**Pool II: Entwicklung von Softwaresytemen (Prozessmodelle)**

**Frage 1: RUP**

* Ausgangssituation: Softwareunternehmen; Sie sind Mitarbeiter und zuständig für die Spezifikation der Kundenanforderungen. Dabei arbeiten Sie eng mit dem Kunden zusammen und haben die Aufgabe, die Wünsche und Anforderungen des Kunden für die eigentliche Entwicklung exakt zu erfassen und zu dokumentieren. (Rolle des Systemanalytikers). Dazu müssen Sie Werkzeuge einsetzen, welche die Sicht des Kunden wiederspiegeln, um sicherzugehen, dass sie alle Geschäftsprozesse und Wünsche des Kunden richtig verstanden haben.

Eine mögliche Vorgehensweise dafür ist der RUP.

Fragen:

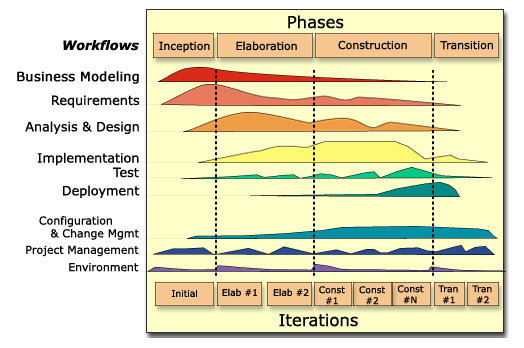
1. Beschreiben sie den RUP, seine Workflows und Phases
2. Erklären Sie die wichtigsten Artefakte des RUP pro Workflow und erstellen Sie beispielhaft die Artefakte für den Workflow „Requirements“ für die Beschreibung am Beiblatt. Beispiel: Bibliotheksverwaltung

(Domain Model und Use Case Diagramm) für die Anwendung

1. Diskutieren Sie Vor- und Nachteile des RUP

Antworten:

a)



**Horizontal axis (development phases and iterations):**

The RUP defines four Phases:

* Inception (Project setup, Concept)
* Elaboration (Draft, Architecture)
* Construction (Implementation)
* Transition (Deployment, Commissioning)

According to the iterative approach these phases are performed in iterations (Elab#1, Elab#2, Const#1, Const#2, …)

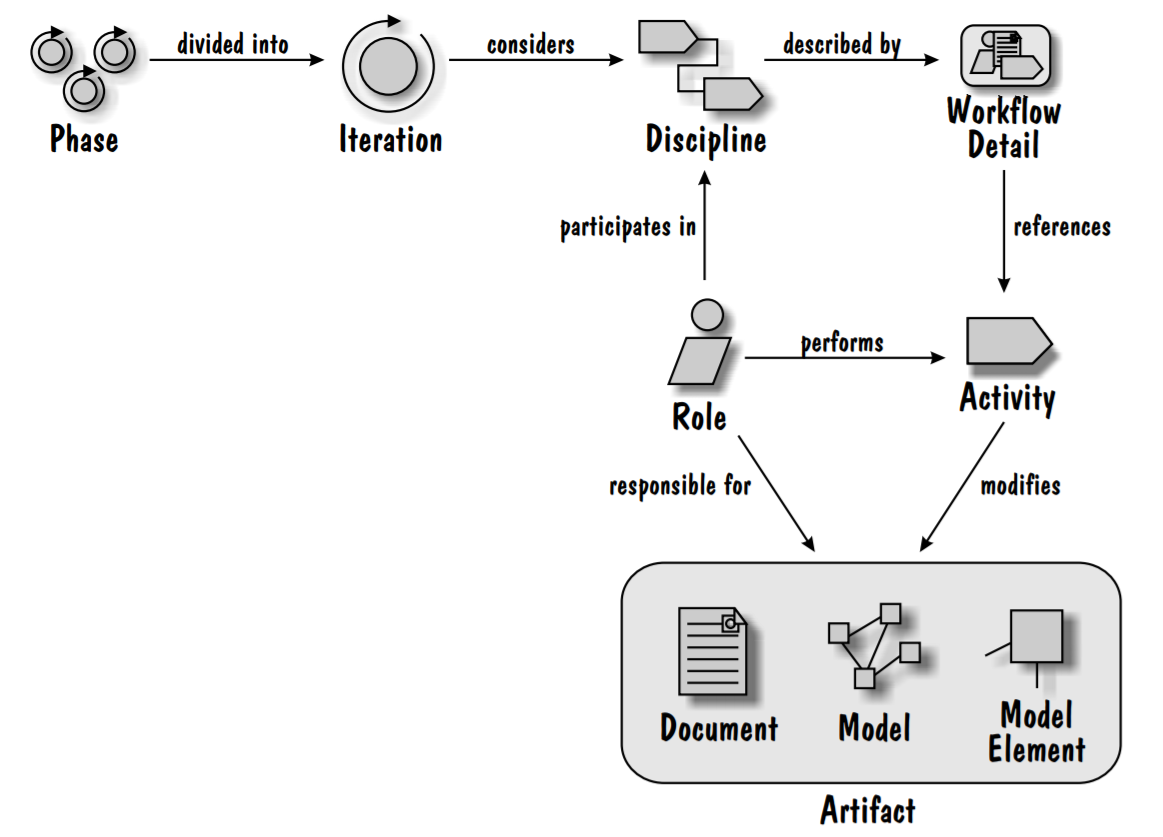
Each iteration results in a milestone. That is where it is decided if another iteration is required or if all set goals are achieved and the team can proceed to the next Iteration.  
  
**Vertical Axis (Disciplines / Workflows):**  
A project is divided into individual disciplines (activities). Each discipline is assigned a workflow, which is performed each iteration. A workflow represents a meaningful procedure for a successful execution of a discipline. Each step of a workflow should yield a part of an artefact.

**Colored Areas (Effort of disciplines per phase)**

The colored areas in the diagram represent the amount of work which goes into each discipline per phase.

An important fact is that disciplines are not restricted to a single phase. This means that unlike the “waterfall model” planning, development and testing does not take place strictly one after another. For example, using the RUP testing already happens during the Inception and Elaboration phases, and modelling / designing is still performed during the Construction phase.

Of course, certain disciplines / workflows are dominant during some phases but they are not performed exclusively in these phases. This is to ensure all requirements which are established near the end of the project can be considered. This difficulty has led to many projects (developed using the waterfall model) being abandoned.



(Abbildung auf Overhead-Folie drucken)!

b)

**Workflows des RUP & deren Artefakte**

Der RUP definiert 6 Core Workflows für die Erstellung des SW-Produktes und und für die 3 Suporting Workflows für die Prozessbegleitung, Die Core Workflows erwarten gewisse Artefakte als Eingabe und produzieren als Ausgabe wiederum eine festgelegte Anzahl von Artefakten.

**Core Workflows**

Decken die fachlichen und technischen Bereiche ab, (Erstelleung des Produktes)

### **Business Modelling**

Ziel dieses Workflows ist es ein Verständnis des Geschäftsprozesses sowie für das Umfeld in dem das System eingesetzt werden soll, zu bekommen. Außerdem sollen die derzeitig existierenden Probleme und deren Verbesserungspotential erkannt werden.

Die wichtigtsen Artefakte sind:

* **Business Vision**
* **Business Use Case Model**
* Business Use Case Diagram
* Business Use Cases ---- Description (Temaplate and ev Ads…Basic Flow, Alternative Flow, Extensions)
* Business Actors
* Business Workers
* **Domain Modell**
* Business Objects und ihre Beziehung untereinander

**Requirements**

Das Hauptziel ist es sich mit den Kunden sowie den Stakeholdern darauf zu einigen, welche Funktionen das Zielsystem haben soll. Dies stellt eine erste Basis für die Planung der technischen Umsetzung sowie für eine grobe Schätzung der Entwicklungsdauer des Projektes dar und gibt den Entwicklern einen Überblick über die Anforderungen.

Die wichtigtsen Artefakte sind:

* **Use Case Model**
* Use Case Diagram
* Use Cases --- Description (Template and ev ADs…Basic Flow, Alternative Flow, Extension)
* Actors (User)
* Workers
* **Refining des Domain Model (Class Diagram)!**
* **Vision**
* **Stakeholder Requests**
* **Prototype** **(Mock ups)**

**Analysis and Design**

Hierbei wird aus den Anforderungen das eigentliche System abgeleitet.

**Analyse Model**

Das Analyse Model ist Teil des RUP Workflows Analyse & Design. Aus den, im Requirements Workflow erstellten Use Case Modellen wird das Analyse Model abgeleitet. Es beschreibt die fachliche Logik des zu entwickelnden Systems aus statischer sowie dynamischer Sicht. Dies erfolgt mittels UML Diagrammen. Hauptziel ist es eine stabile Grundstruktur für das System zu schaffen.

Unterschiede zwischen Anforderungsanalyse (Requirements) und Analysemodel:

Das Use Case Model unterscheidet sich vom Analysis Model durch einige wesentliche Punkte:

* Analysis Model Diagram (AMD) statische Sicht
* Class Diagram, erweitert um die Analyseklassen (AMD oder ECB Pattern)
  + boundary classes (Schnittstellen)
  + control classes (Business Logic)
  + entity classes (Domains)
* Use Case Realization (fachliche Logik des Use Cases als Vorlage für die Programmlogik und Programmarchitektur)
  + Precondition
  + Sequence Diagram Basic Flow, alternate flows pro Use Case
  + Participated classes
  + Participated entities

|  |  |
| --- | --- |
| **Use Case Model** | **Analyse Model** |
| bedient sich der Sprache des Kunden | der des Programmierers |
| beschreibt das System aus einer externen Sicht | zeigt die interne Darstellung (Logik) |
| erhält seine Struktur durch die Anwendungsfälle | hingegen durch die stereotypen Klassen und Module |
| dient als Vertrag zwischen Kunde und Entwickler | wird von den Entwicklern verwendet |
| kann Redundanzen und Inkonsistenzen enthalten | darf keine Redundanzen und Inkonsistenzen enthalten |
| stellt die Funktionalität dar | stellt deren Umsetzung dar |

**Implementation**

In diesem Schritt findet die tatsächliche Implementierung des Systems statt. Bevor diese möglich ist, ist es notwendig den Implementierungsprozess zuerst ausreichend zu planen. (Verwendung v UML Diagrammen)

* Component Diagram
* Implementation Subsystem Diagram
* Integration Build Plan
* Build
* Implementation Model

**Test**

Dient dem Aufspüren von möglichen Fehlern im System sowie der Validierung ob das System den Anforderungen des Kunden entspricht.

* Test Plan
* Test Script
* Test Log
* Test Class/Data
* Test Component

#### Arten von Tests

* Unit-Test: Mittels Unit-Tests werden einzelne Komponenten der Software getestet. Dies erfolgt bereits im Implementation-Workflow.
* Systemtest: Es werden alle neu hinzugekommenen oder veränderten Teile der Software geprüft. Systemtest benötigen immer mehrere Durchläufe. Daher bietet es sich an diese zu automatisieren. In größeren Firmen existieren eigene Abteilungen um Systemtests durchzuführen.
* Integrationstest: Das gesamte System wird mit all seinen Komponenten geprüft.
* User-Acceptance-Test: Der Endnutzer testet das fertige System. Dafür kommt meist das Blackbox-Verfahren zum Einsatz. Im Unterschied zum Whitebox-Verfahren wird hierbei nur auf das Verhalten der Software anstatt auf den eigentlichen Programmcode geschaut.

### **Deployment**

Das entwickelte System wird an den Kunden ausgeliefert.

* Deployment Plan
* Release Notes
* Product
* Training Materials
* End User Support Material

**Angabe: Eine sehr vereinfachte Darstellung einer Bibliotheksverwaltung:**

* Die Bibliothek verleiht Bücher, DVDs, Blue Rays und CDs. Von jedem Ausleihgegenstand kann es mehrere Exemplare geben, diese werden durch Exemplarnummern voneinander unterschieden.
* Als Daten werden Titel, Autor, ISBN (falls Buch), Kurzbeschreibung und Einkaufspreis festgehalten.
* Vor der ersten **Ausleihung** wird der Kunde **registriert**, dabei werden seine Daten (Name, Adresse usw) erfasst.
* Normalerweise ist die Ausleihfrist für Bücher 30 Tage. Wird diese Frist überschritten, wird eine **Mahnung** verschickt.
* Ausleihungen können **verlängert** werden.
* Auch können **Reservierungen** vorgenommen werden.
* Zur Durchführung von **Bedarfsanalysen** werden die Informationen über die Ausleihvorgänge aufbewahrt.
* Ausleihgegenstände, die länger als 3 Jahre nicht angefordert wurden, werden **aus dem Bestand genommen** und karitativen Institutionen geschenkt oder bei Flohmärkten “verscherbelt“.
* Falls Ausleihgegenstände verloren gehen, werden sie unverzüglich bei den jeweiligen Verlagen **nachbestellt.**
* Die **Rückgabe** eines Ausleihgegenstandes wird vermerkt, zuvor wird allerdings dessen Unversehrtheit geprüft.

**Lösung**: selbst ausarbeiten!

Klassendiagramm: Ausleihgegenstand vererbt an Buch, CD, …

Exemplar wird an Kunden verliehen

Business Use Case Diagramm mit Actors (Flohmarkt, Kunde, Trigger, Lieferant/Verlag)

c)

**Vor- und Nachteile des RUP**

**Vorteile:**

* Risikofaktoren können schnell erkannt werden.
* Bessere Arbeitsteilung anhand verschiedener Rollen
* Man kann sich auf mögliche Probleme konzentrieren
* Laufende Qualitätssicherung
* Prozess verbessert sich nach jeder Iteration
* Supporting Workflows integriert (Configuration Management, Change Management, Project Management)

**Nachteile:**

* Komplexer Prozess
* Die Werkzeuge für RUP sind kostenpflichtig
* Oft komplizierte Iterationsplanung
* Hoher Dokumentationsaufwand
* Nur für große Projekte geeignet

Zur Erklärung – Iterationsplan:

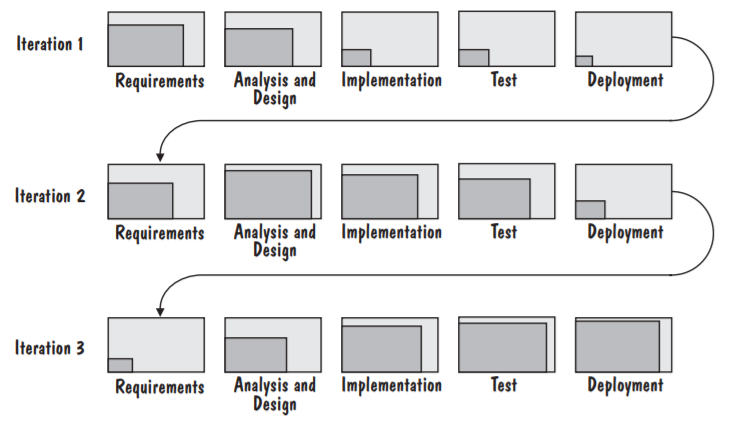
**Iterationen**

Iterationen sind ein Schlüsselmerkmal von erfolgreicher Softwarenentwicklung. Deshalb werden die beschriebenen Phasen üblicherweise in mehreren Iterationen durchlaufen.

Ziel einer Iteration ist es ein „Release“ zu erreichen. Dies kann ein interner Meilenstein sein aber auch ein neues Update für den Endnutzer. Um ein brauchbares Ergebnis zu erzielen ist es sinnvoll, pro Iteration alle Workflows zu Durchlaufen.

Entwickeln in Iterationen resultiert in „wachsender“ Software, d.h. das Produkt wird von Iteration zu Iteration ausgereiften. (besseres Verständnis der Anforderungen, robustere Architektur, erfahreneres Entwicklerteam, …)

Die folgende Grafik zeigt, wie im Laufe von Iterationen der Fokus auf verschiedene Workflows fällt. Es werden zwar alle Aktivitäten während jeder Iteration durchgeführt, aber die dafür aufgewandte Zeit verschiebt sich im Laufe der Iterationen. (Dargestellt durch die Grauen Boxen)



Jedem Release liegen diese Dokumente bei:

* Releasenotes (Beschreibung des Releases)
* Dokumentation für den Benutzer
* Changelog (Änderungen zum vorherigen Release)