Furkan Aydin 1630039

Übung 2

Aufgabe 2.1: Grundlagen des Software Engineering
Bitte beantworten Sie die folgenden Fragen indem Sie in Software-Engineering-Büchern oder im Internet recherchieren.

(a) Welche Merkmale zeichnen **gute** Software aus?
Software Engineering soll zum einen natürlich dazu führen, Software-Entwicklungsprojekte erfolgreich durchzuführen, zum anderen soll aber die dabei entstehende Software **qualitativ hochwertig** sein.
Welches sind die wesentlichen Merkmale guter Software?

(b) Beschreiben Sie den Unterschied zwischen **Systemsoftware** und **Anwendungssoftware.** Geben Sie jeweils Beispiele an.

(c) **Softwareprodukte (= Standardsoftware)** decken einen klar definierten Anwendungsbereich ab (z.B. Buchhaltung, Lagerverwaltung) und können als vorgefertigtes Produkt gekauft oder gemietet werden. **Individualsoftware (Custom Software)** wird vollständig neu speziell für einen Kunden bzw. ein Unternehmen entwickelt (z.B. Zugdisposition der DB).
Geben Sie jeweils die Vorteile der beiden Arten von Software an.

(d) Stellen Sie die **Interessengruppen (Stakeholder)** eines von Ihnen durchgeführten oder aber auch fiktiven Softwareentwicklungsprojektes zwischen einem Auftraggeber (z.B. ein Verkehrsunternehmen) und einem Auftragnehmer (z.B. ein Softwarehaus) gegenüber. Welche **Interessenkonflikte** bestehen zwischen den Gruppen?

a)

- Gute Unterteilung des Programms in Funktionen um Redundanz zu vermeiden
- Code sollte gut dokumentiert sein
- Die Funktionen Sollten die Anforderungen erfüllen im vollen umtun
- Software sollte aufverschiedenen Systemen laufen
- Bedienungsfreundlich

b)
Systemsoftware ist die Software die den Betrieb eines Rechners steuert. Stellt verbindung zur Hardware her und steuert dessen Ressourcen.
(Betriebssysteme)

Anwendungssoftware werden vom Benutzer genutzt um nützliche
Oder gewünschte Funktionalitäten zu nutzen, um Aufgaben zu bewältigen.

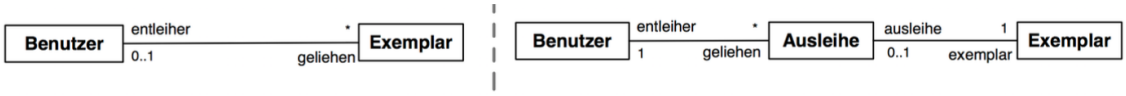
c)
Softwareproduke werden von vielen Käufern genutzt. So können Fehler und Bugs schnell gefixt werden, aber der Fokus auf Individualität geht verloren und steht immer im Konkurrenzdruck

Individualsoftware konzentriert sich nur auf den Kunden und soll in erster Linie nur bei ihm laufen

d)
Dem Auftraggeber ist in erster Linie die konkurrenz im Markt wichtig und möchte So viele Funktionen wie möglich abdecken ohne zu wissen was eigentlich möglich und nicht möglich ist. Der Entwickler hinweg weiß was implementierbar ist und muss dem Auftraggeber erklären was realisierbar ist. Fehlt jetzt beiderseits ein gewisses Vorwissen so kann schnell Ein Konflikt erstehen

Aufgabe 2.2: UML (Fortsetzung)

(a) Im Bibliotheksbeispiel aus der vorigen Übung lässt sich die Beziehung zwischen Benutzer und Exemplar auf zweierlei Weise modellieren.



Welcher der beiden Ansätze ist tragfähiger? Nennen Sie die Vor- und Nachteile der Ansätze!

Variante A:

- Vorteil:
  - Klassenmodell einfacher
  - Beziehung als Attribut

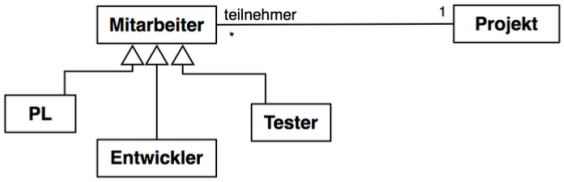
- Nachteil
  - Details zur Ausleihe fehlen z.B Ausleihdatum, Rückgabedatum usw. Passen nicht zue der Klasse => verletz Kohäsion
  - Beziehungen implementierung aufwädinger

Variante B:

- Vorteil
  - Verantwortungen klarer
  - Attribute an Ausleihe möglich für Details z.B Ausleihzeitpunkt, Rückgabedatum usw.

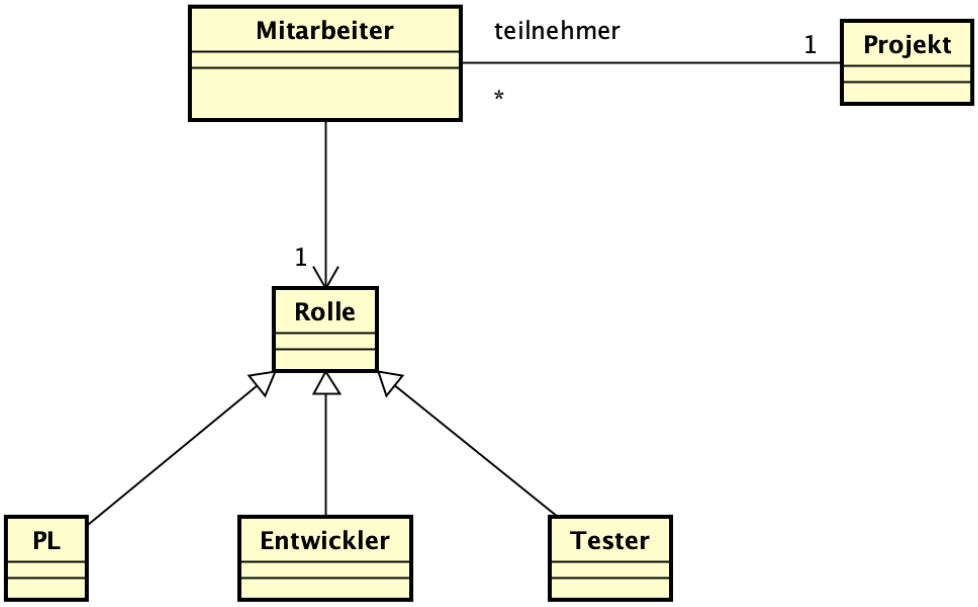
- Nachteil
  - Komplizierter: Mehr Klassen und Konsistenzüberwachnung schwieriger

(b) In einem UML-Klassendiagramm soll ein Projekt mit seinen Beteiligten modelliert werden. Dazu wird zunächst das folgende Modell entwickelt, das insbesondere die verschiedenen Rollen im Projekt abbildet.



Nun soll es aber möglich sein, dass die Rollen im Projekt dynamisch wechseln können. Zum Beispiel kann dieselbe Person während des Projektes zeitweise als Entwickler*in aber auch als Tester*in eingesetzt werden. Wie lässt sich das sinnvoll im Modell abbilden?

- Kann nicht zur Laufzeit Objekt ändern, deswegen geht die Abbildung oben nicht. Man kann natürlich ein neues Objekt erstellen, aber weniger schön



(c) Eine Assoziation dient dazu, Objekte zweier Klassen in Beziehung zu setzen. Welche Informationen werden durch nachfolgende Assoziationen ausgedrückt?

1. Polygon und Punkt kennen sich gegenseitig
2. Polygon besteht aus unbekannt vielen Punkten.
3. Ein Polygon besteht aus beliebig vielen Punkten und ein Punkt zu genau einem Polygon.
4. Ein Polygon besteht aus beliebig vielen Punkten und kennen sich gegenseitig. Zusätzlich besitzt Polygon ein Attribut von der Klasse Punkt
5. Polygon kennt beliebig viele Punkte und besitzt ein Attribut von der Klasse Punkt
6. Das gleiche wie 5, Kreuz dient nur der Verdeutlichung, dass in die andere Richtung die Navigation nicht möglich ist
7. Polygon besteht als Komposition aus beliebig vielen Punkten. Bedeutet, wenn Polygon gelöscht wird, müssen die Punkte auch gelöscht werden
8. Polygon besteht als Aggregation aus beliebig vielen Punkten. Falls ein Polygon gelöscht wird, dann besteht der Punkt weiterhin