary (alphabetisch geordnete Liste aller externen Partner, Datenspeicher, Datenflüssen mit genauer Definition)

z.B. Projektanforderung = Name des Anfordernden + Kalenderdatum +  
{Projektziel} + {Projektnutzen} + {Betroffenes System} +  
Projektpriorität + (Benutzerrepräsentanten)

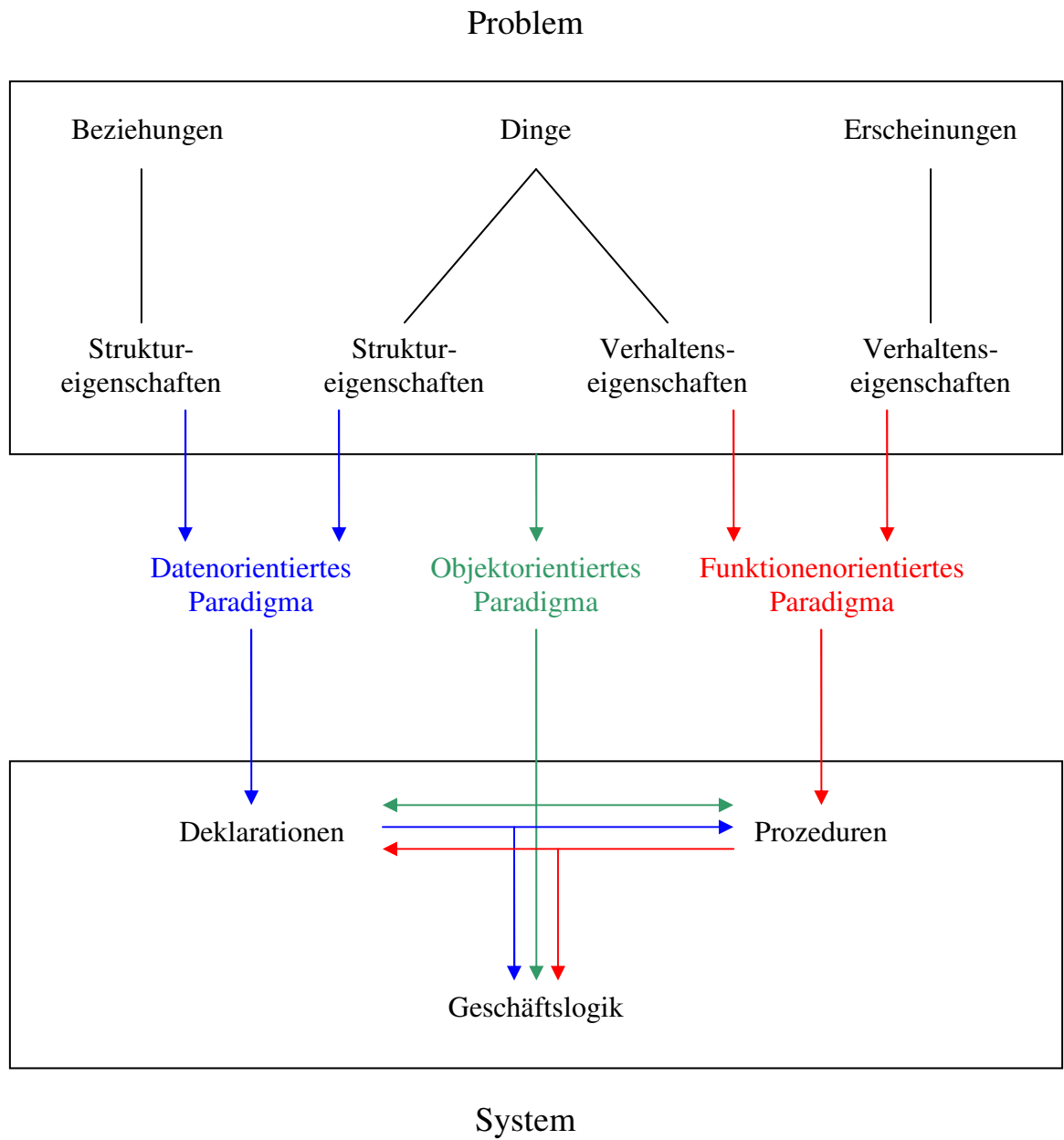
- Prozesse, zu deren Erklärung es kein Detailldiagramm gibt, werden verbal beschrieben.

- **Pseudocode**

Erklärt einen Algorithmus in einer Form, die an eine Programmiersprache angelehnt ist, aber nicht syntaktisch exakt sein muss.

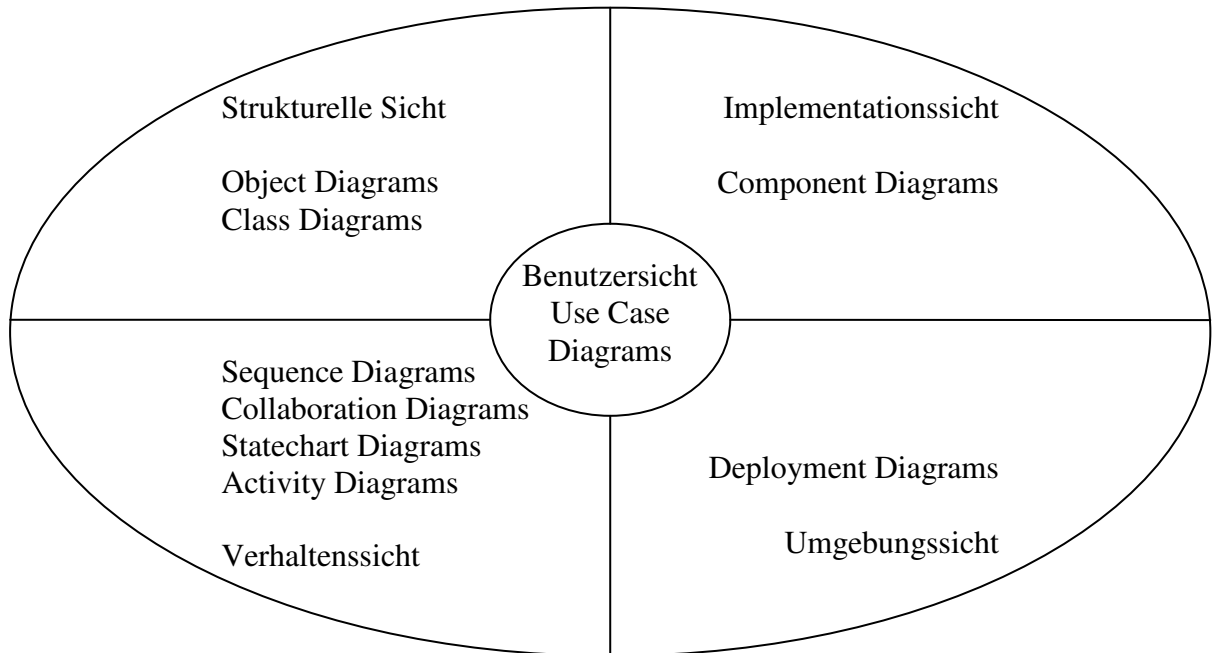
## 6.4 Objektorientierte Modellierung

- Problem → System

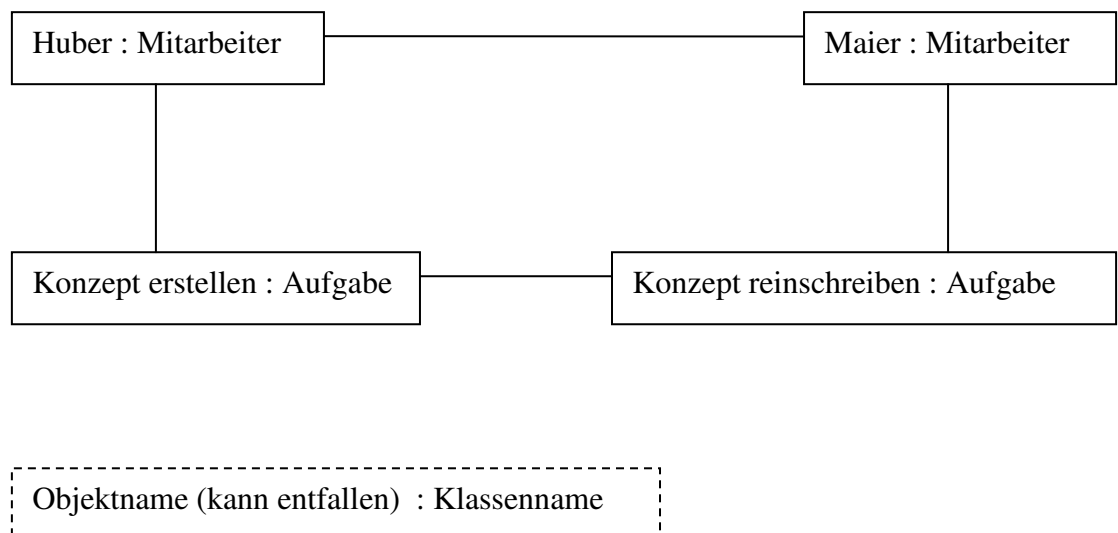


- **UML – Unified Modeling Language**

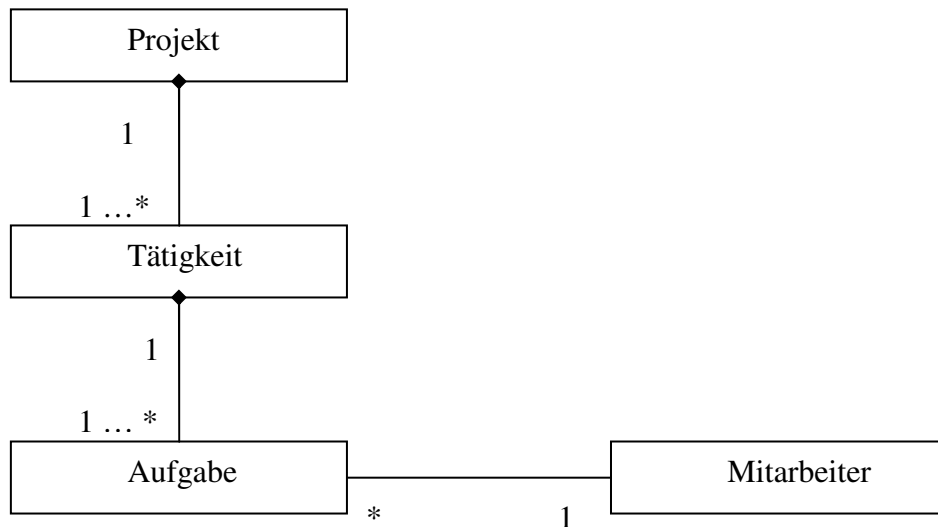
- Standard für die Beschreibung Objektorientierter Programme:



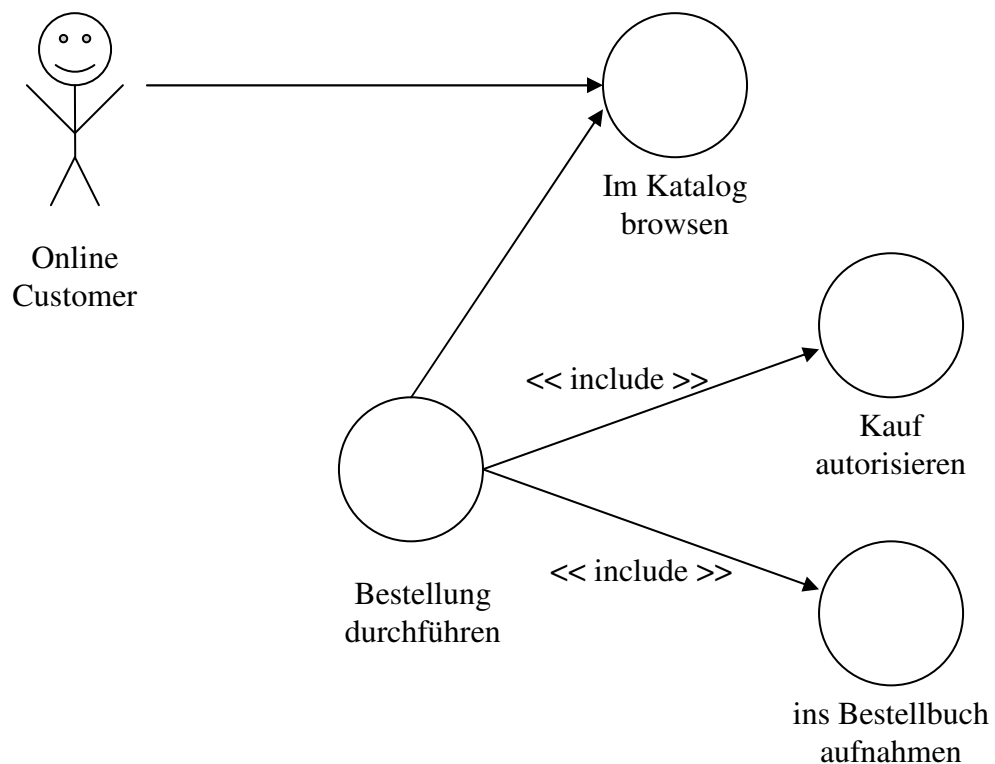
- Beispiel Object Diagram:



- Beispiel zu Class Diagramm:



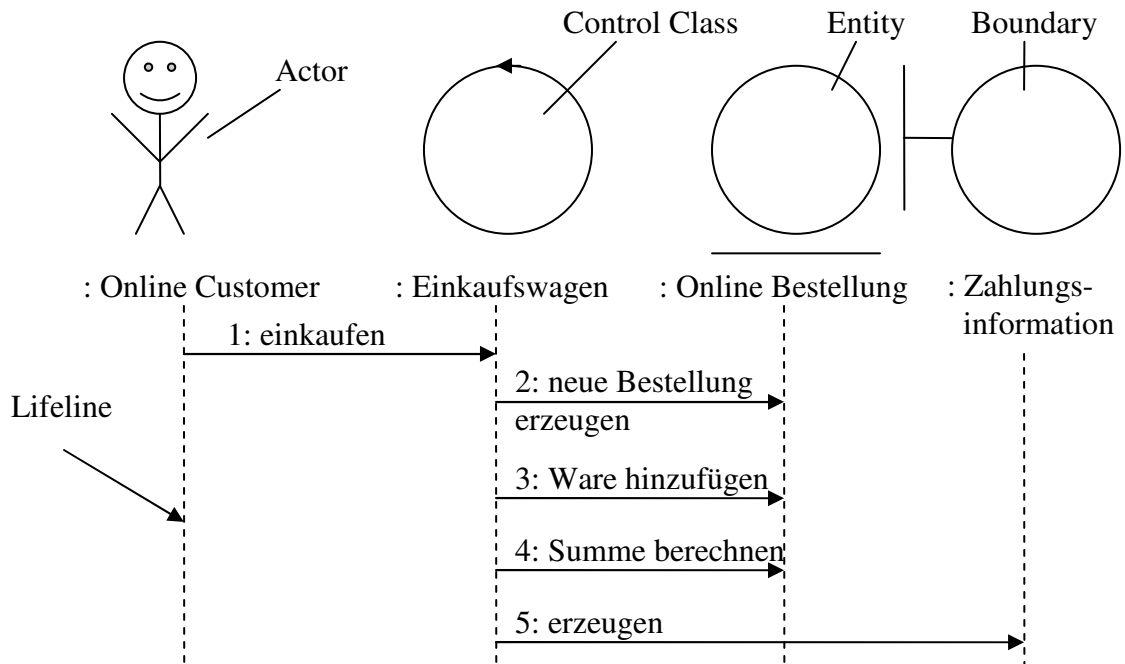
- Beispiel E-Commerce Use Case:



- Sequence Diagram:

Beschreibt einen konkreten Ablauf z.B.: eines Geschäftsfalles.  
In der Kopfzeile stehen die Beteiligten (Objekte), darunter stehen die Vorgänge die zwischen diesen Objekten ablaufen, chronologisch nummeriert und angeordnet von oben nach unten.

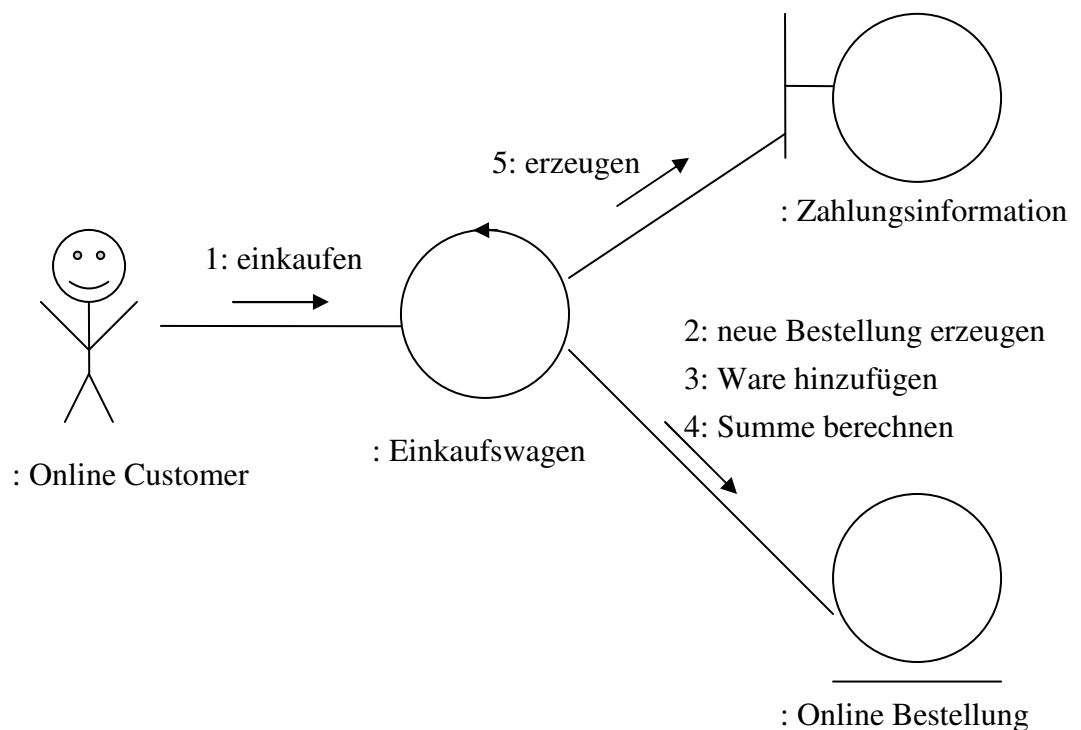
- Beispiel Sequence Diagramm:



- Collaboration Diagram:

Stellt dar, wie Objekte zusammenwirken, um einen konkreten Geschäftsfall abzuwickeln.

- Beispiel Collaboration Diagram:





## 6.5 Entscheidung für ein DV – Verfahren

- **Standardsoftware / Individualsoftware:**

Vorteile von Standardsoftware	Nachteile
billig, Fixpreis schnell verfügbar geringeres Risiko Dokumentation vorhanden Wartung meist gesichert	enthält überflüssiges Anpassungsaufwand Nicht effizient den konkreten Anforderungen nicht optimal angepasst

- Die Entscheidung für ein DV-Verfahren darf nicht später erfolgen als in der Designphase.
- **Systemauswahlverfahren:**
  - Gesamtbeurteilung wird in Teilbeurteilung von einzelnen Merkmalen zerlegt.
  - Jedes Merkmal gehört zu einer der drei Gruppen:
    - **Qualitative Merkmale:** j/n, vorhanden / nicht vorhanden
    - **Ordinale Merkmale:** Reihenfolge von verschiedenen Ausprägungen, nur diese können angenommen werden.  
z.B.: Schulnoten
    - **Quantitative Merkmale:** Sind messbar, beliebige Werte können angenommen werden. z.B.: Preis, Abmessungen
  - Tipps:
    - Keine wesentlichen Eigenschaften vergessen
    - Auswahl der Kriterien vor der Beurteilung
    - Trennung von objektiven und subjektiven Beurteilungen
    - Gegenüberstellung aller Teilbeurteilungen in einer Nutzwertmatrix
    - Sensibilitätsanalysen: Wie hängt die Entscheidung von einer Teilbeurteilung ab?
    - Verzichtsrangfolge: Ändert der Verzicht auf ein Kriterium die Entscheidung?

- Beispiel Nutzwertmatrix:

Kriterium		Gewicht -ung	Varianten					
			A		B		C	
Kosten	Einmalig	10	1.0	10.0	0.8	8.0	0.6	6.0
	Laufend	20	0.8	16.0	0.8	16.0	0.8	16.0
Hardware	CPU	10	1.0	10.0	1.3	13.0	1.3	13.0
	Peripherie	15	1.3	19.5	1.3	19.5	1.3	19.5
Software	Betriebssystem	5	1.25	6.25	1.0	5.0	1.0	5.0
	Dienstprogramme	5	1.25	6.25	1.0	5.0	1.0	5.0
	Compiler	10	1.0	10.0	1.0	10.0	1.25	12.5
	Anwendungsprogramme	10	1.5	15.0	1.0	10.0	1.0	10.0
Service	Wartung	8	1.0	8.0	1.3	10.4	1.3	10.4
	Unterstützung	4	1.0	4.0	2.0	8.0	1.25	5.0
	Schulung	3	1.25	3.75	1.0	3.0	1.0	3.0
Summe				108.75		107.9		105.4
Rang			Erster		Zweiter		Dritter	

- **Systemvergleichsverfahren:**

- Dienen zum Vergleich der technischen Leistungsfähigkeit (Performance).

- **Formeln, Schreibtischtiming:**

Berechnung der Ausführungszeit (Analyse der Anwendungsprogramme und Aufsummieren von Teilzeiten). Billig aber ungenau.

- **Instruction mixes:**

Satz von Instruktionen, der repräsentativ für das Anwendungsprogramm ist. Messen der verschiedenen Durchführungszeiten.

- **Kernels:**

Kleine Programme, deren Ausführungszeit auf die Leistungsfähigkeit der Hardware schließen lässt.

z.B.: Matrixinversion, Quadratwurzelberechnung, Polynomentwicklung

- **Standard Benchmarks:**


Werden von Herstellern verwendet um Unterschiede zwischen verschiedenen Modellen darzustellen.

- **Echte Benchmarks:**

Teil, des Anwendungsprogrammes wird für jede Systemvariante implementiert und die Ausführungszeit gemessen.

Aussagekräftig aber sehr teuer.

- **Simulation:**

Nachbildung des Systems (inkl. Benutzer) auf einem  deren.

- **Wahl der Programmiersprache**
  - Eignung für die Problemlösung
  - Kompatibilität
  - Verfügbarkeit von Personal
  - Kosten: Entwicklungskosten – Betriebskosten
  - Benutzerfreundlichkeit der Entwicklungsumgebung
  - Rücksichtnahme auf zukünftige Entwicklungen

## **7. Projektfreigabe**

- Planung ist vollkommen abgeschlossen
- Danach werden keine Änderungen mehr akzeptiert
- Letzter möglicher Zeitpunkt für einen Projektabbruch
- Bei Projektabbruch bisherige Ergebnisse geordnet ablegen

## **8. Systemimplementierung**

- Besteht aus:
  - Detailentwicklung
  - Programmierung
  - Codierung
  - Einzeltest
- Detailentwicklung:  
Weitere Verfeinerung des Konzepts (z.B.: Namen von Variablen, Dateien, genaue Algorithmen festlegen)
- Herstellung von Sourcecode
- Codierung:  
Umwandlung der Sourcen in ausführbare Module
- Einzel- oder Modultest:  
Programmierer testet seinen Teil unabhängig vom Rest der Applikation.
- Aufgaben des Projektleiters:
  - Kontrolle von Programmierung, Codierung, Dokumentation
  - Kontrolle des Zeitplans
  - Arbeitseinteilung in Programmierteams  
(Chefprogrammierer ist Teamleiter und berichtet dem Projektleiter)
  - Einhaltung von Firmennormen (Kommentare, Formatierung)
  - Strukturierte Programmierung

## 9. Strukturiertes Testen

- Definition: Strukturiertes Testen ist die Planung, Herstellung und Durchführung von Messungen, die zeigen sollen, dass die tatsächlichen Eigenschaften eines Systems mit den geforderten Eigenschaften übereinstimmen.
- Grund für Testen ist die Einschränkung des Risikos.
- Stufen von Tests:

Modultest	\	Entwicklungsorientierte Tests
Integrationstest	>	
Systemtest	/	
Operationaler Test	\	Anwendungsorientierte Tests
Akzeptanztest	/	

Als Regressionstest bezeichnet man die Wiederholung eines Tests.

Als Testfall bezeichnet man eine Zusammenstellung von Eingabewerten mit den erwarteten Ergebnissen.

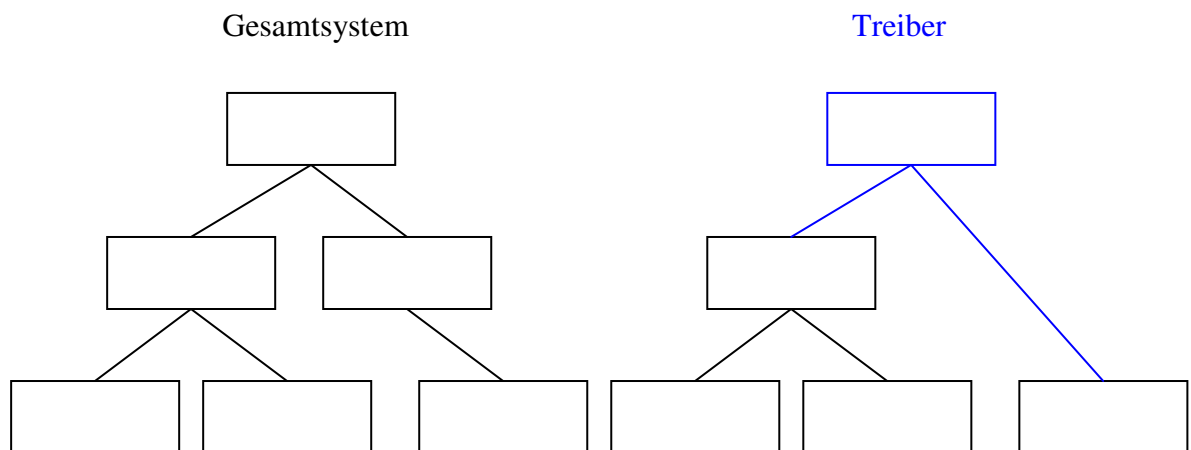
### 9.1 Methodisches Vorgehen

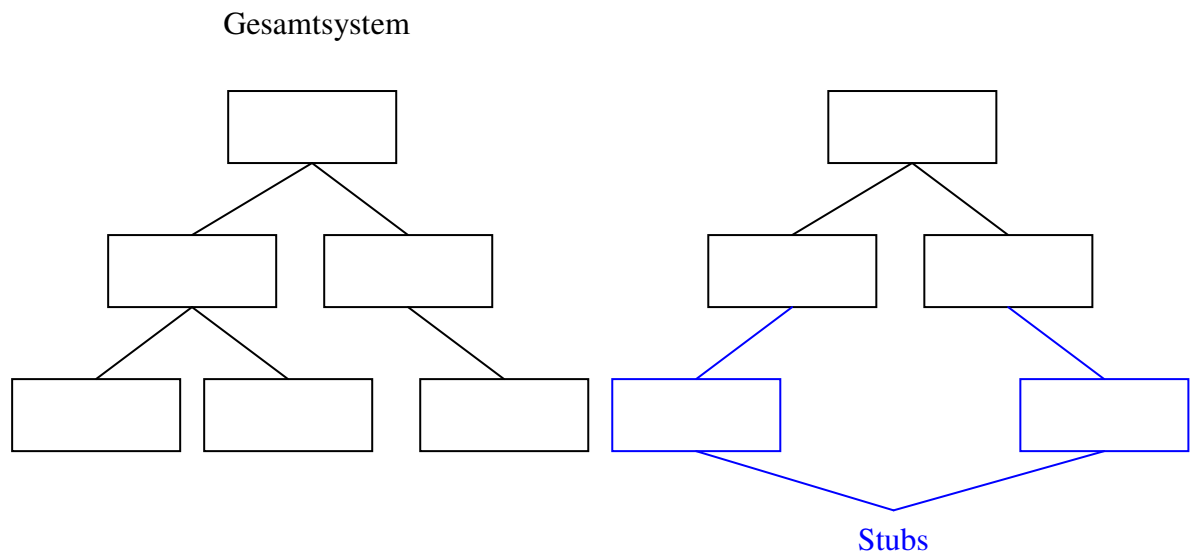
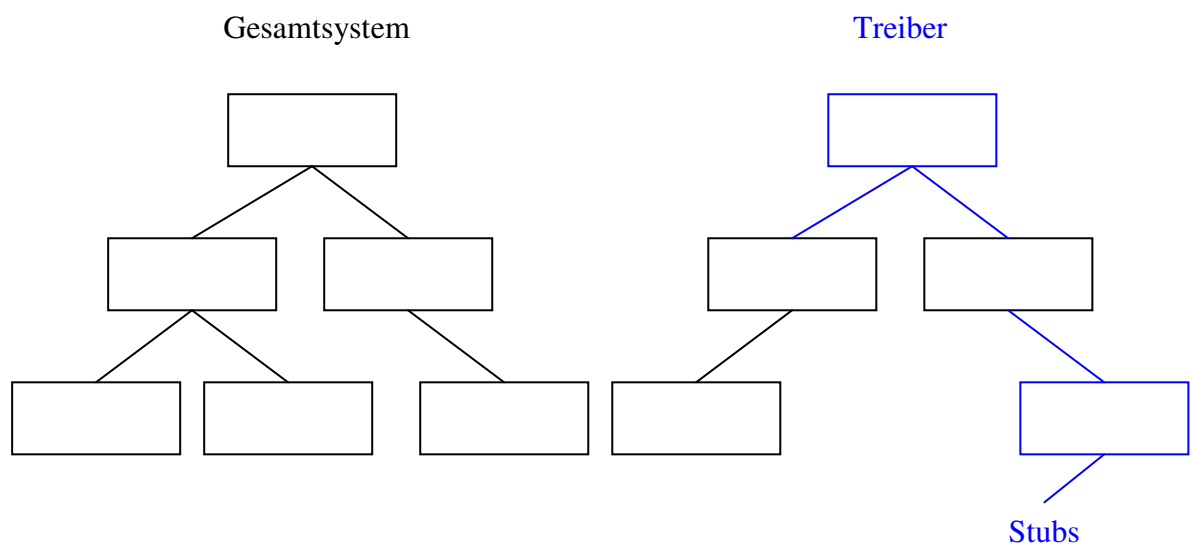
- Testplanung:
  - Planung des generellen Vorgehens, der Hilfsmittel und der Termine.
  - Festlegung der Eigenschaften, die zu testen sind.
- Testherstellung:
  - Festlegung von Testfällen
  - Detaillierung des Testplans
- Testdurchführung:
  - Ausführung der Testfälle, Beobachtung von Systemabstürzen
  - Auswertung von Testaufwand und -ergebnis

### 9.2 Testplanung

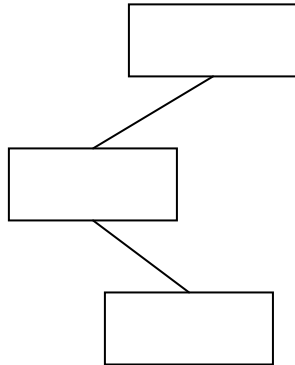
- **Akzeptanztest:**
  - auch Übergabetest, Abnahmetest
  - Soll das Vertrauen in das System wecken und verstärken.
  - Voraussetzungen:
    - Alle Benutzeranforderung erfüllt
    - System ist fertig für Bedienung durch Anwender
    - Benutzerdokumentation ist fertig
    - Schulung ist abgeschlossen
    - System läuft zuverlässig
- **Systemtest:**
  - Betrifft das gesamte System
  - Soll zeigen, dass das System fertig ist für den Akzeptanztest

- Wichtigste Inhalte sind:
  - Funktionaler Test: Sind alle funktionalen Anforderungen erfüllt?
  - Performancetest: Sind Anforderungen an die Ausführungszeit erfüllt?
  - Stress-Test: Bringt das System an die Grenze der Leistungsfähigkeit
  - Sicherheitstest: Arbeitet das System sicher?
  - Zweckmäßigkeitstest: Erfüllt das System den geforderten Zweck?
  - Zuverlässigkeitstest: Arbeitet das System auch nach langer Betriebszeit normal?
  - Destruktiver Test: Bringt das System zum Absturz
- **Integrationstest:**
  - Überprüft mehr als ein, aber weniger als alle Module
  - Ziel: Fehler in den Schnittstellen zwischen den Modulen frühzeitig zu isolieren
  - Um ablauffähige Programme aus mehreren Modulen zu erhalten, müssen fehlende Module durch Hilfsmodule ersetzt werden
  - Als Treiber bezeichnet man ein Hilfsmodul, das ein fehlendes höher stehendes Modul ersetzt (Benutzerschnittstelle).
  - Als Dummy oder Stub bezeichnet man ein Hilfsmodul, das ein fehlendes tiefer stehendes Modul ersetzt (Arbeitsmodul).
- Für die Reihenfolge der Integration gibt es verschiedene Varianten:



**Top-Down:****Integration nach Verfügbarkeit:**

Fertiggestellte Module werden mit Treibern und Stubs integriert.

**Fadenintegration:**

Alle Module, die für eine wichtige Teilaufgabe erforderlich sind, werden hergestellt und integriert.

**Integration nach Risiko:**

Module mit großem Risiko werden zuerst implementiert, integriert und getestet.

**9.3 Testdesign**

- Als Abdeckungsgebiet eines Tests bezeichnet man die Systemelemente, die Gegenstand des Tests sind.
- **Maßzahlen für Tests:**
  - Breite eines Tests: Ist das Verhältnis der getesteten Elemente des Abdeckungsgebietes zur Gesamtzahl von Elementen des Abdeckungsgebietes.
  - Tiefe eines Tests: Ist die Anzahl von Testfällen zu einem Element des Abdeckungsgebietes.
  - Effizienz eines Tests: Ist ein Maß dafür, wie gut ein Test geeignet ist, einen vorhandene Fehler aufzuzeigen.
- **Arten von Abdeckungsgebieten:**
  - Anforderungsorientierte Abdeckungsgebiete:
    - Funktionalität
    - Performance
    - Benutzerfreundlichkeit
    - Sicherheit
    - Zuverlässigkeit
    - Formate von Ein- und Ausgabe
  - Designorientierte Abdeckungsgebiete:
    - Übersichtlichkeit
    - Vollständigkeit
    - Konsistenz
    - Einhaltung von Normen

- Codeorientierte Abdeckungsgebiete:
  - Anweisungen
  - Verzweigungen
  - Pfade durch das Programm
  - Schleifen
  - Abfolge von Daten
  - Kommentare
  - Codierrichtlinien

- **Testspezifikation:**

- Testdesign-Spezifikation: Legt die prinzipielle Testidee fest.
- Testfall- S.: Enthält Festlegung von Input und erwartetem Ergebnis.
- Testdurchführungs- S.: Beschreibt die nötigen Abläufe zur Testdurchführung.

## 9.4 Testdurchführung:

- **Testdatengenerierung:**

- Kopie von Echtdaten oder Teilen davon
- alte Testdaten
- zufällig generierte Daten
- manuelle Eingabe
- Mithilfe des Anwenders

- **Testberichte:**

- Testlog: Chronologische Liste der Testaktivitäten
- Testzwischenfallbericht: Detaillierte Beschreibung von Vorfällen die untersucht werden müssen (falsche Ergebnisse, Programmabstürze, ...)
- Testabschlussbericht: Zusammenfassung der Testaktivitäten, Auswertung von Aufwand und Resultat des Tests

## 9.5 Abschlussbemerkungen

Probleme in der Praxis	Abhilfe
Zu wenig Ressourcen	Auswahl: nach Risiko / Dringlichkeit
Undokumentierte Software	z.B.: codeorientiert testen
Sehr große Zahl möglicher Eingabewerte (Kombinatorischer Input)	Stichproben, Zufallsgenerator
Komplexe Software (z.B.: Echtzeitsysteme)	Designorientiert testen
Unbekanntes Ergebnis	Historische Daten, Parallelentwicklung



## 10. Projektabschluss

### 10.1 Umstellung auf ein neues System:

• <b>Gesamtumstellung</b>	• <b>Teilumstellung</b>
Sämtliche Teile des neuen Systems werden in Betrieb genommen.	Das neue System wird in Teilen nach und nach in Betrieb genommen
Geht nicht bei Echtzeitsystemen und Steuerungsaufgaben	Geht nur, falls das System teilbar ist.
+ keine Inkompatibilitäten	+ weniger Risiko
+ geringerer Zeitbedarf	+ Fehler leichter lokalisierbar
+ bessere Hardwareausstattung	+ geringere Anforderungen an das Personal

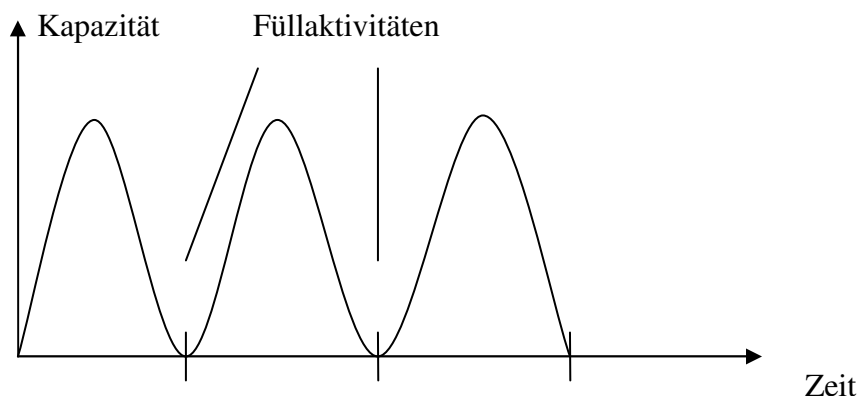
• <b>Stichtagsumstellung</b>	• <b>Parallellauf</b>
Per Stichtag werden Teile des alten Systems (oder das gesamte System) außer Betrieb genommen und die neuen Teile in Betrieb.	Teile des alten Systems (oder das gesamte System) bleiben eine Zeit lang parallel mit dem neuen System in Betrieb.
- sofortige Auswirkungen von Fehlern + billiger + keine doppelte Arbeit + große Identifizierung mit dem neuen System	Nicht möglich bei Steuerungsaufgaben und Echtzeitsystemen.  + geringeres Risiko + altes System als Backup

### 10.2 Abschlussreview

- Rückblick auf das Gesamtprojekt
- Diskussion von Erfolgen und Problemen
- Was sollte besser gemacht werden?
- Vorbereitung für Folgeprojekte
- Auch abzuhalten wenn Projekt abgebrochen wird

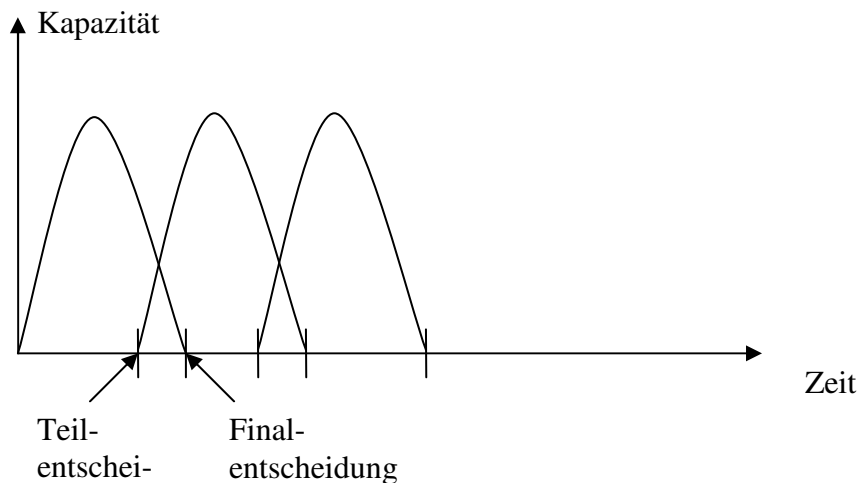
### 10.3 Phasenabfolge

- **Scharfe Trennung der Phasen:**



Folgephase beginnt erst nachdem die vorige vollständig abgeschlossen ist.

- **Überlappende Projektphasen:**



Folgephase beginnt nach einer Teilentscheidung über das Ende der vorigen Phasen.  
Daraus ergibt sich eine kürzere Projektdauer.  
Risiko: Projektabbruch, Änderungen

## **11. Wartung und Änderung**

### **11.1 Ursachen für Änderungen**

- **Von außen:**
  - Technische Weiterentwicklung, Änderungen Kosten/Nutzen
  - Organisatorische Änderungen (neue Filialen, Märkte, Produkte)
  - Gesetzliche/Rechtliche Änderungen
- **Von innen:**
  - erkannte Inkompatibilitäten
  - Leistungssteigerungen

### **11.2 Implementierung von Änderungen**

- Jede Änderung an den Programmen (durch Programmierer) wird begleitet von:
  - Änderungen in der Spezifikation (Konzept) und
  - Änderungen im Benutzerhandbuch
- Diese werden durch den Systemanalytiker vorgenommen.

### **11.3 Validation (Überprüfung) einer Änderung**

- Jede Änderung kann zu einer Kette neuer Fehler führen.
- Alle Elemente im Wirkungsbereich einer Änderung müssen wieder getestet werden.
- Test von einer separaten Testmannschaft durchführen

### **11.4 Programmänderungsverfolgung**

- Bei jeder Änderung neue Versionsnummer und Datum im Sourcefile vermerken.
- Jede neue, geänderte oder gelöschte Zeile mit einem Kommentar versehen.

## 11.5 Wartungsrichtlinien

- Zuerst den Aufbau der vorhandenen Software untersuchen und einen stufenweisen Plan für die Überarbeitung aufstellen.
- Für jede Änderung mindestens zwei Alternativen überlegen:
  - die schnellste und billigste Lösung
  - die sicherste, wo die vorhandenen Strukturen am wenigsten darunter leiden
- Wähle die billigste Variante, falls das Modul demnächst überarbeitet (ersetzt) wird.
- Wähle die sicherste Variante, falls das Modul in nächster Zeit nicht mehr geändert wird.
- Überlege die Auswirkung der Änderung, vor der Durchführung und überprüfe die Auswirkung nach der Durchführung.

## 11.6 Sieben Gebote der Software-Wartung

- Jede Änderung muss von einem kompetenten Gremium genehmigt werden.
- Jede Änderung muss auf ihre Fernwirkungen überprüft werden.
- Änderungen dürfen nur im Quellcode gemacht werden (keine Patches).
- Jedes von der Änderung betroffene Dokument muss gleichzeitig geändert werden.
- Jedes geänderte neue Modul muss wieder getestet werden.
- Jedes geänderte neue Modul muss wieder integriert werden.
- Jede Änderung muss von der Qualitätskontrolle oder vom Anwender abgenommen werden.

## 11.7 Software Evolution

- Nützliche Software spiegelt Vorgänge der realen Welt wieder und muss ihnen ständig angepasst werden, sonst wird die Software zunehmend nutzloser.
- Jede Erweiterung und Änderung zu bestehender Software birgt die Gefahr, dass die Komplexität zunimmt und die ursprüngliche Struktur gestört wird.
- Die Evolution komplexer Softwaresysteme unterliegt gewisser Gesetzmäßigkeiten, die das Wachstum (Änderungsrate) und die Produktivität der Entwickler, über längere Zeiträume hinweg betrachtet, konstant halten.

## 11.8 Software-Dialektik

- Je mehr Geld für zusätzliche Module ausgegeben wird, umso weniger gibt es für den Test. Also wird das System instabiler.
- Je instabiler ein System wird, umso mehr Geld muss für Tests ausgegeben werden. Also wird das Systemwachstum gebremst.
- Je stabiler ein System ist, desto größer ist der Drang, es funktional zu erweitern.
- Als Konsequenz ist das Wachstum eines Softwaresystems über mehrere Jahre konstant.

## 11.9 Management-Konsequenz

- Der Software-Manager muss die Grenzen des Wachstums der Software erkennen und beachten.
- Ein langfristiger Plan ist zu erstellen, um eine stetige, regelmäßige Weiterentwicklung des Software festzulegen.
- Kurzfristige Anwenderwünsche sollten diesen Plan möglichst nicht stören.

- Plane keine Perioden überhöhter Produktivität.
- Nach einer Phase überhöhter Produktivität plane eine Zeit für Fehlerverbesserungen.
- Messe die Produktivität, um besser planen zu können.
- Jedes Release soll gegenüber dem vorangehenden etwa die gleiche Anzahl Neuerungen haben.
- Zwischen aufeinander folgenden Releases soll einige Zeit verstreichen.

### **11.10 Zusammenfassung**

- Software-Evolution muss geplant werden (Release Konzept)
- Software-Evolution muss kontrolliert werden (Release Abnahme)
- Software-Evolution muss gesteuert werden

## **Dokumentation**

### **1. Gestaltung von Dokumenten**

#### **1.1 Layoutgestaltung**

- einheitliches Format (DIN A4)
- Gleichbleibende Anzahl von Kopf- und Fußzeilen  
(Seitennummer, Seitenanzahl, Projektname, Autor, Datum, Version, ...)
- jedes Kapitel beginnt auf einer neuen Seite
- Nummerierung von Abbildungen, Graphiken, Tabellen, Schirmbildern in hierarchischer Form
- Inhaltsverzeichnis  
(Kapitel, Abschnitte mit Seitenzahlen, Reihenfolge wie im Dokument)
- Stichwortverzeichnis: alphabetisch sortierte liste wichtiger Begriffe mit Seitenverweisen
- Glossar: Begriffserklärungen
- Deckblatt (Vorspann)
  - Projektname (Produkt-)
  - Art des Dokumentes
  - Datum, Version
  - Versionen History
  - Autoren
  - Qualitätskontrolle
  - Status (Grad der Fertigstellung):
    - IN ARBEIT
    - FERTIGGESTELLT
    - ABGENOMMEN
    - FREIGEgeben
  - Vertraulichkeit
  - Copyright

#### **1.2 Sprachliche Gestaltung**

- Fehler vermeiden
- eindeutige Begriffsbildung
- kurze Sätze
- keine eingeschobenen Nebensätze
- kurze Wörter
- eindeutig formulieren
- keine Modalverben (müsste, sollte, würde)
- besser aktiv formulieren als passiv
- nicht zu wortreich formulieren

### 1.3 Didaktische Gestaltung

- Fachausdrücke erklären
- wichtige **Begriffe** hervorheben
- Aufzählungen hervorheben
- optische Trennung
- Leerzeilen zwischen Abschnitten
- Leerzeichen nach Satzzeichen
- nicht mehrere verschiedene Inhalte in einem Satz zusammenfassen
- graphische Unterstützung zur Erklärung
- komplexe Inhalte in anderer Form wiederholen
- bei Verweisen nicht nur die Nummer verwenden
- Gliederungen und Aufzählungen dürfen nicht zu viele Punkte enthalten (etwa sieben als Maximum)

## 2. Bildschirmgestaltung

### 2.1 Aufgaben eines Bildschirms

- Funktionsauswahl
- Eingabe von Werten
- Ausgabe von Werten
- Ausgabe von fixen Texten

### 2.2 Einteilung des Bildschirms

- **Kontrollbereich:**
  - Name des Systems
  - Name des Bildschirms
  - Nummer der Maske
  - Nachrichten und Fehlermeldungen
  - Möglichkeit von Befehlseingaben
- **Anwenderbereich:**
  - Bereich für die Ein-/Ausgabe von Werten

### 2.3 Funktionsauswahl

- Auswahl aus den angebotenen Alternativen:
  - (Cursorposition, Mausbewegung)
  - Menüauswahl (gut geeignet bei wenigen Alternativen, ungeübten Anwendern)
- Eingabe von Befehlscodes:
  - flexibler
  - schneller für geübte Anwender
  - Shortcuts
- empfehlenswert ist es beide Alternativen anzubieten

## 2.4 Ein- und Ausgabebildschirme

- Je Bildschirm soll nur eine Datenart bearbeitet werden  
z.B.: Trennung Stammdaten von Bewegungsdaten
- übersichtliche Gestaltung
- nicht zu viele Informationen (z.B. Aufteilen auf Karteiblätter)
- wiederkehrende Angabe auf jedem Bildschirm auf derselben Stelle im selben Format (Schriftart, Farbe)
- Anpassung an gedruckte Formulare
- missverständliche und ungebräuchliche Ausdrücke vermeiden
- Keine Abkürzungen verwenden
- Mehrfacheingabe von Daten vermeiden
- bei längeren Ausgaben (mehrsseitig) sollte eine gezielte Auswahl möglich sein
- Schirminhalte sollten ausgedruckt werden können

## 2.5 Textbildschirm

- leicht verständlich formulieren
- bedarfsgerecht nur aktuell benötigte Informationen anbieten

## 2.6 Farbverwendung

- Nicht zu viele Farben (max. 4)
- grelle Farben sparsam verwenden
- gleiche Information immer in derselben Farbe (z.B.: Eingaben schwarz, Fehler rot)
- individuelle Anpassung sehr wichtig

## 2.7 Diskussionsgestaltung

- **Zielsetzung:**
  - Hohe Akzeptanz
  - Geringer Lernaufwand
  - Verminderte Fehlerquote
  - Hohe Produktivität
- **Qualitätskriterien:**
  - Aufwand für Erlernen einer Aufgabe (erstmalig, im Wiederholungsfall)
  - Aufwand zur Durchführung einer Aufgabe
  - Aufwand für die Behebung von Bedienungsfehlern
- **Einflussfaktoren bei der Dialoggestaltung:**
  - Aufgabenstellung:
    - Zielfestlegung
    - Unterteilung in Teilaufgaben
    - zeitliche Reihenfolge
  - Benutzer
    - EDV Erfahrung
    - fachliches Wissen
    - Lernwilligkeit, Flexibilität

- **Unterstützung bei der Dialoggestaltung:**
  - Welche Hilfsmittel sind für die Abarbeitung notwendig?
    - Information, Bewilligung
    - Systemteil
  - Befragen der Betroffenen aus der Fachabteilung, ev. ins Projektteam aufnehmen.
- **Hilfe für den Benutzer:**
  - Benutzer soll jederzeit feststellen können,
    - wo im System er sich befindet
    - was das System gerade macht
    - was seine Aufgabe ist
    - wie er Eingabefehler verbessern kann

### **3. Inhalt und Aufbau eines Lastenheftes**

#### **3.1 Allgemeines**

- Ein Lastenheft kann als Grundlage für eine Ausschreibung dienen (Bitte um Angebot)
- Lastenheft ist die erste grobe Produktbeschreibung

#### **3.2 Aufgabe**

- Zusammenfassung aller technischen Basisanforderungen aus der Sicht des Auftraggebers.
- Bewusste Konzentration auf die fundamentalen Eigenschaften.
- Beschreibung auf hinreichendem Abstraktionsniveau.

#### **3.3 Adressaten**

- Auftraggeber: Benutzer, Geschäftsführer
- Auftragnehmer: Projektleiter, Produktdefinierer, Systemanalytiker

#### **3.4 Inhalt**

- Fundamentale Produkteigenschaften
- nichts über den Prozess
- Beschreibung des WAS, nicht des WIE

#### **3.5 Form**

- Vorgegebenes, standardisiertes, grobes Gliederungsschema mit festgelegten Inhalten.

#### **3.6 Sprache**

- verbale Beschreibung
- angepasst an den Leser

#### **3.7 Didaktik**

- verständliche Formulierung
- leicht lesbar und erfassbar



### 3.8 Zeitpunkt

- Erstes Dokument, das ein Produkt beschreibt
- vor Voruntersuchung, vor Vertragsabschluss

### 3.9 Umfang

- meist nur wenige Seiten

### 3.10 Gliederungsschema

- **1.) Zielbestimmung:**

Beschreibt, welche Ziele durch den Einsatz des Produktes erreicht werden sollen.

- **2.) Produkteinsatz:**

Beschreibt für welche Anwendungsbereiche/Zielgruppen das Produkt vorgesehen ist.

- **3.) Produktfunktionen:**

Beschreibt die Hauptfunktionen des Produktes aus Sicht des Auftraggebers, nicht die sekundären Funktionen. Enthält keine Detailbeschreibungen.

Jede Anforderung wird bezeichnet mit LF (Lastenheftfunktion) und einer fortlaufenden (eventuell hierarchischen) Nummer.

z.B.: Bankomat

LF 1: Geld abheben

LF 2: Quick laden

- **4.) Produktleistungen:**

Beschreibt Leistungsanforderungen an das Produkt, d.h. messbare Werte wie z.B. Datenmenge, Datendurchsatz (Daten/Zeit), Geschwindigkeit, Genauigkeit.

- **5.) Qualitätsanforderungen**

Beschreibt welche Anforderungen an die Qualität des Produktes besonders wichtig sind.

## 4. Inhalt und Aufbau eines Pflichtenheftes

### 4.1 Allgemeines

- Das Pflichtenheft ist ausführliche Beschreibung der Leistungen (technische, wirtschaftliche und organisatorische), die erforderlich sind, damit die Ziele des Projektes erreicht werden.
- Das Pflichtenheft dient zur Konfliktvermeidung und Konfliktbegrenzung. Es enthält eine detaillierte Beschreibung aller Punkte, die später zu unterschiedlicher Auffassung führen könnten.

- Pflichtenheft wird aus zwei Sichten erstellt:
  - die zu erbringenden Leistungen
  - die einzuhaltenden Einschränkungen

## **4.2 Aufgabe**

- Zusammenfassung aller technischen Anforderungen aus der Sicht des Auftraggebers.

## **4.3 Adressaten**

- Auftraggeber (Manager, Benutzer)
- Auftragnehmer (Projektleiter, Systemanalytiker, Designer)

## **4.4 Inhalt**

- Funktionsumfang des Produktes
- Leistungsumfang des Produktes
- Qualitätsumfang des Produktes
- Genaue Formulierung (Teil eines juristischen Vertrages)
- Vertragliche Beschreibung des Lieferumfangs
- Grundlage für die Produktabnahme
- Keine Entwurf- oder Implementierungsentscheidungen

## **4.5 Form**

- Vorgegebenes standardisiertes grobes Gliederungsschema mit festgelegten Inhalten.

## **4.6 Sprache**

- Detaillierte verbale Beschreibung

## **4.7 Didaktik**

- Möglichst leicht verständlich
- Soll dem Projektteam Einarbeitung in die Problemstellung erlauben.
- Auf die Adressaten Rücksicht nehmen

## **4.8 Zeitpunkt**

- Nach den Planungsarbeiten (Vorstudie, Voruntersuchung), aber vor der Phase Projektplanung, also bei der Projektgründung.
- Änderungen zu späteren Zeitpunkten schriftlich fixieren.

## **4.9 Umfang**

- Das Pflichtenheft muss ausreichend detailliert sein und präzise.
- Beschreibung aus der Sicht des Auftraggebers.

## **4.10 Geheimnisprinzip**

- Pflichtenheft darf nur darüber informieren, was das Produkt nach außen liefert (exportiert) und was es von außen erhält (importiert).
- Interne Abläufe bleiben geheim.

- Der Detaillierungsgrad der Informationen kann sehr unterschiedlich sein. Sehr detailliert werden im Pflichtenheft meist die Berührungspunkte mit dem Anwender beschrieben.
- Dazu gehören:
  - Eingaben:                    Datenformate, Datenträger  
                                  Nachrichten, Kommandos, Eingabemasken
  - Ausgaben:                  Datenformate, Datenträger  
                                  Bildschirmmasken, Listbilder
  - Bedienungsabläufe
  - Hardware (-konzept)
  - Betriebssystem
  - Organisatorisches:       Termine, Kosten  
                                  Abnahmebedingungen, Verantwortlichkeiten

#### 4.11 Zusammenfassung

- Die Erstellung eines Pflichtenheftes ist eine schwierige Aufgabe.
- Geforderte Eigenschaften: fachlich richtig, vollständig, widerspruchsfrei, konsistent, verständlich
- Pflichtenheft schützt vor Auffassungsdifferenzen und unkontrollierter Ausweitung des Projektumfangs.
- Ein Pflichtenheft erfüllt seine Aufgabe, wenn es
  - Dem Team für Entwurf (Design) und Implementierung alle notwendigen Informationen liefert und
  - bei der Abnahme eine eindeutige Aussage ermöglicht, ob das Produkt die gestellten Anforderungen erfüllt oder nicht.

#### 4.12 Gliederungsschema

- **Zielbestimmungen:**
  - Beschreibt die Ziele des Produkteinsatzes
  - unterteilt in Unterpunkte:
    - Musskriterien:  
Angabe der Leistungen, die vom Produkt gefordert werden
    - Wunschkriterien:  
Beschreiben Wünsche an das Produkt, die nicht unbedingt erfüllt werden müssen.
    - Abgrenzungskriterien:  
Liste von Anforderungen, die das Produkt NICHT erfüllt
- **Produkteinsatz:**
  - Anwendungsbereich:  
Angaben über den Ort des Produkteinsatzes, des Wirtschaftszweiges oder des Unternehmens.  
z.B. Textverarbeitung im Büro, Lagerverwaltung in der Lebensmittelbranche

- Zielgruppen:  
Angaben über geforderte Kenntnisse, Fähigkeiten, Lernbereitschaft des Benutzers z.B. Büromitarbeiter, LKW-Fahrer
- Betriebsbedingungen:
  - physikalische Umgebung des Systems
  - tägliche Betriebszeit
  - ständige Beobachtung durch den Benutzer oder unbeaufsichtigter Betrieb

- **Produktumgebung:**

- Beschreibt das technische Umfeld des Produktes
- Hardware:
  - Angaben über Komponenten: Prozessor, Taktfrequenz, RAM, Festplatte, Peripheriegeräte (Drucker, Scanner,...), Backupsystem
  - Angabe von Minimal- und Maximalkonfigurationen
- Software:
  - Betriebssystem
  - Datenbank
  - Laufzeitsystem
  - Utilities
- Org. – Ware:  
Organisatorische Voraussetzungen für den Produkteinsatz  
z.B. Benutzerlogin, Formulare, Betriebsabläufe, Verantwortlichkeiten
- Produktschnittstellen:
  - Einordnung des Produktes in eine bestehende oder geplante Produktfamilie
  - Definitionen für den Datenaustausch

- **Produktfunktionen:**

- Funktionale Beschreibung des Produktes aus Benutzersicht
- Unterteilung je nach Anzahl der Funktionen
- Bezeichnung: F und eine (hierarchische) Nummer:  
z.B.: Bankomat
  - L1: Bargeld abheben
  - L2: Quick laden
- Wunschkriterien werden durch ein angefügtes W markiert:  
z.B.: F4W: Störfallbehandlung
- Benutzerführung wird nicht bei den Funktionen einzeln beschrieben, sondern für das Gesamtprodukt in einem der folgenden Kapitel

- **Produktleistungen:**

- Anforderungen die zeitbezogen oder umfangbezogen sind z.B. Dialogantwortzeit, Datenumfang, Datendurchsatz, Rechengenauigkeit, Aktualität der Daten
- Messbare Größen

- Angaben erfolgen für Minimum, Maximum, Durchschnitt oder prozentuelle Grenzen (Perzentile)
- Bsp.: Die Antwortzeit beträgt im Durchschnitt 2 Sekunden, maximal 20 Sekunden, und in 90% der Fällen unter 5 Sekunden

- **Benutzerschnittstelle:**

- Beschreibung der Mensch-Maschine-Schnittstelle
- z.B.:
  - Bildschirm-Layout
  - Tastaturbelegung
  - Dialogstrategie, Benutzermodell
  - Druck-Layout

- **Qualitätszielbestimmungen:**

- z.B. in Form einer Qualitätszielbestimmungsmatrix

Qualitätsmerkmal	Wichtig	Mittel	Unwichtig
Änderbarkeit		X	
Überprüfbarkeit		X	
Verständlichkeit		X	
Wartungsfreundlichkeit	X		
Benutzerfreundlichkeit	X		
Laufzeiteffizienz			X
Speichereffizienz			X
Funktionale Korrektheit	X		
Funktionale Vollständigkeit	X		
Robustheit gegenüber Benutzer	X		

- **Globale Testfälle**

Anwendungsbezogene Testfälle, die meist mehr als eine Produktfunktion betreffen:  
Dienen als Grundlage für den Abnahmetest.

- **Entwicklungsumgebung**

Beschreibt die Umgebung, in der das Produkt entwickelt wird.

- Hardware: Beschreibt die Hardware für die Produktentwicklung
- Software: Beschreibt die Software für die Produktentwicklung
- Orgware: Beschreibt die organisatorischen Voraussetzungen
- Entwicklungsschnittstellen: Beschreibung des Übergangs von der Entwicklungsumgebung zur Produktionsumgebung.

- **Ergänzungen**

Optionaler Abschnitt (kann entfallen)

Einteilung in Unterabschnitte erfolgt je nach Bedarf z.B.:

- Wirtschaftlichkeit: Kosten, Nutzen, Rentabilität
- Entwicklungsbedingungen: Bereitstellung von Personal, Räumen, Material
- Termine
- Übergabe: Programme, Dokumentation, Stammdaten
- Installationsbedingungen: bauliche und räumliche Voraussetzungen
- Ausbildungen
- Wartung
- Rechte und Pflichten: Eigentumsvorbehalt, Gerichtsstand, Liefer- und Zahlungsbedingungen, Behördenformalitäten, Schadensersatz, Garantie, Pönale