

# SWT Übung

## Zweck:

- Wiederholung und Übung des Stoffes der Vorlesung
- (Diskussion zur Gruppenarbeit)

- Prüfungsvorleistung: „Lösen von Übungsaufgaben“

Notwendig 50% der Punkte, die für die Gruppenarbeit einschließlich der Verteidigung des eigenen Beitrages vergeben werden

# Modulhandbuch Lern- und Qualifikationsziele- SWT

## **Fachkompetenz:**

- Prinzipien und Techniken des **Software Engineering; Modellierung, Softwarearchitektur, Muster und Bibliotheken**
- **Prinzipien der Aufwandsabschätzung und Projektplanung**
- Standards

## **Methodenkompetenz:**

- Umgang mit **Entwicklungsumgebungen und –werkzeugen**
- Fähigkeit zur schnellen Einarbeitung in neue Anwendungen
- Auswahl geeigneter Prozesse und Methoden

## **Selbst- und Sozialkompetenz:**

- **Berücksichtigung nichttechnischer Rahmenbedingungen bei der Bearbeitung einer komplexen Aufgabe**
- **Organisation von Teamarbeit**
- Kernkompetenzen für Berufsqualifizierung im nichtakademischen Bereich

# Gruppenarbeit

## Gruppenaufgabe in Gruppen von 7-10 Studierenden

Themenstellung 2019/2020:

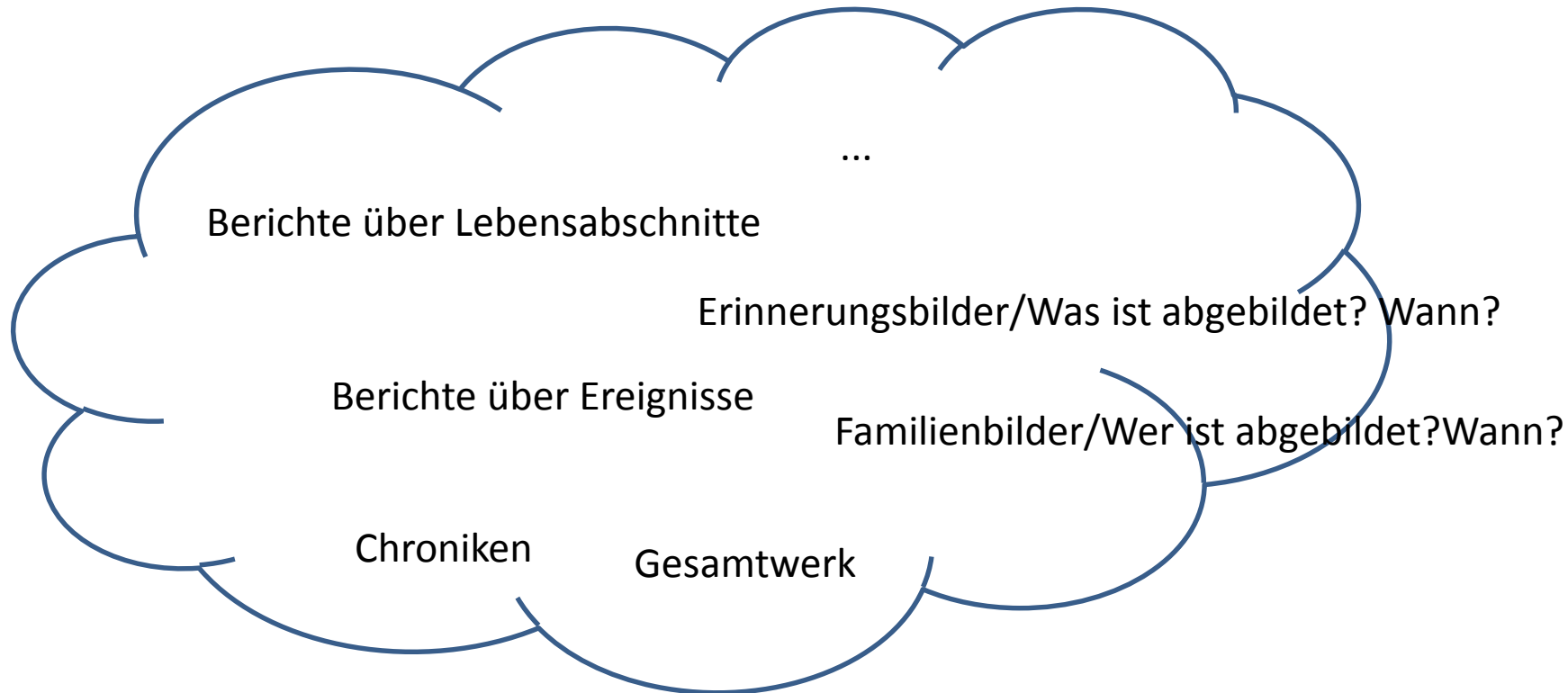
### **Software zur Unterstützung älterer Menschen zur Verwaltung und Präsentation ihrer Erfahrungen**

Berichte und Erfahrungen älterer Menschen gehen mitunter verloren. Mit der Software soll ihnen die Gelegenheit gegeben werden, ihnen wichtige Sachverhalte, Geschichten oder Bilder so zu verwalten, dass sie für die nachfolgende Generation sichtbar werden.

Ideen für die Anforderungen der Software sind aus Diskussionen mit Verwandten und Bekannten zu entwickeln. Sie sind mit den Lehrkräften abzustimmen. Diese übernehmen die Rolle der Auftraggeber.

# Gruppenarbeit

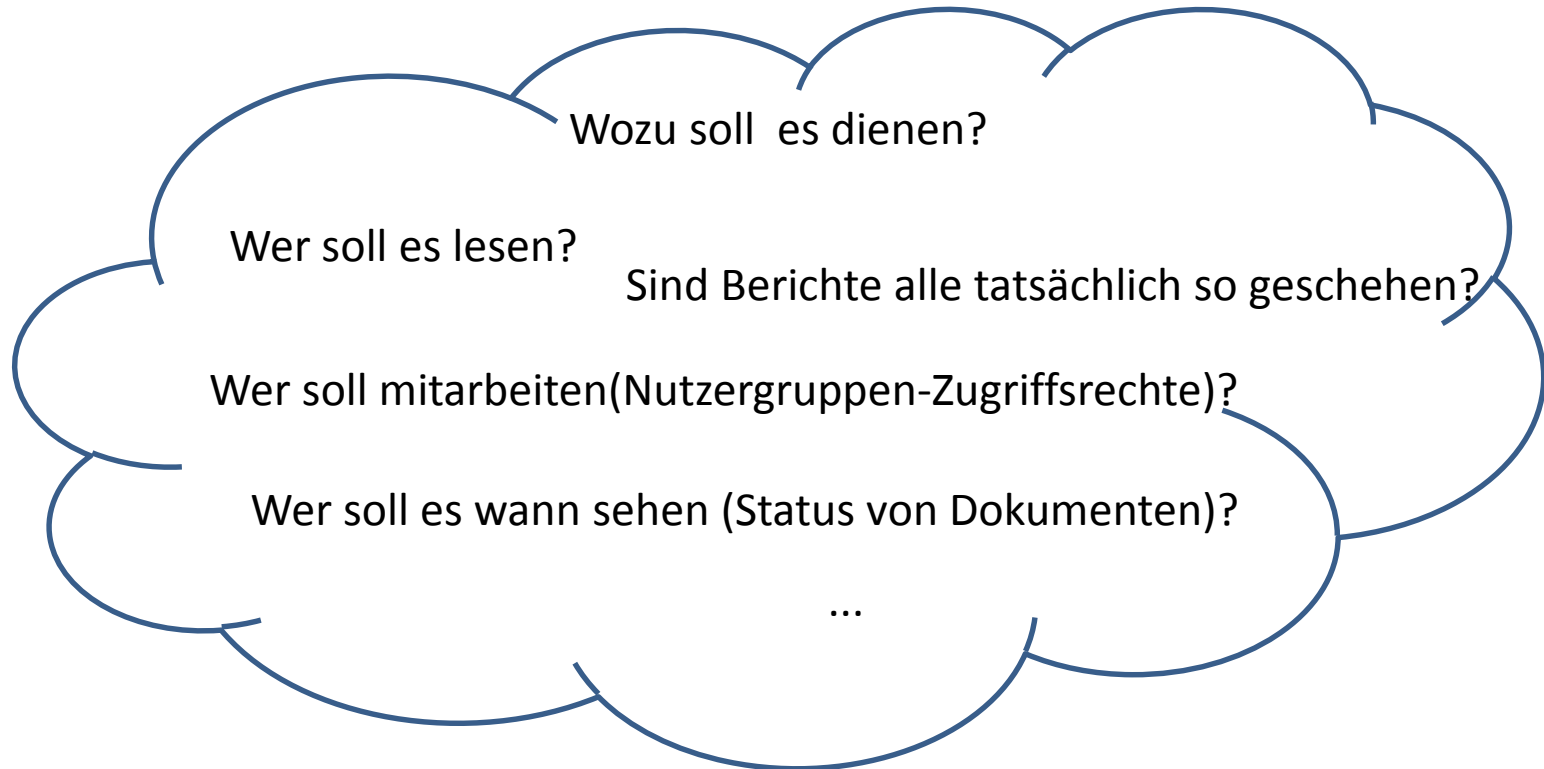
## Software zur Unterstützung älterer Menschen zur Verwaltung und Präsentation ihrer Erfahrungen



Leiten Sie Anforderungen ab!

# Gruppenarbeit

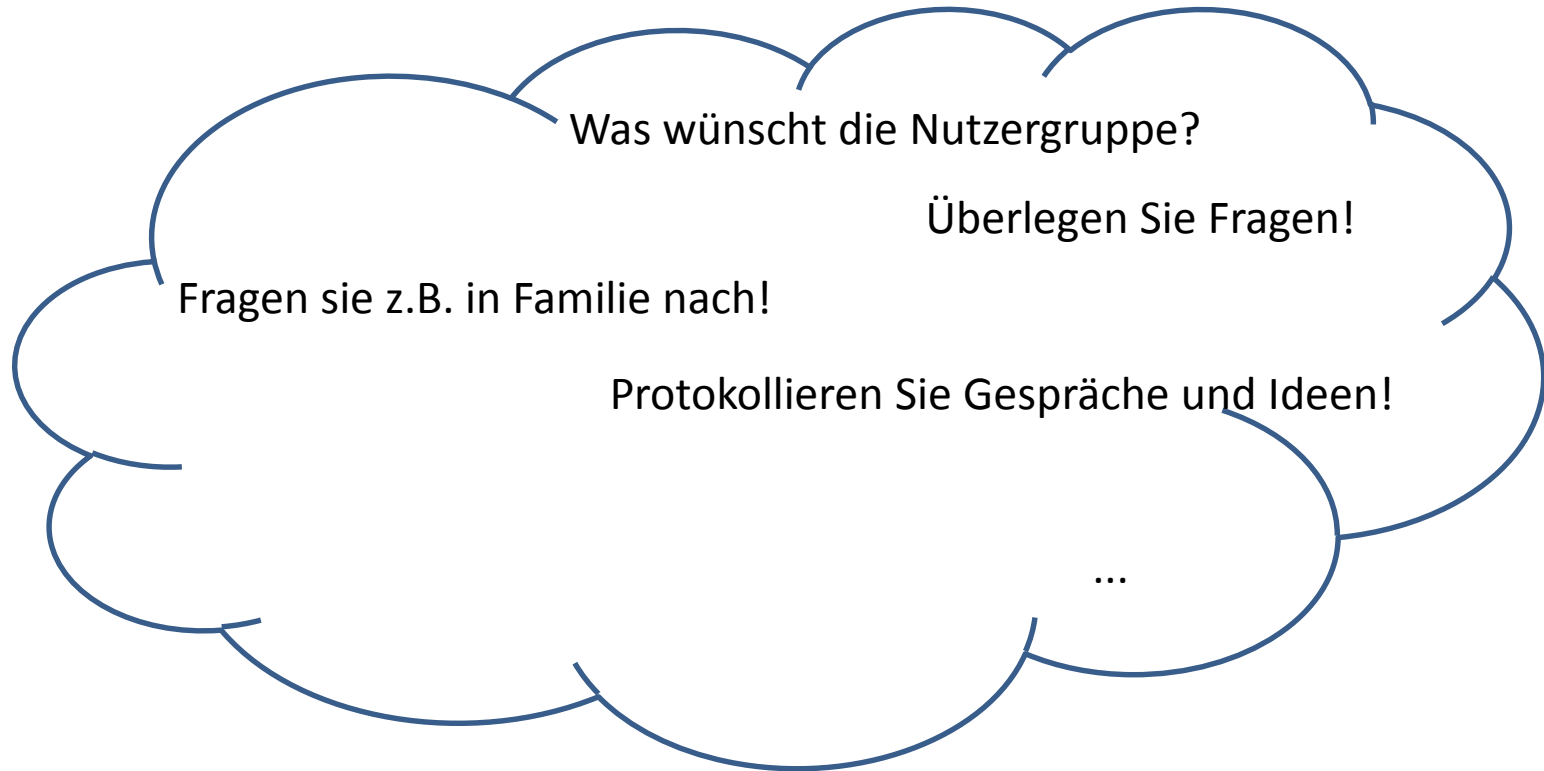
## Software zur Unterstützung älterer Menschen zur Verwaltung und Präsentation ihrer Erfahrungen



Leiten Sie Anforderungen ab!

# Gruppenarbeit

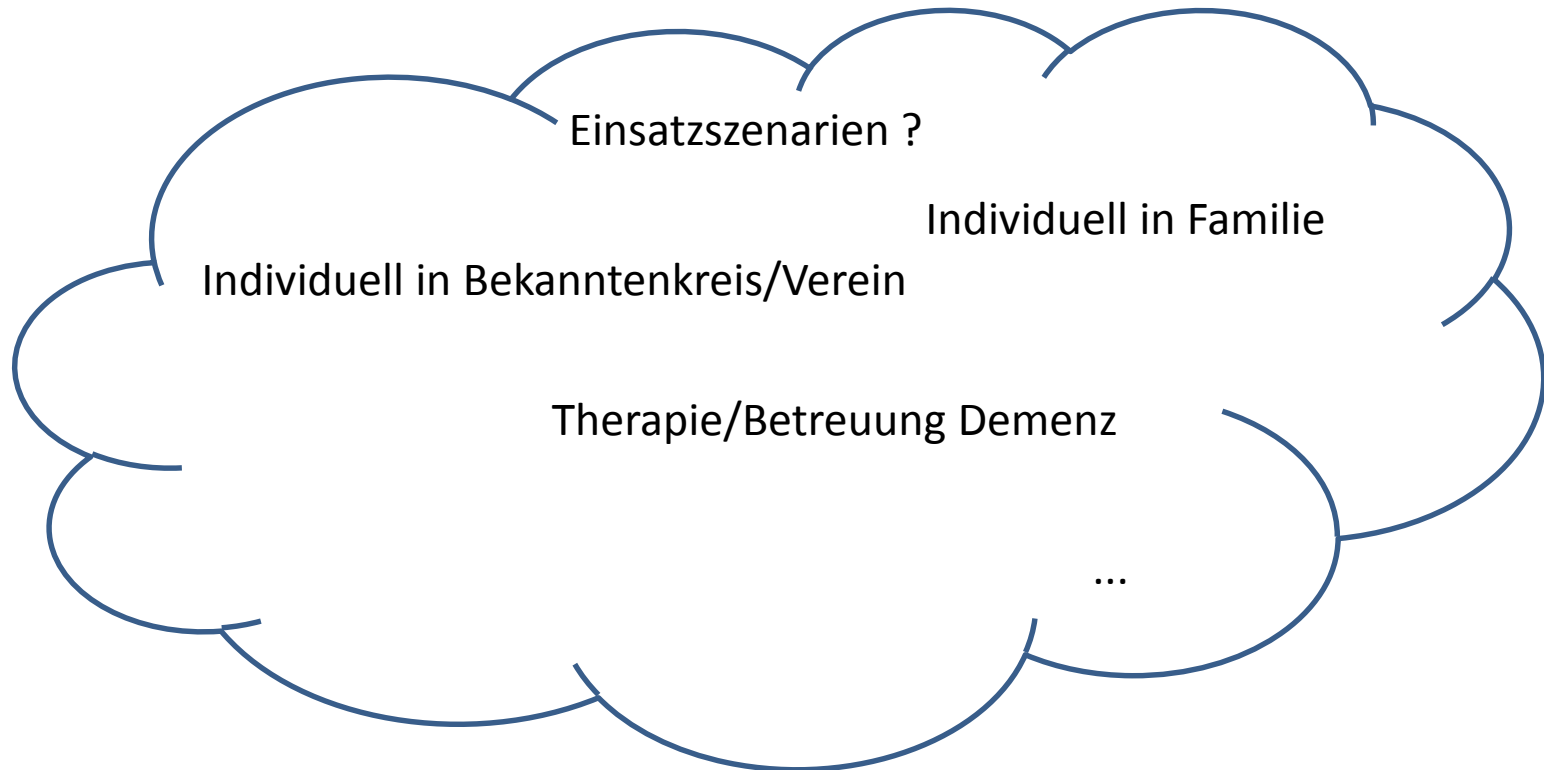
## Software zur Unterstützung älterer Menschen zur Verwaltung und Präsentation ihrer Erfahrungen



Leiten Sie Anforderungen ab!

# Gruppenarbeit

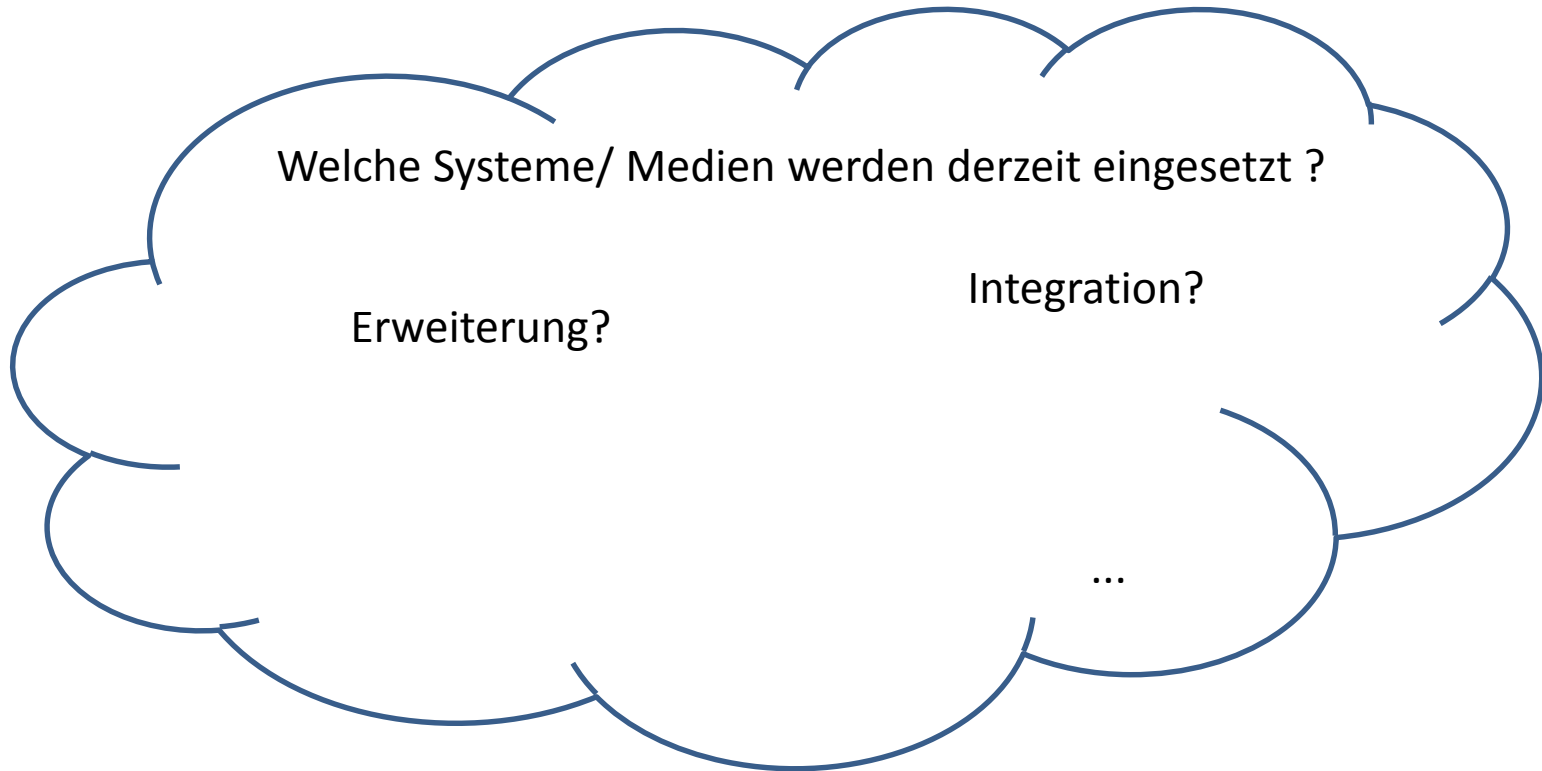
## Software zur Unterstützung älterer Menschen zur Verwaltung und Präsentation ihrer Erfahrungen



Leiten Sie Anforderungen ab!

# Gruppenarbeit

## Software zur Unterstützung älterer Menschen zur Verwaltung und Präsentation ihrer Erfahrungen



Leiten Sie Anforderungen ab!



# Gruppenarbeit

- Welche Nutzergruppen gibt es?
- Welche Zugriffsrechte müssen vergeben werden?
- Welche **Anwendungsfälle** können identifiziert werden?
- Welche Informationen werden wann bereitgestellt?
- Welchen Status hat ein Dokument zu einem Zeitpunkt?
- Können Benutzer gemeinsame Dokumente bearbeiten/teilen?
- Welche Art von Geräten werden unterstützt?

# Gruppenarbeit

## Rahmenbedingungen

- Es soll sich um eine in Java programmierte Client-Server-Anwendung handeln.
- Vor der Berücksichtigung der technischen Aspekte ist zu beachten:

Zunächst sind inhaltliche Überlegungen notwendig!!!

Wer? soll Wofür? Wie? unterstützt werden?

**Entwickeln Sie eine Vision!**

# Gruppenarbeit

## Rahmenbedingungen

- Für das Projektmanagement ist Redmine mit SVN ist zu nutzen.

(Dokumente, Protokolle, Diskussionen,...)

Ich bin in allen Projektgruppen Mitglied – beobachte  
Gruppenarbeiten bin diesmal nicht der direkte  
Auftraggeber, der Wünsche vorgibt.

Ich stelle meine Übungsmaterialien nur im Redmine bereit  
(nächste Woche sind die Projektgruppen in Redmine)

- Gruppen auf Redmine-Server des Lehrstuhls Softwaretechnik  
[proxy.informatik.uni-rostock.de/redmine/projects/swt20XX/](http://proxy.informatik.uni-rostock.de/redmine/projects/swt20XX/)  
Zugriff mit Itmz-Account
- Einrichtung der Gruppen in nächster Woche (Bis dahin möglichst stabile Gruppenzugehörigkeiten der Studierenden erreichen)

# Gruppenarbeit

## Meilensteine für die Gruppenarbeit

- Zeitplan (Abgabetermine Hausaufgaben):
  - Lastenheft 46. KW 2019 (11.11.-17.11.2019) 50 P
  - Papierprototyp 49. KW 2019 (02.12.-08.12.2019) 50 P
  - Pflichtenheft 51. KW 2019 (16.12.-22.12.2019) 50 P
  - Zwischenverteidigung 4. KW 2020 (29.01.2019) 100 Punkte
  - Darstellung des individuellen Beitrages 4./5. KW 2020 150 Punkte
- Ein Teil der Punkte gibt es für formale Anforderungen (Termine/vorgegebene Gliederungen/Umfang). Kurze Phasen zur Verbesserung mit Zielpunkten zur Verbesserung.
- Abgabe – Vorläufige Bewertung – Verbesserung – Endwertung
- Am Ende des Semesters hat jeder seinen Beitrag vor dem Übungsleiter und der Gruppe nachzuweisen!

# Gruppenarbeit (nur Informatik/Lehramt)

## Sommersemester – Erstellung des Prototypen

- Zeitplan (Abgabetermine):

- |  |               |            |
|--|---------------|------------|
| • Entwurfsdokument                             | 16. KW 2020   | 50 Punkte  |
| • Vorabgabe des Projektes                      | 21. KW 2020   | 50 Punkte  |
| (einschließlich Testbericht – eigenes Projekt) |               |            |
| • Abgabe Gesamtprojekt                         | 23. KW 2020   | 100 Punkte |
| • Bewertung von 2 Projekten                    | 25. KW 2020   | 50 Punkte  |
| (Projekte von anderen Gruppen)                 |               |            |
| • Endverteidigung                              | 27. KW 2020   | 100 Punkte |
| • Darstellung des individuellen Beitrages      | 27/28 KW 2020 | 150 Punkte |

# SWT Übung

## Heute

- Start Gruppenarbeit
  - Gruppenfindung
  - Vorbereitung Gruppentreff:
    - Arbeitsteilung festlegen
    - Ideenfindung für Aufgabenstellung (Vorbereitung Lastenheft)
- Fragen zur Vorlesung ?

# SWT Übung Themen heute

- **Arbeitsteilung bei der Softwareentwicklung**
- **Begriffe in der Softwaretechnik**
- **Anforderungsartefakte**
- **Anwendungsfälle**
- **Szenarium, Hauptszenarium, Alternativszenarium, Ausnahmeszenarium**
- **Prototyp der Benutzungsoberfläche (Hi-Fi, Lo-Fi)**

# Arbeitsteilung

## **Arbeitsteilung bei der Softwareentwicklung:**

- **Name für die Organisationsstruktur**
- **Idee der Arbeitsteilung, Rollen**
- **Vorteile des Verfahrens**
- **Nachteile des Verfahrens**



# Arbeitsteilung

## Namen für Organisationsstrukturen

- **Kontrollierte zentralisierte Organisationsstruktur**  
Chefprogrammierer-Projektassistent-Projektsekretär-Programmierer
- **Demokratisch dezentralisierte Organisationsstruktur**  
Mitarbeiter
- **Kontrollierte dezentralisierte Organisationsstruktur**  
Projektleiter – Teamleiter - Mitarbeiter

# Softwarelebenszyklus

**Softwarelebenszyklus:**

# Softwarelebenszyklus

**Softwarelebenszyklus:** ( kommt später in Vorlesung noch ausführlicher)

- Analysieren
- Spezifizieren
- Entwerfen
- Implementieren
- Testen
- Dokumentieren und Nutzen

# Begriffe der Softwaretechnik

- **Korrektheit**
  - **Flexibilität**
  - **Robustheit (Stabilität)**
  - **Verständlichkeit**
  - **Effizienz**
- 
- **Wiederverwendung**
  - **Reverse-Engineering**
  - **Re- Engineering**
  - **Prototyping**

# Begriffe der Softwaretechnik

- **Korrektheit**

Das Maß der Übereinstimmung zwischen Aufgabenstellung des Anwenders, Dokumentation und Implementation der Software

- **Flexibilität**

Das Maß der Anwendbarkeit von Software unter veränderten Bedingungen

- **Robustheit (Stabilität)**

Das Maß für sinnvolles Verhalten von Software in Ausnahme- und Fehlersituationen

- **Verständlichkeit**

Das Maß für die Handhabbarkeit (Lesbarkeit) von Software und ihrer Dokumente

- **Effizienz**

Das Maß von Rechenzeitaufwand und Speicherplatzbedarf

# Begriffe der Softwaretechnik

## Vorgehensweisen

- Wiederverwendung
- Forward-Engineering
- Reverse-Engineering
- Re- Engineering

# Begriffe der Softwaretechnik

## Vorgehensweisen

- **Wiederverwendung**  
Nutzung bereits erarbeiteter Arbeitsergebnisse
- **Forward-Engineering**  
Entwicklung von Software von der Spezifikation über Entwurf zur Implementierung (von Modellen zu Quelltext)
- **Reverse-Engineering**  
Formulierung der Eigenschaften eines bereits erarbeiteten Systems auf abstraktem Niveau (von Quelltext zu Modellen)
- **Re- Engineering**  
Durchführung des Reverse-Engineering und Neugestaltung des Systems auf der Basis der gewonnenen abstrakten Beschreibung

# Begriffe der Softwaretechnik

## Prototyping

- **Rapid Prototyping**
- **Exploratives Prototyping**
- **Partizipatives Prototyping**



# Begriffe der Softwaretechnik

## Rapid Prototyping

Schnelles Erstellen eines lauffähigen Systems, das wesentliche Eigenschaften des endgültigen Softwaresystems besitzt

## Exploratives Prototyping

Erkundendes Prototyping bei kritischen Teilproblemen (Meist technischer Art)

Erkunde Client-Server unter Java,

## Partizipatives Prototyping

Einbeziehung des späteren Nutzers in das Prototyping (besonders bei der Gestaltung der Benutzungsschnittstelle).

Papierprototypen

Lo-Fi low-fidelity versus Hi-Fi height fidelity

# Anforderungen

**Funktionale Anforderungen**

**Nicht-funktionale Anforderungen**

**Domäne-Anforderungen**

**Anforderungsartefakte**

- **Ziele**
- **Szenarien**
- **Strategien**

# Anforderungen

## **Funktionale Anforderungen**

Aussagen über die vom System zu liefernden Dienste, wie das System auf bestimmte Eingaben reagieren sollte und wie sich das System in bestimmten Situationen verhalten sollte. ( Manchmal kann auch explizit ausgedrückt werden, was das System nicht machen sollte.)

## **Nicht-funktionale Anforderungen**

Aussagen zu Einschränkungen an die vom System angebotenen Dienste oder Funktionen ( z.B. zeitliche Einschränkungen, Einschränkungen an den Entwicklungsprozess, Bedienbarkeit, Sicherheit, ...).

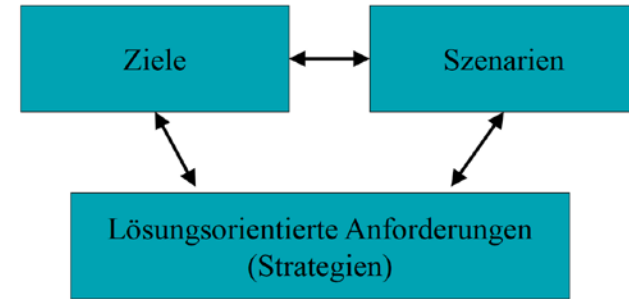
Diese Anforderungen werden oft an das System als Ganzes gestellt.

## **Domäne-Anforderungen**

Anforderungen, die aus dem Anwendungsbereich des Systems kommen und dessen Eigenschaften und Bedingungen reflektieren. ( z.B. Standards)

# Anforderungsartefakte

## Anforderungsartefakte:



- **Ziele** Definition: Ein Ziel ist die intentionale Beschreibung eines charakteristischen Merkmals des zu entwickelnden Systems bzw. des zugehörigen Entwicklungsprozesses.
- **Szenarien** Definition: Ein Szenarium beschreibt ein konkretes Beispiel für die Erfüllung bzw. Nichterfüllung eines oder mehrerer Ziele. Es konkretisiert dadurch eines oder mehrere Ziele. Ein Szenarium enthält typischerweise eine Folge von Interaktionsschritten und setzt diese in Bezug zum Systemkontext.
- **Strategien** Definition: Eine Strategie ist ein längerfristig ausgerichtetes planvolles Anstreben einer vorteilhaften Lage oder eines Ziels. Formal mathematisch ist eine Strategie eine Folge von Funktionen von einer Zustandsmenge ( z. B die Menge der denkbaren Spielsituationen eines Spielers) in eine Menge von Aktionen ( die entsprechend dem Spieler vorschreibt, was er tun soll).

# Szenarien

## Szenarien (Szenarientypen, Anwendungsfälle)

**Anwendungsfall** Definition: Ein Anwendungsfall beschreibt eine Menge von Aktivitäten eines Systems aus der Sicht seiner Akteure, die für diese zu wahrnehmbaren Ergebnissen führen. Er wird immer durch einen Akteur ausgelöst.

Ein Akteur ist eine außerhalb eines Systems liegende Rolle. Diese Rolle kann durch einen Anwender oder ein System ausgefüllt werden.

Es ist häufig sinnvoll, zu einem Anwendungsfall einen Prototypen der Benutzungsoberfläche zu entwickeln ( bzw. aus einem entwickelten Prototypen Anwendungsfälle abzuleiten)

**Anwendungsfallmodell** Definition: Ein Anwendungsfalldiagramm ist eine kompakte Repräsentation der Zusammenhänge von Anwendungsfällen. (z.B. Use-Case-Diagramm aus UML)

# Anwendungsfallbeschreibung

## **Anwendungsfallbeschreibung:**

Verkürzte Darstellung des Anwendungsbereiches in gegliederter Textform.  
Gliederung (in Vorlesung Softwaretechnik):

- Ziel
- Primärer Akteur
- Sekundärer Akteur
- Auslöser
- Interaktionsfolge
- Ausnahmen und Fehlerfälle
- Enthaltenen Anwendungsfälle

## **Hilfreich beim erstellen können folgende Elemente sein:**

**Prototypen** (Wie erfolgt die Bedienung?)

**Sequenzdiagramme** (kommen später bzw laut Vorlesung siehe Tafel)

**Use Case Diagramme** ( Zusammenfassungen und Überblick)

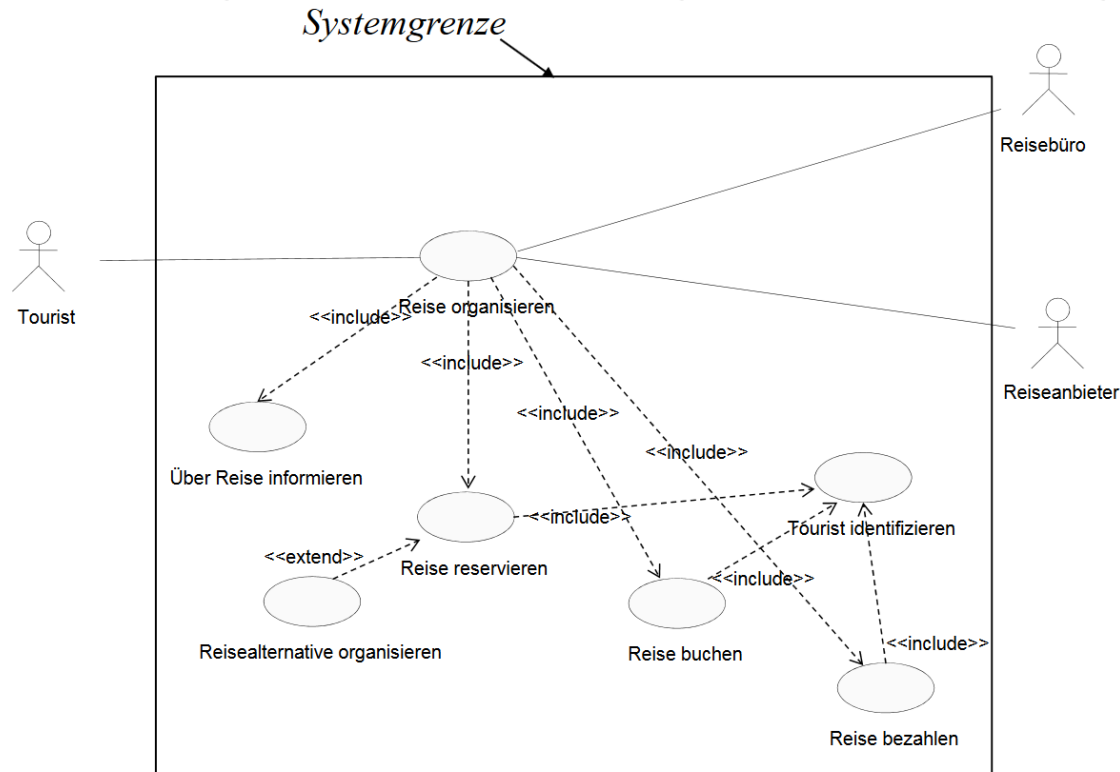
# Anwendungsfallbeschreibung

## **Anwendungsfallbeschreibung: „Reise organisieren“ aus Vorlesung wiederholen**

- Use Case Diagramm ?
- Sequenzdiagramm – Hauptszenario ?
- (Szenarium, Hauptszenarium, Alternativszenarium, Ausnahmeszenarium)
- Anwendungsfallbeschreibung ?

# Anwendungsfallbeschreibung

## Anwendungsfall-Use Case Diagramm: „Reise organisieren“- Vorlesung



Konsequenzen für Anwendungsfallbeschreibung „Reise organisieren“?



# Anwendungsfallbeschreibung

## **Anwendungsfallbeschreibung: „Aufladen von Mensakarten“**

Diskussion der Beschreibungen für den Anwendungsfall „Aufladen von Mensakarten“ mit dem momentan fest installierten Automaten und für eine Variante zur Bezahlung über ein mobiles Gerät.

# Aufgabe

Für einen Schrankproduzenten soll eine Software zur Verwaltung von Aufträgen erstellt werden, die die Kunden über einen Webshop auslösen können. Die Arbeitsweise der Firma ist wie folgt:

Ist ein Auftrag von einem Kunden eingegangen, dann stellt ein Disponent die notwendigen Materialien mit der Verwaltungssoftware zusammen. Im Normalfall sind diese im Lager vorrätig. In Ausnahmefällen muss der Disponent aber eine Bestellung auslösen. Ein Produktionsleiter kann sich Materialzusammenstellungen anzeigen und die Produktion der Schränke beginnen lassen. Die Produktion selbst wird nicht durch die Verwaltungssoftware unterstützt. Der Produktionsleiter bestätigt nur die Fertigstellung des Auftrages. Die Rechnungslegung wird allerdings wieder unterstützt. Vom Buchhalter wird eine Rechnung mit der Verwaltungssoftware erstellt. Der Kunde kann erstellte Rechnungen einsehen und unter Einbeziehung seiner Bank den Rechnungsbetrag überweisen.

**Modellieren Sie ein Anwendungsfalldiagramm, das die dargestellten Aktivitäten als Anwendungsfälle spezifiziert.**

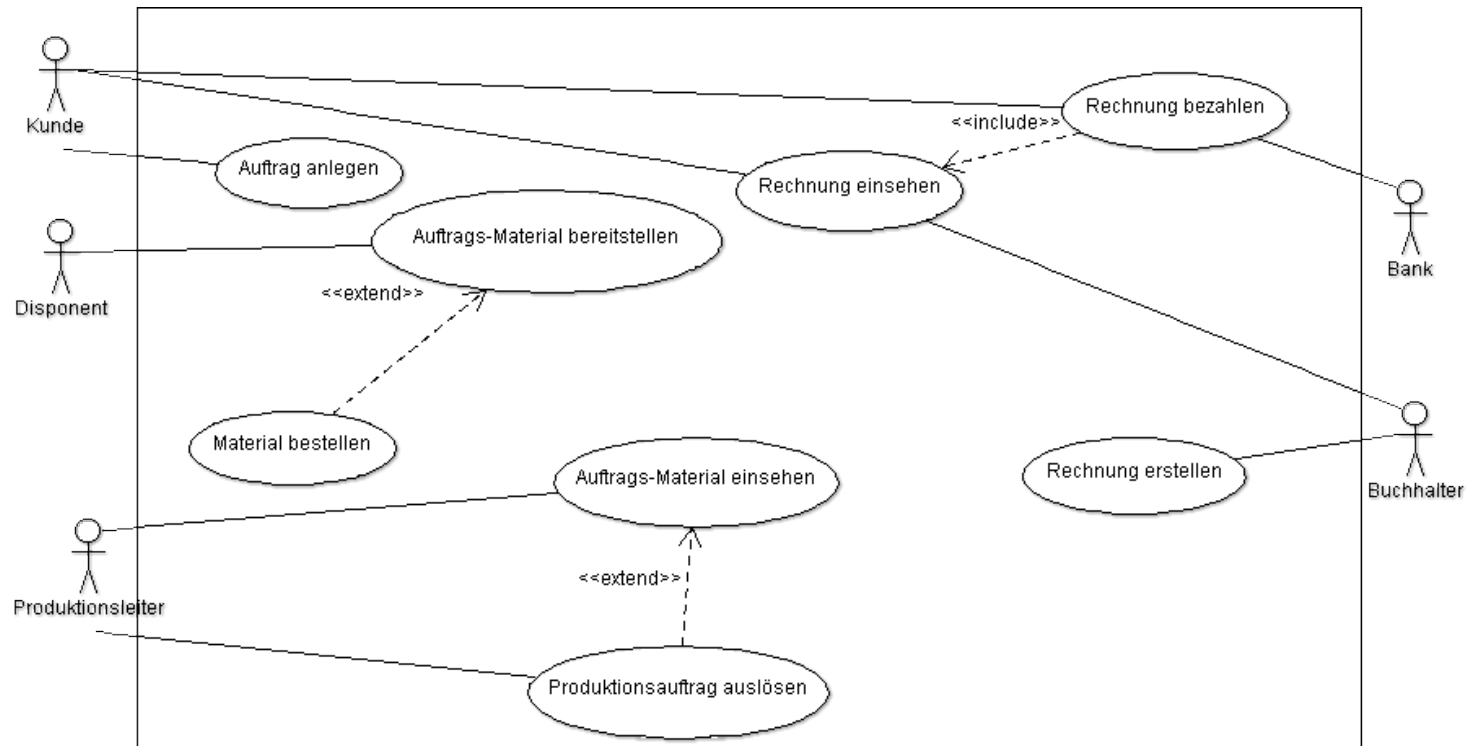
# Mögliche Lösung

Anwendungsfalldiagramm

Ideen für Anwendungsfälle:

- Auftrag anlegen (durch Kunden im Webshop)
- Auftrags-Materialien bereitstellen (durch Disponent) mit Sonderfall Material bestellen( wer ist Primärer Akteur ?)
- Auftrags-Material einsehen ( Produktionsleiter) mit Sonderfall Produktionsauftrag auslösen ( Ist das eigenständig ?)
- Rechnung erstellen ( Buchhalter)
- Rechnung einsehen ( Kunde ) mit include Rechnungsbetrag überweisen ( sekundärer Akteur Bank)
- Offen: Produktionsleiter bestätigt Fertigstellung des Auftrages
- (Eigenständiger Anwendungsfall oder Startereignis für Rechnung erstellen)

# Mögliches Anwendungsfalldiagramm



# Aufgabe

Geben Sie für den Anwendungsfall des Bezahlens einer Rechnung die in der Vorlesung behandelte strukturierte textuelle Form an. Dabei kann davon ausgegangen werden, dass der angemeldete Kunde nur den Betrag und die Rechnungsnummer eingeben muss. (Kontodaten vom Kunden und vom Schrankproduzenten seien in der Verwaltungssoftware hinterlegt). Die Bestätigung der Überweisung erfolgt durch die Eingabe einer Tan, die beim Kunden vorliegt. (Deren Übermittlung wird aus Vereinfachungsgründen nicht spezifiziert.)

Spezifizieren Sie mindestens vier Ausnahmefälle für den Anwendungsfall „Rechnung überweisen“.

**Anwendungsfallbeschreibung angeben!**

# Aufgabe Anwendungsfallbeschreibung

1. Ziel:
2. Primärer Akteur:
3. Sekundärer Akteur:
4. Auslöser:
5. Interaktionsfolge:
6. Ausnahmen und Fehlerfälle:
7. Enthaltene Anwendungsfälle:

# Aufgabe Anwendungsfallbeschreibung

1. **Ziel:** Eine korrekt gestellte Rechnung wird durch Einzug des Rechnungsbetrages bezahlt. Dabei werden die mit Kunden verbundenen Zahlungsart und die dort gespeicherten Daten genutzt.
2. **Primärer Akteur:** Kunde
3. **Sekundärer Akteur:** Bank, Schrankproduzent
4. **Auslöser:** Ein Kunde kennt den Rechnungsbetrag und will ihn überweisen.
5. **Interaktionsfolge:**
  1. Kunde lässt sich Rechnung anzeigen
  2. Kunde wählt Zahlungsart Einzugsermächtigung aus
  3. System zeigt den ausgefüllten Überweisungsträger
  4. Kunde erteilt Einzugsermächtigung
  5. System zieht Rechnungsbetrag ein
  6. Kunde verlässt System
6. **Ausnahmen und Fehlerfälle:**
  - 1a. Rechnung reklamieren
  - 2a. Falsche Zahlungsart gewählt
  - 3a. Kontodaten nicht aktuell
  - 4a. Onlinezugang gestört
7. **Enthaltene Anwendungsfälle:** Rechnung anzeigen