

Dies ist keine „Musterlösung“, sondern eine gute von vielen möglichen Lösungen. Kommentare, die nicht Teil der Lösung sind, sind kursiv gesetzt.

Aufgabe 5-1:

a. Nicht-funktionale Anforderungen und Bereichsbedingungen

b. Sie sind aufwändig und machen es schwer, sinnvolle Verbesserungen zu realisieren. Denn beim Beobachten sieht man nur, **wie** es die Leute derzeit machen, nicht wie sie es vielleicht besser machen **sollten**.

c. Keine. Man wird in der Regel sogar mehrere gleichzeitig anwenden. In Aufgabe 5-2 z.B. Interview, Szenarien und Anwendungsfälle.

Aufgabe 5-2:

1. Anfangs gibt das Sekretariat, am Ende die Dozenten Veranstaltungen ein. Im Folgenden sollen nur die Dozenten dies machen können. Man könnte auch beide zulassen. Es gibt vermutlich einige andere Inkonsistenzen im Text, wenn man genau drüber liest. Diese ist am offensichtlichsten. Was macht man als Softwaretechniker mit Inkonsistenzen? Sie sind die ersten, die man aufklärt, d.h. man fragt beim Kunden nach.

2.

* funktional:

- Sekretariat gibt Prüfungsleistung ein
- Sekretariat gibt Veranstaltung ein
- Dozenten geben Veranstaltungen ein
- Studierende sehen Prüfungsleistungen ein
- Studierende buchen Veranstaltungen
- Studierende drucken Scheine aus.

* nicht-funktional (jeweils mit Messskalen):

- „Bedienung aber nicht viel anders sein als beim KVV“ (Messen der Einarbeitungszeit)
 - Besondere Zuverlässigkeit wegen „rechtlichen Folgen“ (Anzahl korrekt laufender Testfälle)
- „Dringend“ (Fertigstellungsdauer) ist eine Anforderung an den Prozess, nicht an das Produkt.

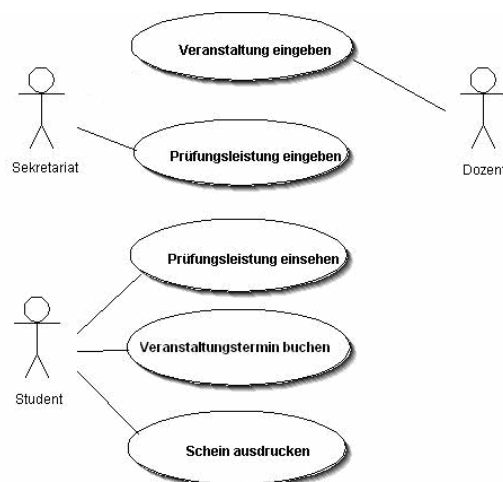
3. Die Studien- und Prüfungsordnungen der einzelnen Studiengänge.

4. Studierende, Sekretariat, Dozenten, Prüfungsamt, Systemadministrator, Studien/Prüfungsordnungsgeber, Geschäftsführung als Auftraggeber, evtl. auch die Entwickler.

5.a. Die genannten Akteure sind das Sekretariat, der Student und der Dozent. Man könnte auch den Drucker hinzunehmen. In der Regel sieht man von einem solchen Detail ab, aber man merke: Auch außen stehende, technische Geräte können Akteure sein, selbst solch passiven wie ein Drucker.

5.b. Siehe funktionale Anforderungen aus 2.

5.c.



Sekretariat: Macht Verwaltungsaufgaben für eine Arbeitsgruppe

Dozent: Legt Veranstaltungen und deren Termine fest und führt Veranstaltungen durch

Student: Nimmt an Veranstaltungen teil und bekommt Prüfungsleistungen gutgeschrieben

Veranstaltung: Eine einsemestriger Kurs zur Erlangung von Prüfungsleistungen

Schein: Beleg einer Prüfungsleistung

Prüfungsleistung: Voraussetzung zur Erlangung eines Studienabschlusses des Studenten

Achtung: Wenn man für das „Veranstaltung eingeben“ Dozent und Sekretariat zugelassen hätte, wird es etwas schwieriger. Denn man kann nicht einfach beide Akteure assoziieren, denn das würde heißen, dass immer beide beteiligt sind. Es bleibt dann nur, einen generalisierten Akteur zu definieren (z.B. Eingebener) und Sekretariat und Dozent als Unterakteur (ähnlich wie Unterklasse) zuzulassen. Dieses Vererbungsbeispiel zeigt übrigens die große Flexibilität von UML.

Der Drucker würde dann zum Fall „Schein ausdrucken“ assoziiert. Natürlich fehlen einige wichtige Anwendungsfälle. Die wurden in dem Text aber nicht explizit erwähnt. Nie etwas hinzudenken was nirgends steht! Wenn Zweifel aufkommen, immer den Kunden fragen.

6.a/b.

Voraussetzung: Student ist angemeldet und hat eine Veranstaltung ausgewählt

1. Student wählt einen Termin (wenn es mehrere gibt, z.B. bei Übungen) für die Veranstaltung.
2. System stellt fest, dass der Termin noch einen freien Platz hat.
3. System stellt fest, dass die Veranstaltung für den Studenten belegbar ist.
4. System bucht den Termin und gibt positive Rückmeldung.

Zusicherung: Voraussetzung + Termin ist für Student gebucht + Termin hat einen freien Platz weniger

Man könnte die Voraussetzungen auch zu dem Anwendungsfall hinzunehmen, also dass sich der Student erst anmeldet und dann eine Veranstaltung auswählt. Beide sind aber auch selbst wieder Anwendungsfälle, die man dann idealerweise im Anwendungsfalldiagramm mittels <<include>> zu diesem in Beziehung setzt. Die include- und extends-Beziehungen zwischen Anwendungsfällen sollte bekannt sein. Hier ein selbsterklärendes Beispiel:

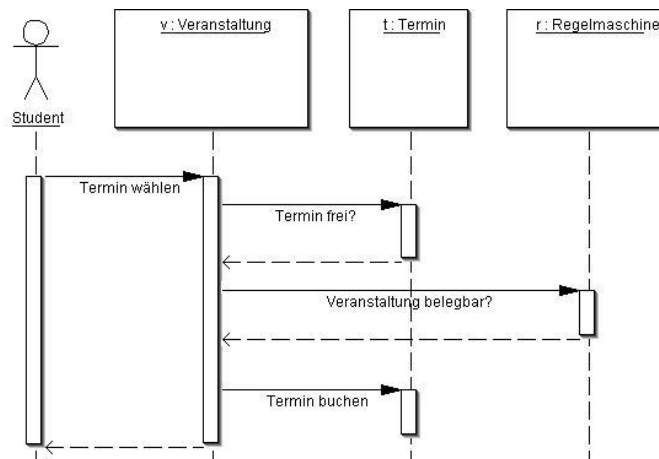
6.c.

- 2a. Der Termin ist schon komplett voll. Die Buchung wird abgelehnt.
- 3a. Die Veranstaltung passt nicht zum Studienverlauf des Studenten. Die Buchung wird abgelehnt.
- 3b. Der Student ist schon zweimal durchgefallen (nur Vordiplom). Es wird eine Studienberatung empfohlen.
- 3c. Wenn der Student für das Semester zu viele Leistungspunkte bucht, kommt eine Warnung.

6.d.

Sequenzdiagramm oder Aktivitätsdiagramm. Bei ersterem muss man sich aber die Klassen der Objekte überlegen. Was hier einfach das „System“ genannt wird, sollte weiter aufgeschlüsselt werden. Das ist aber schon Teil der Anforderungsanalyse. So gesehen sind Sequenzdiagramme ein gutes Mittel, systematisch von der Erhebung zur Analyse zu gelangen.

Das Sequenzdiagramm sieht z.B. so aus:



Die Beziehung der Objekte wird hier nicht geklärt (z.B. dass t ein Termin in v ist). Hier wurde eine Regelmaschine eingeführt, die die Studienordnungen kennt (und leicht änderbar ist, da sich Studienordnungen auch gelegentlich mal ändern). Achtung: Der Begriff ist schon sehr technisch und nimmt eigentlich unnötigerweise schon eine Entwurfsentscheidung voraus, die hier noch gar nicht gefragt ist.

Aufgabe 5-3:

Aufgabe ist es sich zu überlegen, welche Folgen diese Anforderungsänderung hat, wenn die Software schon fertig gestellt ist. Ideen: Die Software des Praxishelferrechners muss um diesen Fall erweitert werden, die Helfer brauchen entsprechende Autorisierungen, die softwareseitige Koordination bei gleichzeitigen Erfassungen muss erweitert werden (Entwurf), die Dokumentation und Schulungen müssen erweitert werden (und letztere evtl. verlängert), usw. (Weitere Ideen für Implikationen?)

Aufgabe 5-4:

- * VO-1/2: Passivphrasen; der Akteur ist im Kopf beschrieben, aber unklar bleibt die Funktion des weiteren Akteurs.
 - * VO-2: In 2. wird eine Erweiterung/Alternative im Hauptablauf beschrieben; das gehört da nicht hin
 - * VO-3: Passivphrase, besser wäre: „1. Der Verordnungsgeber signiert das Dokument (...)“
 - * VO-3: In der Alternative ist der „papiergebundene Prozessweg“ unklar referenziert.
- Insgesamt fehlt das typische Hin- und Her zwischen Akteuren und System komplett!