



# Software Engineering

---

Besprechung zur Uebung 3  
Softwareprozesse



# Formelles

---

- Gruppen
  - 3 Einzelgruppen noch übrig.
- Rechtschreibung, Grammatik, strukturierte Argumentation
  - Teilweise mühsam zu lesen.
  - Gab keine Abzüge.
  - Im Allgemeinen o.k.
- Allgemein
  - Lösungen durchgehend zumeist positiv.



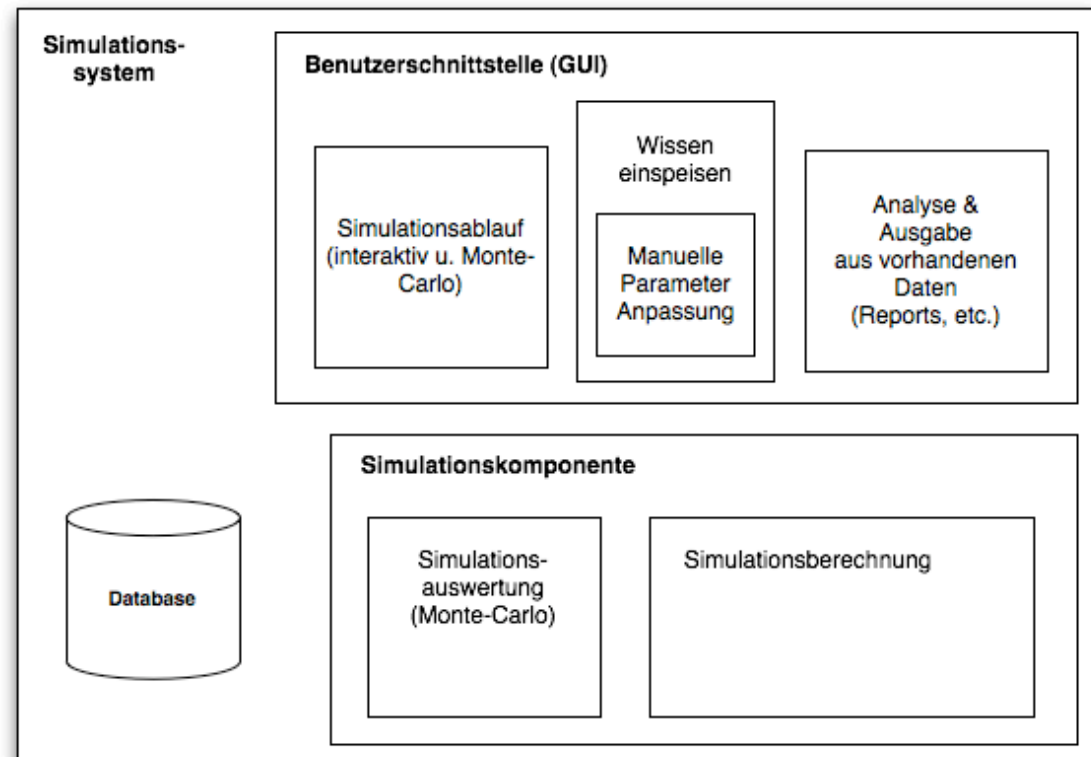
# Aufgabe 2.1: Analyse der Systemkomponenten (1)

---

- Identifikation der Systemteile / Komponenten
- Grobe Architektur des Systems
- Ziele
  - Zerlegung des Systems in Teilsysteme
    - Top-down Entwicklung
  - Vorbereitung für den Entwicklungsprozess
    - Organisation von Teillieferungen, Ressourcen, ...
- Wichtiges Kriterium (bei Zerlegung)
  - Starke **Kohäsion** und geringe **Kopplung** der Komponenten
  - Abgrenzungen dort wo sinnvoll, keine Erschwernisse produzieren

# Aufgabe 2.1: Analyse der Systemkomponenten (2)

- Komponenten und Sub-Komponenten (Hierarchie)
- Am Besten ein Modell.
  - Wichtig: Legende zur Notation
- Eine mögliche Lösung die nur die Komponenten zeigt:





# Aufgabe 2.2: Prozessmodelle (1)

---

- Ziel: Identifizieren eines passenden Prozessmodells
  - a. Eignung der verschiedenen Prozessmodelle (je 1.5 P)
  - b. Lieferplan für Wachstumsmodell (1.5 P)
  - c. Wahl eines geeigneten Prozessmodells (1 P)
  
- Grund
  - Komplexität in grossen Projekten
  - Qualität der Lösung hängt auch von der Qualität des Prozesses ab
  
- Meist gut gelöst



# Aufgabe 2.2:

## Prozessmodelle (2)

---

- Aufgabe a.
  - Ergebnisorientierte Phasenmodelle
    - **Vorteile:** verständlich, natürlicher Lebenslauf, planvoll, gut führbar, frühe Fehlererkennung
    - **Nachteile:** ignoriert SW-Evolution, lauffähiges System erst spät, abrupte Inbetriebnahme
    - **Eignung:** nicht zu grosse Projekte, exakte Anforderungen definierbar, Know-How vorhanden, Entwicklungsrisiko ist gering
  - Wachstumsmodelle
    - **Vorteile:** natürliches Verhalten (Evolution), gut zur Projektführung (Lieferungen), frühe lauffähige Teile (beruhigt den Auftraggeber früh), viel Rückmeldung vom Kunden, schrittweise Einführung
    - **Nachteile:** schrittweiser Aufbau kann SW-Strukturen zerstören, Insel-Bildung der Teillieferungen möglich
    - **Eignung:** grosses System mit langer Entwicklung, Basisprodukt kann schon früh verwendet werden, ...



# Aufgabe 2.2:

## Prozessmodelle (3)

---

- Aufgabe a. (continued)
  - Agile Software Entwicklung
    - **Vorteile:** kurze Rückkopplungszyklen, frühe Fehlererkennung, Qualität entsteht bereits an der Quelle, gut zur Projektführung
    - **Nachteile:** fähige/erfahrene Entwickler u SW-Architekt nötig, Kunden müssen direkt am Projekt mitarbeiten und entscheidungsbefugt sein
    - **Eignung:** Problem gut in kleine Teilaufgaben zerlegbar, gute/erfahrene Leute vorhanden, guter/erfahrener SW-Architekt vorhanden, Teamarbeit; *Kunde* ist direkt in Entwicklung beteiligt, ist jederzeit verfügbar, ist eingearbeitet und ist entscheidungsbefugt



# Aufgabe 2.2: Prozessmodelle (4)

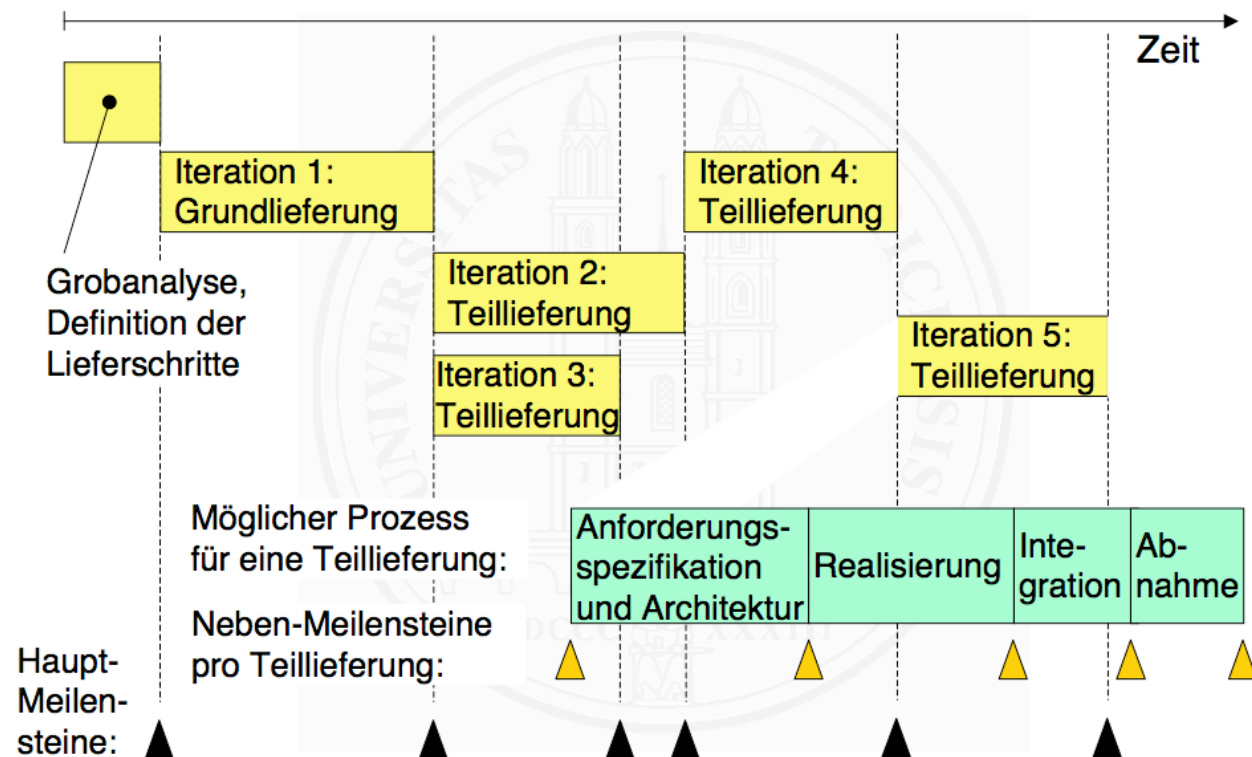
---

- Aufgabe a. (continued)
  - Charakterisierung des **Projektes**:
    - Evolutionäre Entwicklung (Basissystem das laufend erweitert wird)
    - Frühe Evaluierung, laufende Erweiterung der Anforderungen
    - mehrere Kunden vorhanden an verschiedenen Standorten
      - Kunden können wahrscheinlich nicht immer vor Ort in Entwicklung einbezogen werden
      - Nicht jeder Kunde kann einen befugten und fähigen Mitarbeiter für die Entwicklung bereitstellen
    - Entwickler Know-How ist gut (bezügl. Simulationen)
    - “Computer” Know-How der Kunden ist vermutlich nicht sehr hoch
    - etc.
- Aufgabe c. Wahl fällt auf Wachstumsmodelle
  - Ereignisorientiertes Phasenmodell ist zu starr (keine Evolution)
  - Agile Entwicklung hätte zu wenig Kundennähe



# Aufgabe 2.2: Prozessmodelle (5)

- Aufgabe 2.2.b. (und 2.3.)
  - Lieferplan eines **Wachstumsmodells**





# Aufgabe 2.3: Meilensteine (1)

---

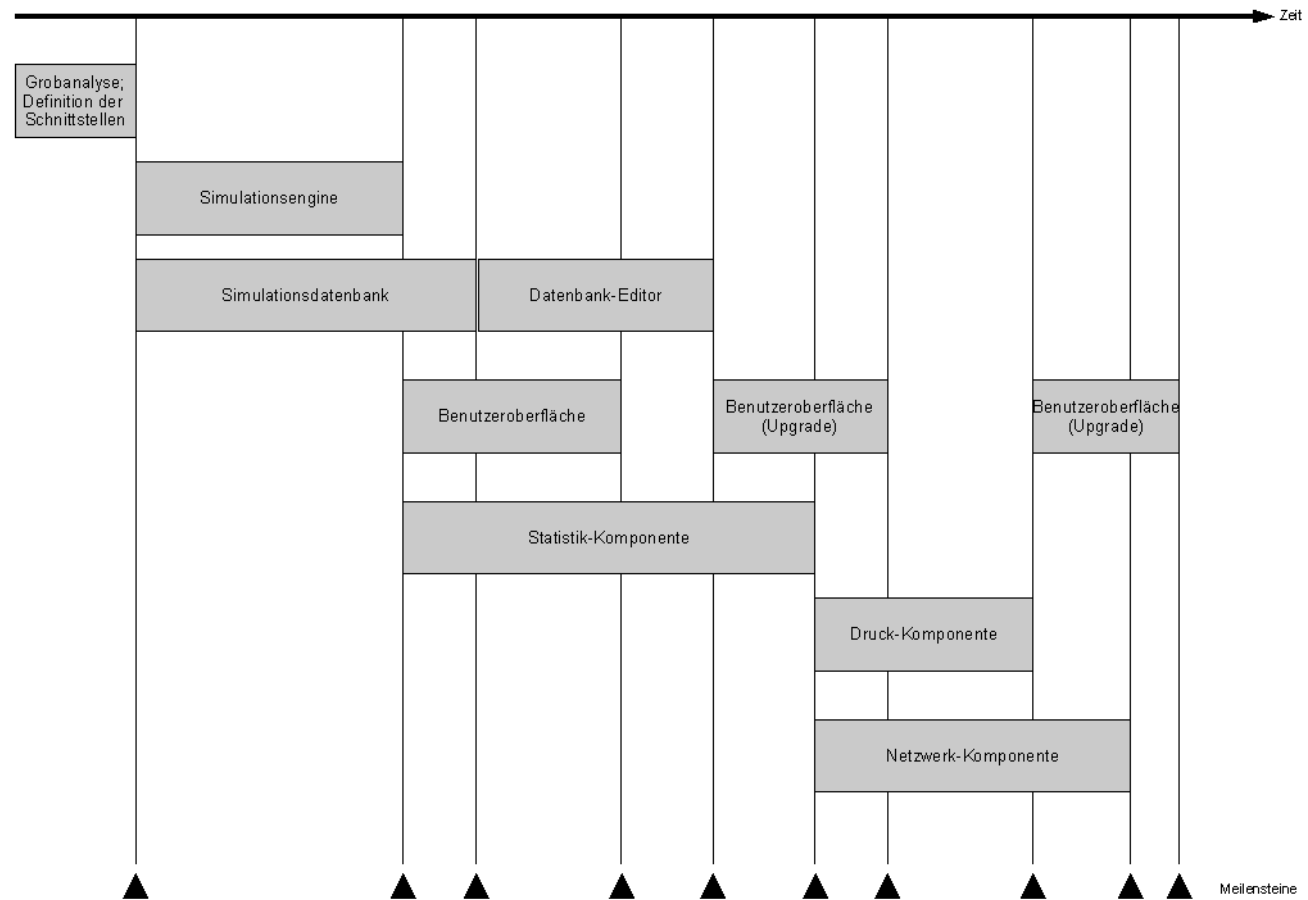
- Ziel
  - Strukturierung in
    - Lieferungen
    - und Ressourcen
  - Kontrollierbarer Fortschritt
    - Prüfbarkeit der Erreichung der definierten Meilensteine
      - Reviews, Abnahmetests, etc.
- Haupt-Meilensteine (Lieferungen bzw Iterationen)
- und Neben-Meilensteine (innerhalb einer Iteration)

# Aufgabe 2.3: Meilensteine (2)

- Möglicher Lieferplan

(aus einer Abgabe)

Haupt-Meilensteine:



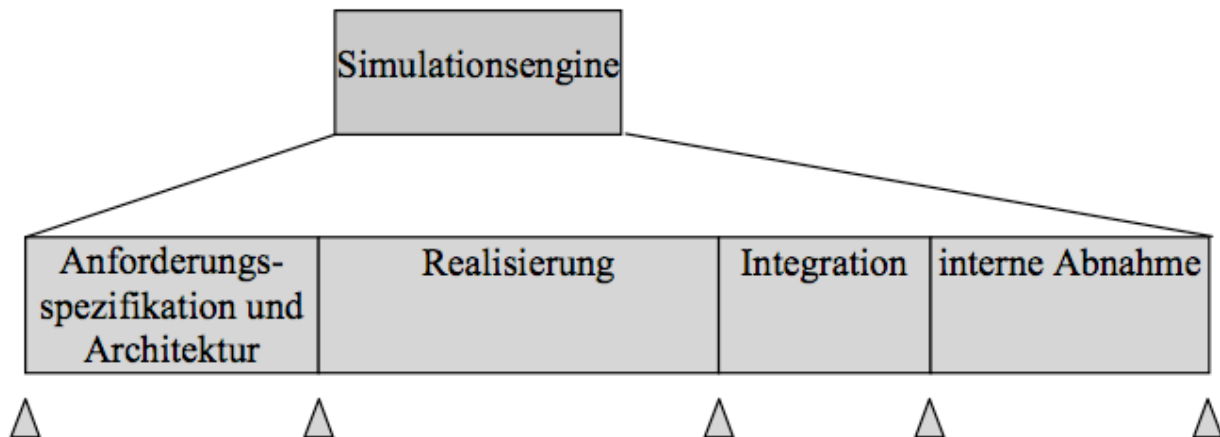
Viele Iterationen / Teillieferungen verbessern die Führbarkeit; ist positiv, wenn möglich.

# Aufgabe 2.3: Meilensteine (3)

- Möglicher Prozess für eine Iteration

(aus einer Abgabe)

Neben-Meilensteine:





# Aufgabe 2.4: Prototyping (1)

---

- Ziel: Frühe Erkennung und Lösung von Problemen
- Exploratives Prototyping
  - Klärung und Bestimmung von Anforderungen
  - *Demonstrationsprototyp*: Machbarkeit und Nützlichkeit von Systemideen
  - *Im engeren Sinn*: Angemessenheit von Anforderungen, Tauglichkeit von Lösungen
- Experimentelles Prototyping
  - *Labormuster*: Realisierbarkeit kritischer Teile, Bewertung von Alternativen
- --> *Wegwerf-Prototypen*
  - quick&dirty, undokumentiert, Risiko-Minderung
  - Wirtschaftlichkeit: wenn Ersparnis/Nutzen grösser ist als die Kosten.
  - Tatsächliches Wegwerfen nicht vergessen ...



# Aufgabe 2.4: Prototyping (2)

---

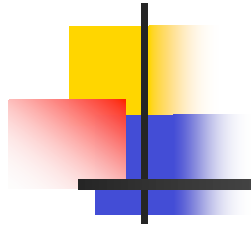
- Evolutionäres Prototyping
  - Prototyp entspricht einem Pilotsystem (wird nicht weggeworfen)
  - dokumentiert, sorgfältig entwickelt, ...
  - Entspricht dem Wachstumsmodell -- Basissystem das in weiteren Schritten zur Reife ausgebaut wird.



# Aufgabe 2.4: Prototyping (3)

---

- Im gegebenen Projekt sinnvoll:
  - GUI-Prototyp (Explorativ, Prototyp im engeren Sinn)
    - Zu Beginn, Anforderungsspezifikation und Validierung
    - Angemessenheit und Tauglichkeit der vorgeschlagenen Lösung ...
  - Simulationsengine (experimentell, Labormuster)
    - Realisierbarkeit technischer Lösungen (vgl. Source Code Uebung 4)
  - Pilotsystem (evolutionär, ab Basisversion ...)
    - Der Simulationsengine ...
    - Ausbau mit folgenden Iterationen im Wachstumsmodell



- Danke für die Aufmerksamkeit und  
schönes Wochenende!