

Übungen: Anforderungsanalyse und Spezifikation

AUFGABEN MIT LÖSUNG

Aufgabe 1 (UML-Anwendungsfalldiagramm)

Anwendungsfälle zur Elektronischen Gesundheitskarte (s. Dokument *b4h_usecase_l_v1-1.pdf* im Upload-Bereich „Erarbeitung einer Strategie zur Einführung der Gesundheitskarte“, insb. Seite 62):

Als Entwickler bist du an dem Projekt *Elektronische Gesundheitskarte* beteiligt. Der Projektleiter kommt zu dir, wirft die Dokumente der Anforderungsspezifikation auf den Schreibtisch und sagt: „Wir haben da etwas Wichtiges vergessen. Wenn ein Patient seine Einwilligung zu einer freiwilligen Anwendung geben muss, was ist, wenn der Patient ein Kind ist? Ergänze bitte in den betroffenen Anwendungsfällen die entsprechenden Diagramme um das Einholen einer Elterneinwilligung! Ach ja, und eine Patientenquittung soll entweder per Drucker ausgedruckt oder per E-Mail dem Patienten zugesendet werden. Kannst du das nicht irgendwie geschickt in das Diagramm einbauen?“

Aufgabe 2 (UML-Aktivitätsdiagramm)

Entwerfe ein UML-Aktivitätsdiagramm für den Ablauf einer Klausur als Prüfungsleistung an einer Hochschule und an einer Berufsakademie. Der Ablauf umfasst folgende Aspekte:

- Jeder Studierende sollte sich vor der Klausur sorgfältig durch Lernen der Lehrinhalte vorbereiten.
- An einer Hochschule muss sich ein Studierender parallel zum Vorbereiten beim Prüfungsamt zur Klausur anmelden. An einer Berufsakademie ist eine Anmeldung nicht erforderlich, da diese implizit erfolgt.
- Wenn der Studierende gesund ist, tritt er seinen Klausurversuch an. Wenn der Studierende erkrankt ist, sendet er dem Prüfungsamt ein Krankheitsattest und versucht anschließend zu genesen.
- Ein Krankheitsattest wird parallel vom Prüfungsamt geprüft, d.h. der Studierende wartet nicht auf eine Rückmeldung des Prüfungsamts, sondern beginnt umgehend mit der Genesung.
- Das Attest ist als Datenstruktur bekannt und dient als Output der Aktion *Attest senden* und als Input der Aktion *Attest prüfen*.
- Nach der Genesung beginnt der Studierende wieder mit der Klausurvorbereitung. Auch die Anmeldung zur Klausur ist an einer Hochschule erneut erforderlich.
- Nach einer angetretenen Klausur kann ein Studierender online seine Note über einen Dienst des Prüfungsamts abfragen.
- Wenn die Klausur erfolgreich bestanden wurde (Note < 5), ist der Ablauf der Prüfungsleistung erfolgreich beendet. Der Studierende feiert.
- Wenn die Klausur nicht erfolgreich bestanden wurde (Note = 5), beginnt der Studierende wieder mit der Vorbereitung (inkl. erneuter Anmeldung an einer Hochschule) für seinen nächsten Versuch.
- Wird eine Klausur zum 3. Mal nicht bestanden, wird der Ablauf der Prüfungsleistung mit einem Ausnahmefall beendet. Der Studierende wird durch das Prüfungsamt exmatrikuliert.

Aufgabe 3 (UML-Anwendungsfalldiagramm)

Entwerfe ein UML-Anwendungsfalldiagramm (*Use Case Diagram*) für die folgenden Funktionen eines Geldautomaten:

- Ein Kunde kann Geld abheben, seine aktuellen Kontoumsätze einsehen und die Sprache wechseln.
- Wenn ein Kunde Geld abheben möchte, sind folgende Teilschritte des Automaten erforderlich: Karte prüfen, eingegebene PIN prüfen, Geldbetrag erfragen und ggf. Geld auszahlen.
- Wenn die PIN nicht richtig ist, das Konto nicht ausreichend gedeckt oder das Tageslimit überschritten ist, wird ein Fehler ausgegeben. Das Ausgeben eines Fehlers erweitert die zugehörigen obigen Teilschritte.
- Wenn ein Kunde seine aktuellen Kontoumsätze einsehen möchte, sind folgende Teilschritte erforderlich: Karte prüfen, eingegebene PIN prüfen und Kontoumsätze anzeigen.
- Der Geldautomat kann Geld für Kunden aus der Schweiz auch in Franken auszahlen. Ein Schweizer Kunde ist also ein spezieller Kunde, der einen spezifischen Anwendungsfall nutzen kann: Geld in Franken abheben, was sich außer in der ausgezahlten Währung nicht von der normalen Geldabhebung unterscheidet.

Aufgabe 4 (UML-Aktivitätsdiagramm)

Entwerfe ein UML-Aktivitätsdiagramm für die Abwicklung eines Snowboard-Wettkampfes. Dabei sind folgende Aspekte zu berücksichtigen:

- Ein Snowboarder meldet sich zu einem Wettkampf an und wartet auf Rückmeldung des Veranstalters.
- Der Veranstalter (Sponsor) prüft, ob der Snowboarder sich im Voraus zur Teilnahme qualifiziert hat und gibt dem Snowboarder eine positive oder negative Rückmeldung zu dessen Anmeldung.
- Bei einer negativen Rückmeldung, versucht sich der Snowboarder bei einem anderen Wettkampf anzumelden. Die Aktivität beginnt von vorne.
- Bei einer positiven Rückmeldung beginnt der Snowboarder mit dem spezifischen Training für den Wettkampf und überweist parallel die Teilnahmegebühr. Anschließend nimmt er am Wettkampf teil und wartet nach dem Rennen auf die Auswertung des Veranstalters.
- Der Veranstalter vergibt nach dem Wettkampf entsprechend der Leistung des Snowboarders im Vergleich zu den anderen Teilnehmern eine bestimmte Punktzahl.
- Wenn die Punktzahl > 100 ist, lädt der Veranstalter den Snowboarder für die kommende Saison qualifiziert ein und prüft parallel sein Sponsoring-Budget. Wenn das Sponsoring-Budget noch nicht ausgeschöpft ist, sagt er dem Snowboarder eine Förderung über einen angemessenen Geldbetrag zu. Die Rückmeldung an den Snowboarder wird im Ganzen übermittelt.
- Der Snowboarder reagiert je nach Rückmeldung des Veranstalters unterschiedlich:
 - er gibt enttäuscht auf, wenn er keine Einladung erhält, so dass die Aktivität endet,
 - oder er feiert und wiederholt schnell die Anmeldung zum nächsten Wettkampf, wenn er eine Einladung und evtl. sogar eine Sponsoring-Zusage erhält. Bei einer Sponsoring-Zusage wird der Snowboarder vor der nächsten Anmeldung zum Pro.

Verwenden Sie hierbei als Notationselemente zwei Aktivitätsbereiche *Snowboarder* und *Sponsor* sowie als weitere Notationselemente *Signalsendung* und *-empfang*.

Aufgabe 5 (UML-Aktivitätsdiagramm)

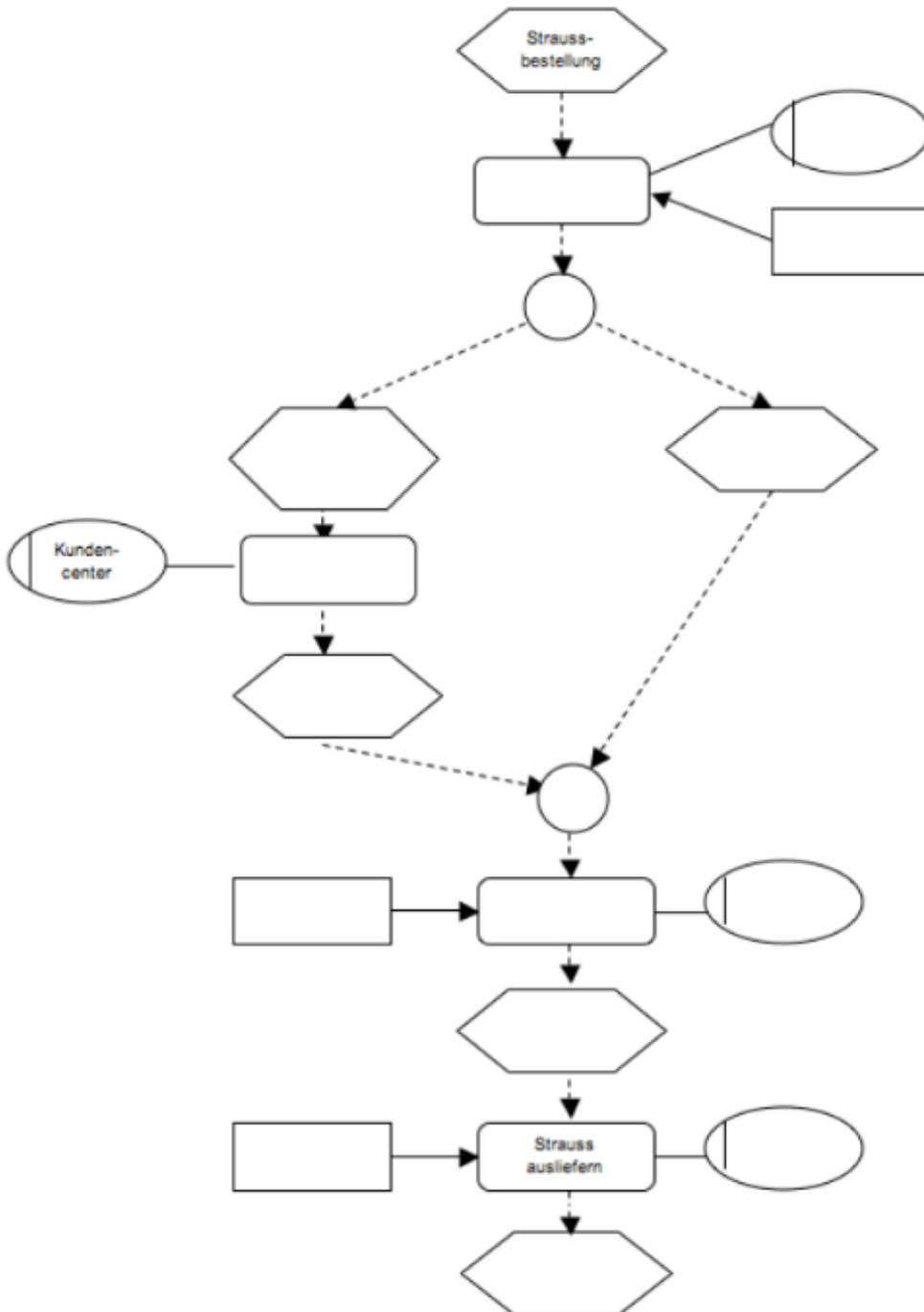
Entwerfe ein UML-Aktivitätsdiagramm für folgenden Arbeitsprozess:

Jan geht jeden Morgen als erstes zum Kaffeeautomaten und schaltet ihn ein. Während der Automat aufwärmt, holt er eine Tasse und spült sie. Beim Spülen unterhält er sich mit seinem Kollegen Michael. Dann gießt er sich seinen Kaffee ein. Anschließend nimmt er Milch und Zucker und verabredet sich gleichzeitig mit Michael zum Abendessen. Danach geht er zu Jürgen. Falls Jürgen da ist redet er eine Weile mit ihm. Abschließend kehrt er beschwingt in sein Büro zurück.

Aufgabe 6 (EPK, BPMN)

Gegeben ist der folgende Ablauf bei der Online-Bestellung eines Blumenstraußes: Die über das Internet aufgegebenen Bestellungen werden zuerst vom Kundencenter überprüft. Sind die Bestelldaten nicht korrekt oder ist die Bestellung nicht plausibel, so nimmt das Kundencenter mit dem entsprechenden Kunden Kontakt auf, um die offenen Fragen zu klären. Ist mit der Bestellung alles in Ordnung, bzw. sind die offenen Fragen geklärt, dann stellt die Floristin mit Hilfe der Bestelldaten den Strauß zusammen. Sobald der Strauß fertig ist, wird er vom Lieferdienst an die auf der Bestellung vermerkte Adresse geliefert.

Vervollständige die unten dargestellte Ereignisgesteuerte Prozesskette (EPK) durch Befüllung aller leeren Ereignisse, Funktionen, Konnektoren, Organisationseinheiten und Informationsobjekte. (Modelliere den gleichen Geschäftsprozess anschließend in der BPMN.)



AUFGABEN OHNE LÖSUNG

Aufgabe 7 (UML-Aktivitätsdiagramm)

Ein IT-Unternehmen verwendet einen agilen Prozess zur Erstellung seiner Software. Die Entwicklung erfolgt iterativ, in jeder Iteration wird ein Modul realisiert. Modelliere den folgenden Entwicklungsprozess mit Hilfe eines UML-Aktivitätsdiagramms:

Zu Beginn werden alle zu realisierenden Module identifiziert und daraufhin wird ein Projektplan erstellt. Die nun folgenden Iterationen sind so aufgebaut, dass zuerst die Anforderungen für das jeweilige Modul erhoben werden. Dann wird der Entwurf erstellt. Danach wird gleichzeitig das Modul implementiert und entsprechende Unit Tests erstellt. Wenn diese beiden Aktivitäten beendet sind, wird eine abschließende Qualitätssicherung für dieses Modul durchgeführt und daraufhin das Modul ausgeliefert. Damit ist die Iteration für dieses Modul beendet. Eine Besonderheit des agilen Entwicklungsprozesses liegt darin, dass gleichzeitig zu einer Iteration auch beständig Feedback von Benutzern eingeholt wird. Falls noch weitere Module zu entwickeln sind, wird mit der nächsten Iteration fortgefahren, ansonsten endet der Prozess.

Aufgabe 8 (BPMN)

Erstelle ein BPMN-Modell für folgenden Geschäftsprozess eines Neuwagenverkaufs mit direktem Finanzierungsabschluss an dem die Abteilungen *Verkauf*, *Vorbereitung* und *Finanzierung* eines Autohauses beteiligt sind:

Der Prozess startet durch den Eingang der schriftlichen Bestellung eines Neuwagens in der Abteilung *Verkauf*. Dort wird die Bestellung in ein System eingegeben und es läuft ein Unterprozess ab, der auf der Hauptprozessebene nicht weiter abgebildet ist. Der Unterprozess liefert das Ergebnis, ob der Neuwagen bei der Fabrik bestellt werden muss oder nicht. Ist keine Bestellung erforderlich, bereitet die Abteilung *Vorbereitung* den Neuwagen für die Auslieferung vor. Ist eine Bestellung bei der Fabrik erforderlich, wird ein Unterprozess zur Bestellung ausgelöst. Der Unterprozess liefert das Ergebnis, ob der Neuwagen verfügbar ist. Ist er nicht verfügbar, endet der Prozess ohne Verkauf. Das Terminierungsende bewirkt, dass alle im Prozess aktiven Flüsse (z.B. die Organisation der Finanzierung) sofort enden. Ist er verfügbar, wird der Neuwagen geliefert (Zwischenereignis) und die Abteilung *Vorbereitung* bereitet den Wagen für die Auslieferung vor. Gleich nach der Eingabe der Bestellung im System organisiert die Abteilung *Finanzierung* in einem Unterprozess die Finanzierung. Ist keine Finanzierung verfügbar, endet der Prozess ohne Verkauf. Das Terminierungsende bewirkt, dass alle im Prozess aktiven Flüsse (z.B. die Bestellung des Neuwagens bei der Fabrik) sofort enden. Ist die Finanzierung verfügbar und sind die Vorbereitungen der Abteilung *Vorbereitung* abgeschlossen, wird in der Abteilung *Finanzierung* die Bestellung abgeschlossen. Die Abteilung *Vorbereitung* übergibt den Neuwagen an den Kunden. Der Prozess endet und es ist zum Verkauf des Neuwagens gekommen.