

Software Engineering Übungsblatt 04

Abgabefrist: Mittwoch - 16.11. - 23:59

Aufgabe 1 Vorgehensmodelle (4Punkte)

Beantworten Sie die folgenden Fragen in je 1-2 Sätzen.

1. Was ist der Unterschied zwischen iterativen und linearen Vorgehensmodellen?

Beim linearen Vorgehen durchläuft man die die Phasen einer festen Reihenfolge (mit eventuellen Rückkopplungen). Bei iterativen Methoden werden die Phasen in mehreren Iterationen durchlaufen.

2. Welche Vorteile hat das Wasserfallmodell gegenüber dem Lebenszyklusmodell?

Im Wasserfallmodell gibt es in jeder Phase eine Rückkopplung zur vorherigen Phase. Im Lebenszyklusmodell ist eine solche Rückkopplung nicht vorgesehen, sodass die Ergebnisse der vorherigen Phase nicht mehr hinterfragt werden können.

3. Wie erhöht Extreme Programming die Kundenzufriedenheit?

Durch eine enge Zusammenarbeit mit Benutzern können Probleme frühzeitig erkannt werden und Anforderungen ggf. verfeinert oder angepasst werden.

4. Was ist der Unterschied zwischen dem Product-Backlog und dem Sprint-Backlog im Scrum?

Das Product-Backlog beinhaltet alle bekannten Anforderungen an das System. Das Sprint-Backlog enthält konkrete Aufgaben für den aktuellen Sprint.

Aufgabe 2 Anwendungsfalldiagramme (4 Punkte)

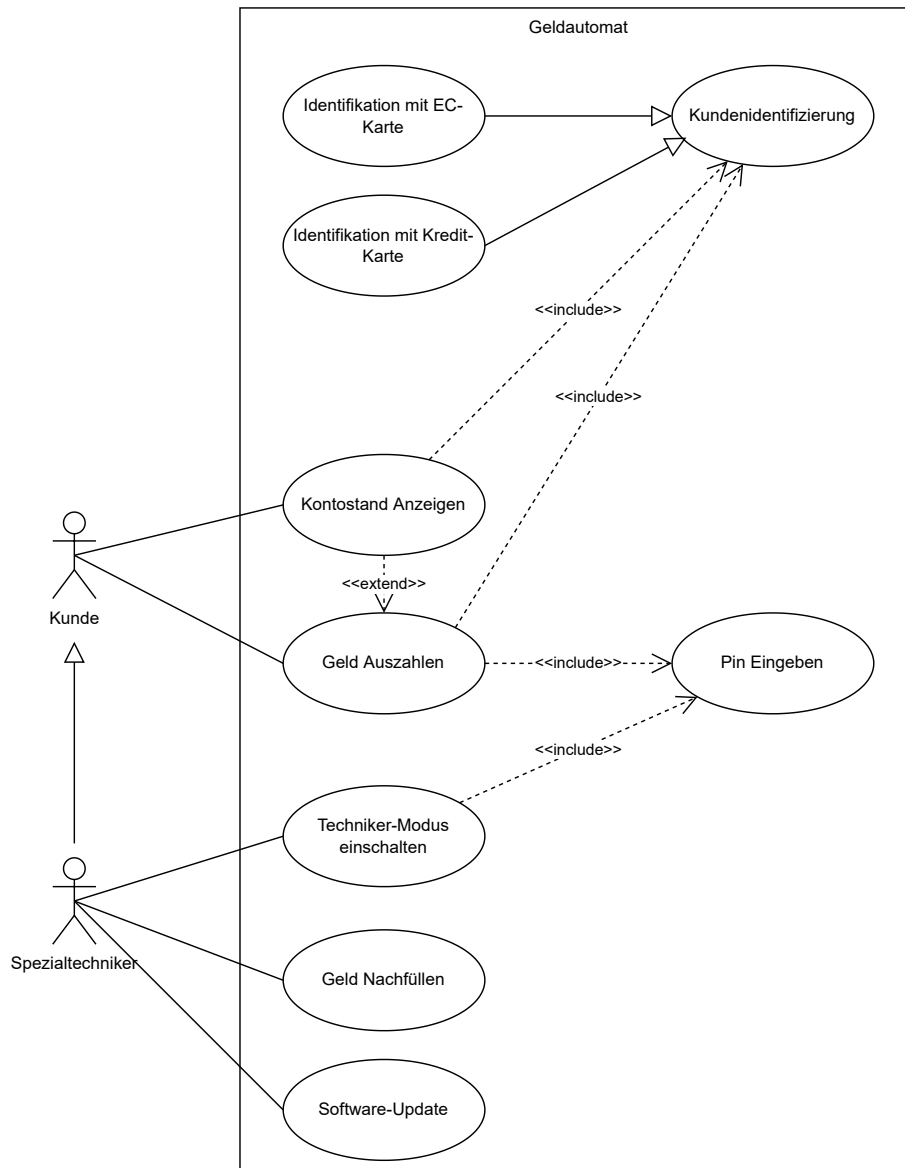
Ihre Firma soll einen Geldautomaten inklusive Software für ein großes Bankhaus entwickeln. Der Kunde beschreibt Ihnen die Funktionalität des Automaten wie folgt:

Ein Kunde kann überhaupt nur mit dem Automaten interagieren, wenn er sich identifiziert. Das passiert, indem er eine Chipkarte einführt. Das muss nicht nur eine EC-Karte sein, unser Bankhaus unterstützt auch Kreditkarten. Nach dem Einführen der Chipkarte, egal von welchem Typ, hat der Kunde dann ein paar Möglichkeiten: Er kann sich den Kontostand anzeigen lassen, oder Geld ausgezahlt bekommen, aber maximal 10.000€ am Tag. Bei mehr müsste er an einen Schalter gehen. Wenn der Kunde aber irgendeine Aktion macht, bei der sein Guthaben belastet wird, reicht die Karte zur Identifikation allein nicht aus, dann muss der Kunde auch noch seinen vierstelligen PIN-Code eingeben. Nach der Auszahlung von Geld hat der Kunde außerdem die Möglichkeit, sich nochmal seinen Kontostand anzusehen.

Und dann gibt's ja auch noch den Fall, dass der Automat gewartet werden muss. Das macht ein Spezialtechniker. Er füllt frisches Geld nach, kann aber auch ein Softwareupdate

aufspielen. Um all diese Sachen machen zu können, hat der Automat eine Art Techniker-Modus. Um da rein zu kommen, muss er einen Geheimcode eingeben, das ist bei uns einfach immer 0000. Damit der Techniker dann auch alle Funktionen ausprobieren kann, verfügt er bei uns außerdem über ein extra Testkonto.

1. Zeichnen Sie ein UML-Anwendungsfalldiagramm für die beschriebene Anwendung.

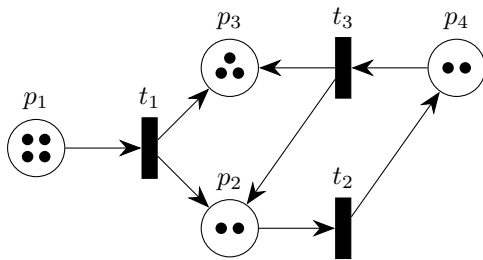


2. Nennen Sie ein weiteres Beschreibungsmittel für Anforderungen und erläutern Sie die Vorzüge und Nachteile gegenüber Anwendungsfalldiagrammen.

Story Cards. Use-Cases auf Storycards meist genauer spezifiziert als in Anwendungsfalldiagrammen. Auch Schätzung Aufwand und Risiko jedes Use-Cases ist auf Story Cards angegeben. Im gegensatz dazu sieht man beim Anwendungsfalldiagramm alle Anforderungen auf einen Blick und vor allem auch im Kontext vom Gesamtsystem, was bei Story Cards unübersichtlich werden kann.

Aufgabe 3 Petri-Netze (4 Punkte)

Betrachten Sie folgendes Petri-Netz:



1. Hat das Petri-Netz eine Verklemmung? Begründen Sie unter Verwendung der Bezeichnungen p_i für die Stellen und t_i für die Transitionen!

Nein, das Petri-Netz ist frei von Verklemmungen, denn t_2 oder t_3 im rechten Zyklus kann immer schalten.

2. Ist das Petri-Netz lebendig? Begründen Sie Ihre Antwort kurz!

Nein, das Petri-Netz ist nicht lebendig. Wenn aus p_1 alle Token entfernt wurden, so kann t_1 nicht mehr schalten.

3. Wie viele Zustände hat der Erreichbarkeitsgraph dieses Petri-Netzes? Begründen Sie Ihre Antwort kurz.

Der Erreichbarkeitsgraph ist unendlich groß, da mit jedem Schalten von t_3 ein zusätzlicher Token generiert wird.

4. Kann ein verklemmtes Petri-Netz lebendig sein? Begründen Sie!

Eine Verklemmung ist eine initial erreichbare Markierung, unter der keine Transition aktiviert ist. Ein Petri-Netz ist lebendig, wenn für jede Transition von jeder initial erreichbare Markierung eine Markierung erreichbar ist, unter der diese Transition aktiviert ist. Von einer Verklemmung aus kann gar keine andere Markierung mehr erreicht werden, sodass ein verklemmtes Petri-Netz nicht lebendig ist.