

Übungsblatt 4

Aufgabe 4.1: Unified Process

Im Unified Process wird ein Software-System **inkrementell** in aufeinander aufbauenden Iterationen (Stufen bzw. Versionen) entwickelt. Bei einem neuen Software-Entwicklungsprojekt stellt sich nun die Frage, in wie viele Iterationen es unterteilt werden soll. Beschreiben Sie die **Vor-** und **Nachteile** von sehr **großen**, d.h. eher wenige, **Iterationen** bzw. sehr **kleinen**, d.h. eher viele, **Iterationen**.

große Iterationen

Vorteile:

- Beschäftigung hoch (Mitarbeiter haben zu tun)

Nachteile:

- Ergebnisse können nicht schnellstmöglich getestet werden
- zu viele Use Cases parallel

kleine Iterationen

Vorteile:

- Prototyp kann schnell erstellt werden

Nachteile:

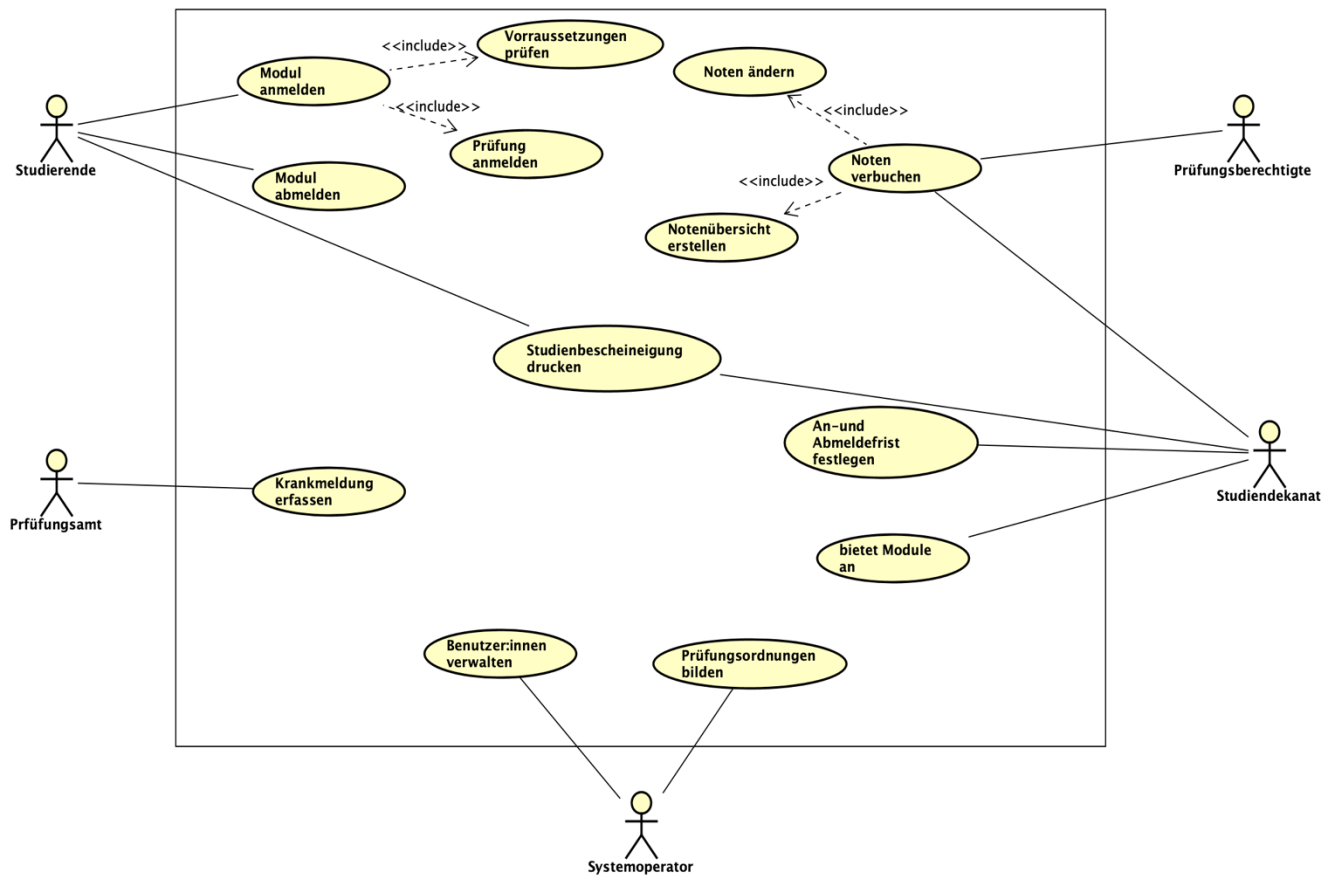
- Nicht alle Mitarbeiter sind beschäftigt

Aufgabe 4.2: Anforderungsanalyse – Pflichtaufgabe

Führen Sie für das im folgenden beschriebene System eine Anforderungsanalyse durch und fügen Sie alle erstellten Ergebnisse in einem Dokument als **Anforderungsmodell** (bzw. Pflichtenheft) zusammen.

- Erstellen Sie ein **Use Case Diagramm**. Finden Sie dazu alle Akteure, welche das System benutzen und bestimmen Sie alle Use Cases (Anwendungsfälle).
Strukturieren Sie das Use Case Diagramm ggf. mit Hilfe von `<<include>>`- und `<<extend>>`-Beziehungen!
- Beschreiben** Sie die zwei wichtigsten Use Cases.
- Erstellen Sie ein einfaches **Domänenmodell** !
- Bestimmen Sie beispielhaft die wichtigsten **nichtfunktionalen Anforderungen** für das System.

a)



b)

Use Case: Modul anmelden

Akteur: Studierende

Vorbedingung: Studierender erfüllt Voraussetzung für Teilnahme

Ereignisfluss:

1. Studierender klickt auf Modul anmelden
2. System speichert gewünschtes Modul
3. System prüft Anmeldefrist
4. Student erhält Textfeld
5. Anmeldung erfolgreich, Use Case beendet

Alternativen: zu 4: Anmeldefrist verpasst. Anmeldung nicht möglich

Nachbedingung: —

Use Case: Modul anlegen

Akteur: Studiendekanat

Vorbedingung: —

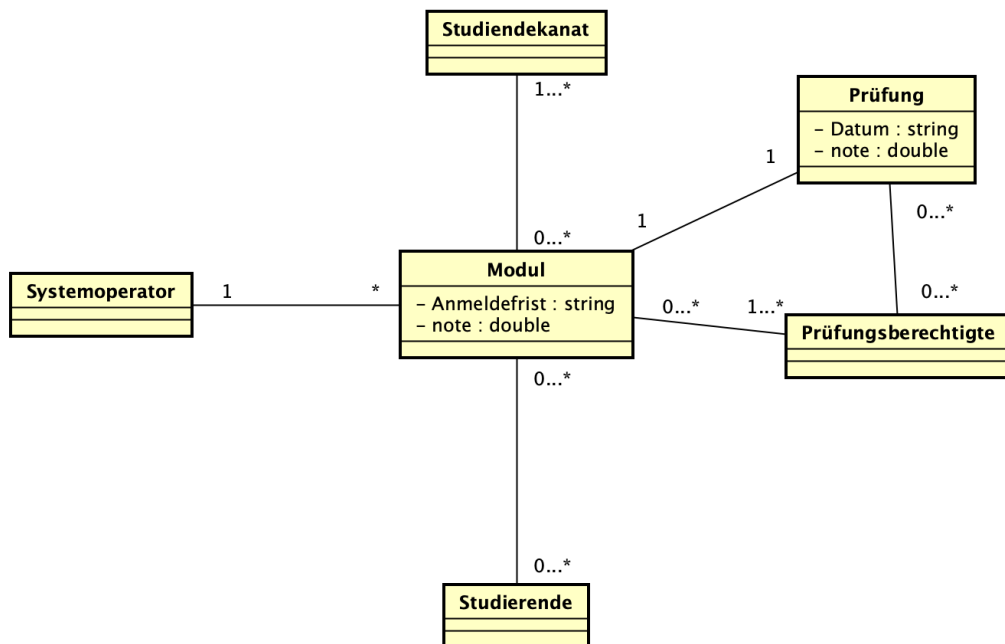
Ereignisfluss:

1. Modul anlegen klicken
2. Modulnamen eingeben
3. System speichert Modulnamen
4. Anmeldefristen festlegen
5. System speichert Fristen
6. Modul wird angelegt, Use Case beendet

Alternativen: —

Nachbedingung: —

c)



d)

wichtigsten nichtfunktionalen Anforderungen:

- Sicherheitsanforderungen
- Datenschutzanforderungen
- Zuverlässigkeit