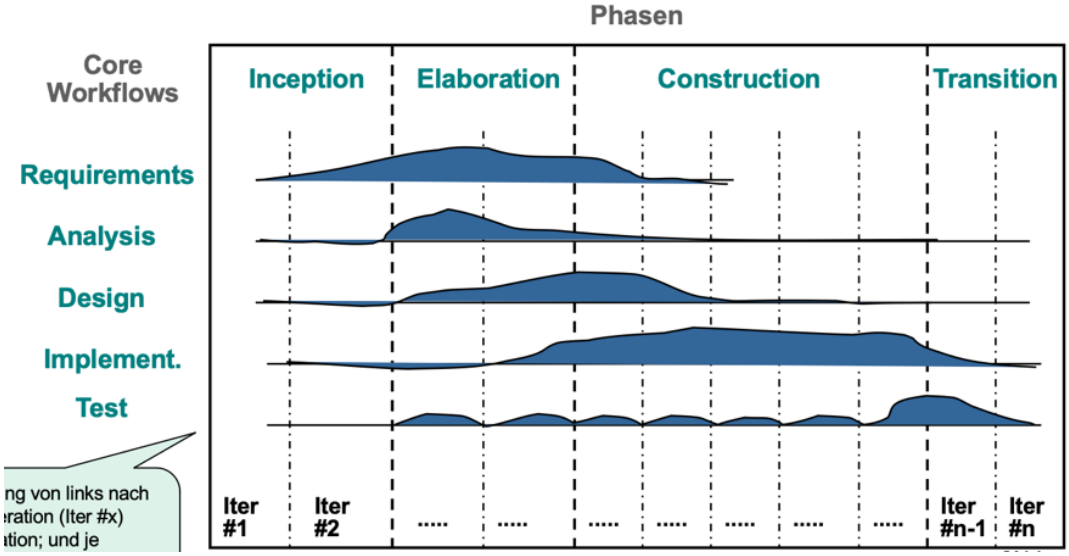
SE1 – SoSe18 – Dunkel

A1. Unified Process

a) Einzeichnen der Aufwände des Unified Process



b) Bennenen und Erläuterung der Projektphasen

- **Inception**: Projektorganisation, sammeln der kritischen Use Cases &
- Projektinfrastrukturvorbereitung und erste Gedanken zur Systemarchitektur
- **Elaboration**: sammeln aller Use Cases, Analyse und Implementierung der kritischen Use Cases & Entwurf und Implementierung des Kern der Architektur
- **Construction**: Alle Use Cases analysieren und inkrementell System weiterentwickeln der wichtigsten Use-Cases. Dabei pro Iteration Use-Cases neu priorisieren
- **Transition**: Abnahme des Systems. Abschließende Arbeiten und Deployment

d) Warum ist der UP iterativ & inkrementell

- **Iterativ**: es werden alle Arbeitsschritte ständig wiederholt
- **Inkrementell**: System wird stufenweise weiterentwickelt

e) Workflows bescheiben & Ergebnisse der Workflows beschreiben

- **Anforderungsanalyse**: Anforderungsmodell bestehend aus einer Systembeschreibung, einem Domänenmodell, der Anforderungen wie Use-Case Diagramm, Use-Case Beschreibung, nichtfunktionale Anforderungen und ggf. einen GUI-Prototyp
- **Analyse**: Analysemodell bestehend aus einem Klassendiagramm, Sequenzdiagramm, Paketdiagramm und ggf. einem Aktivitätsdiagramm & Zustandsdiagramm
- **Design**: Verteilungsdiagramm mit der logischen Struktur sowie Architektur- und Entwurfsmuster festlegen
- **Implementierung**: Integrationsstrategie festgelegt, Implementierungsmodell und das Build-Management
- **Test**: Testplan und Testfälle mit Test- und Fehlerbericht

f) Beschreiben der 3 Eigenschaften von UP

- **Use-Case Driven**: Da das System aus Sicht des Anwenders entwickelt wird und dessen Anforderungen im Vordergrund stehen. Die Anforderungen sind als Use-Cases beschrieben und diese steuern den Entwicklungsprozess. Bedeutet, dass die kritischen Use-Cases zuerst analysiert und implementiert werden
- **Architekturzentriert**: Technische Aspekte werden ebenfalls berücksichtigt und haben einen Einfluss auf die Architektur und der Priorisierung der Use Cases
- **Inkrementell**: System wird stufenweise weiterentwickelt

Weitere Aufgaben !!

a.Was beschreibt ein UC

i. Zusammenhang zweier Java-Objekte **||** ii. Beziehung zweier Akteure **||** **iii. Bez. Akteur und System**

b. Was beschreibt ein Domänenmodell?

i. Techn. B und K **||** **ii. Fachliche Begriffe und Konzepte** **||** iii. Grundlage für die Benutzerschnittstelle

c. Kann ein zu beschreibendes System ein Akteur sein

i. Ja **||** ii. Nein **||** iii. Kommt drauf an

d. Welche Anforderung ist funktional?

i. Erreichbar zwischen 6:00 – 20:00 **||** ii. Nur registrierte Kunden dürfen Bestellungen aufgeben **||**

e. Was bildet die Anforderungsanalyse ab

i. Externe Sicht **||** **ii. Interne fachliche Sicht** **||** iii. Technische Anforderungen

f. Was ist das Geschäftsprozessmodell?

i. Statische Systemsicht **||** **ii. Dynamische Systemsicht** **||** iii. Objektorientierte Systemsicht

g. Wie oft wird die Anforderungsanalyse ausgeführt?

i. 1 mal **||** ii. 2 mal **||** **iii. N-mal**

h. Wie ist der englische Fachbegriff für Anforderungsanalye?

i. Domain-Specification **||** ii. Conceptual Analysis **||** **iii. Requirement Analysis**

i. Wie wird ein Domänenmodell dargestellt

i. Sequenzdiagramm **||** ii. Zustandsdiagramm **||** **iii. Klassendiagramm**

j. Warum müssen nichtfunktionale Anforderungen quantifizierbar sein

i. Um zu prüfen, ob sie erfüllt sind.

2.) Analyse

a) Was sind funktionale und nichtfunktionale Anforderungen?

- Nichtfunktionale Anforderungen sind Qualitätsanforderungen, wie gut eine Funktion wird.
    - Antwortzeiten unter 3s
    - Verfügbarkeit zu 95% an Werktagen von 8 bis 20 Uhr
    - Last: über 500 Anfragen gleichzeitig annehmen können innerhalb 1s

b) Warum müssen nichtfunktionale Anforderungen quantifizierbar sein?

- Weil sonst keine Aussage getroffen werden kann, ob die Anforderung erfüllt ist.
- Zum Beispiel Antwortzeit soll schnell sein, was bedeutet das? Oder hohe Verfügbarkeit? Das kann vertraglich mit dem Kunden zu Problemen führen, da es verschiedene Ansichten geben kann.
- Wenn diese Anforderungen messbar sind, dann kann man sich darauf einigen im Vertrag und es kann deshalb nicht zu Streitigkeiten kommen

c) Warum kann das System kein Akteur sein?

- Akteure repräsentieren die Außenumgebung wie Personen oder Schnittstellen, die das System von außen nutzen und Funktionen erwarten. Deshalb kann das System selbst kein Akteur sein

d) Wie oft findet die Analyse im Unified Process statt?

- So häufig wie viele Iterationen festgelegt wurden

e) Was sind Entity-, Boundary- und Controlklassen? Wie würden diese in einem Sequenzdiagramm miteinander kommunizieren?

- **Entityklassen** repräsentieren Objekte in der realen Welt und halten datentragende Daten
- **Controlklassen** führen Geschäftsprozesse aus
- **Boundaryklassen** bieten die Schnittstelle nach außen für das System an

Kommunikation: der Akteur kommuniziert von außen z.B. über einer GUI mit den Boundary-Klassen. Diese kommunizieren mit den Control-Klassen und delegieren Aufrufe. Die Control-Klassemacht führt Geschäftsprozess aus und kommuniziert mit Entity-Klassen und mit anderen Control-Klassen

⇨ Akteur => Boundary => Control (=> Control) => Entity