

Name: _____		Matr-Nr: _____	
DHBW STUTTGART		Ausbildungsbereich:	Technik
		Fachrichtung:	Informationstechnik/ Information Technology Angewandte Informatik
Deckblatt PROBE-KLAUSUR		Studienjahrgang / Kurs:	TINF18C
		Studienhalbjahr:	3+4
Datum: 06.03.2020		Bearbeitungszeit: 60 Minuten	
Studienfach: Software Engineering I (Teilprüfungsleistung)		Dozenten: Markus Rentschler Christian Ewertz	
Hilfsmittel: keine			
Punkte: _____		Note: _____	

1. Sind Sie gesund und prüfungsfähig?
2. Sind Ihre Taschen und sämtliche Unterlagen, insbesondere alle nicht erlaubten Hilfsmittel, seitlich an der Wand zum Gang hin abgestellt und nicht in Reichweite des Arbeitsplatzes?
3. Haben Sie auch außerhalb des Klausorraumes im Gebäude keine unerlaubten Hilfsmittel oder ähnliche Unterlagen liegen lassen?
4. Haben Sie Ihr Handy ausgeschaltet und abgegeben?

(Falls Ziff. 2 oder 3 nicht erfüllt sind, liegt ein Täuschungsversuch vor, der die Note „nicht ausreichend“ zur Folge hat.)

Hinweise:

Die Lösung muss auf dem ausgeteilten Blättersatz untergebracht werden. Der Platz reicht auf jeden Fall aus.

Verwenden Sie keinen Bleistift und keinen Rotstift.

Es sind alle ausgegebenen Blätter wieder abzugeben.

Die meisten Aufgaben sind so gestellt, dass sie unabhängig voneinander lösbar sind.

Aufgabe 1: Systemmodellierung (ca. 35 min, 35 Punkte)

Frage 1.1: Aufzugssteuerung

(15 Punkte)

Während der Systemanalyse bei der Automatisierung einer Aufzugssteuerung soll geklärt werden, wie das Projekt in verschiedene Aufgabenbereiche gegliedert wird. Grundsätzlich wird zwischen Hardware- und Softwarekomponenten unterschieden.

a) Funktionsbaum

(7 Pkt)

Bringen Sie die im Folgenden aufgelisteten Komponenten in eine sinnvolle Struktur und zeichnen Sie einen Funktionsbaum.

- | | |
|--------------------------------------|------------------------------|
| ▪ Lichtschranke (Türen) | ▪ Motoren für Kabinen |
| ▪ Alarm- und Fehlerbehandlung | ▪ Motoren für Türen |
| ▪ Bedienelemente | ▪ Regelung für Kabinenmotor |
| ▪ Ermittlung der aktuellen Fahrziele | ▪ Türöffnungssensoren |
| ▪ Kabinengeschwindigkeitsmesser | ▪ Türsteuerung |
| ▪ Kabinenpositionssensoren | ▪ Verwaltung der Fahrwünsche |
| ▪ Kabinensteuerung (Normalbetrieb) | |

b) Entscheidungstabelle

(8 Punkte)

Das Sollverhalten der Aufzugstüren jedes Stockwerks soll in einer Entscheidungstabelle modelliert werden.

1. Bezüglich der Türöffnung sollen die drei Zustände "zu", "halb offen" und "offen" unterschieden werden.
2. Bei der Kabinenposition wird lediglich geprüft, ob sie sich im zugehörigen Stockwerk befindet (und zwar angehalten) oder nicht.
3. An den Motor zur Türbewegung wird das Signal "Tür auf" bzw. "Tür zu" geschickt.
4. Im Innern einer Kabine kann vom Benutzer explizit über jeweils einen Knopf die Türe geöffnet bzw. geschlossen werden.
5. Der Knopf zur Türöffnung hat aus Sicherheitsgründen eine höhere Priorität.
6. Liegt eine Fahratanfrage in ein anderes Stockwerk vor, so schließen sich die Türen automatisch.
7. Aus Sicherheitsgründen gehört zu jeder Tür eine Lichtschranke, die bei Unterbrechung die Türen sofort öffnet.

In zwei kritischen Situationen wird die weitere Steuerung einer Alarmkomponente übergeben:

- Die Lichtschranke ist unterbrochen, obwohl die Türen geschlossen sind.
- Die Türen sind nicht geschlossen, obwohl sich der Aufzug momentan nicht im zugehörigen Stockwerk befindet.

- a) Vervollständigen sie Entscheidungstabelle
- b) Überprüfen Sie die Tabelle auf Vollständigkeit
- c) Ergänzen Sie die eventuell fehlende Regeln.

ET Aufzugstür		Türe schließen	Türe öffnen	Fahratanfrage	Benutz aktiviert Lichtschranke	Fehler 1	Fehler 2
		R1	R2	R3	R4	R5	R6
B1	Türzustand (zu, halb offen, offen)						
B2	Kabine in aktuellem Stock						
B3	Tür auf – Knopf						
B4	Tür zu – Knopf						
B5	Fahratanfrage von anderem Stock						
B6	Lichtschranke unterbrochen						
A1	Tür auf - Signal						
A2	Tür zu - Signal						
A3	Aktiviere Alarmkomponente						

Prüfung der Vollständigkeit



Frage 1.2: Café-Automat - Zustandsautomat

(20 Punkte)

Sie entwickeln für einen namhaften Hersteller einen Kaffeeschnellautomaten der Einsteigerklasse, welcher nur über ganz wenige Funktionen verfügt.

1. Der Kaffeeautomat besitzt einen Schalter S1 um ihn ein- und auszuschalten.
2. Er besitzt einen Taster T1 um zwischen den Getränken Café, Espresso und Café Latte zu wechseln.
3. Er besitzt einen Taster T2 um die Milchmenge zu beeinflussen. Dieser wechselt zwischen „ohne Milch“, „wenig Milch“ und „viel Milch“.
4. Er besitzt einen Taster T3 um die Zuckermenge zu beeinflussen. Dieser wechselt zwischen „kein Zucker“ und „Zucker“.
5. Ebenso verfügt er über 2 Taster (Menge- und Menge+) um die Menge des zubereiteten Kaffees (Tassengröße) in 6 Stufen zu regeln.
6. Da es ein sehr einfaches Modell ist, kann es sich die Einstellungen nicht merken und ist beim Einschalten grundsätzlich auf der Mengengröße 3, Café schwarz mit Zucker eingestellt.

- a) Zeichnen Sie den Zustandsautomaten für die Steuerung des Automaten als parallelen Automaten.

(Verwenden Sie die Vorlage auf der nächsten Seite.)

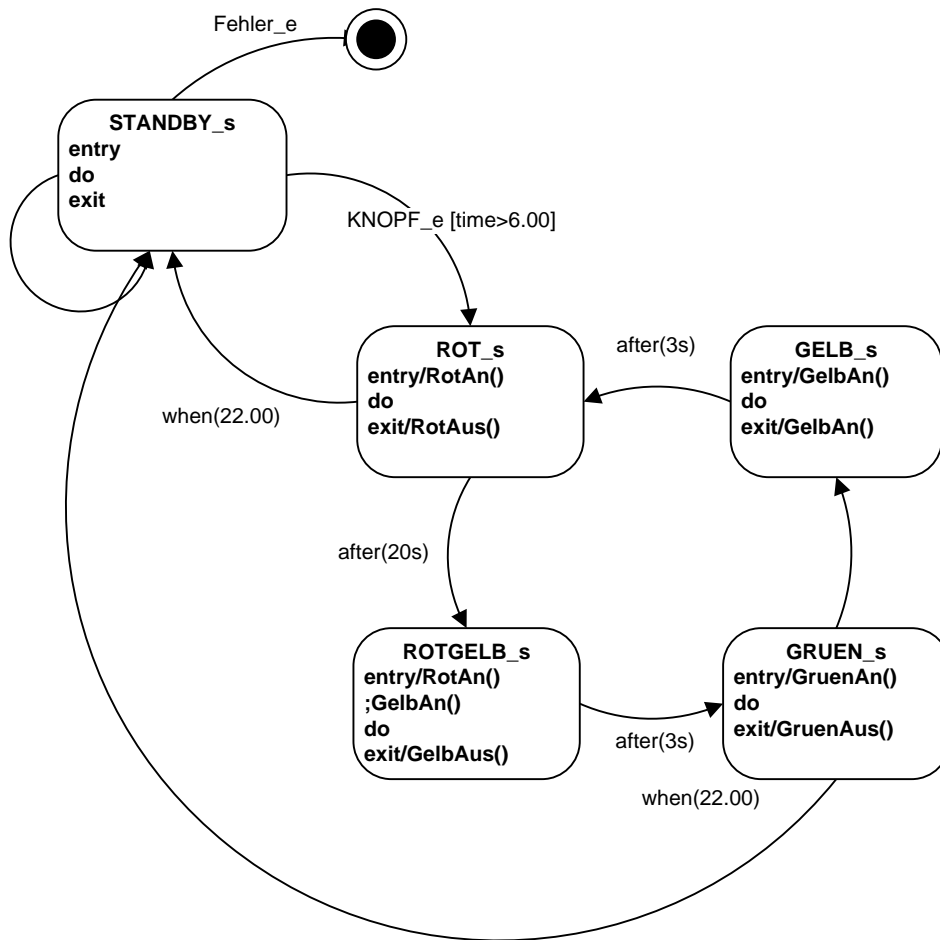
- b) Natürlich benötigt dieser Automat noch einen Taster T4 um die Zubereitung des Getränkes zu beginnen. Während der Zubereitung können die Einstellungen nicht geändert werden (Taster T2, T3, Menge+ und Menge- sind ohne Wirkung). Ebenso kann die Maschine nicht ausgeschaltet werden, bis die Zubereitung beendet wurde.

Ergänzen Sie den Automaten aus Aufgabe a) mit dieser Funktionalität (am besten in einer anderen Farbe).

- c) Schreiben Sie für den Teilautomaten des Tasters T3 ein Programm in Pseudocode. Vergessen Sie die Deklaration der Enumeratoren nicht.

Ampelschaltung

- d) Praktikantenalarm! Die Modellierung dieser Ampelschaltung ist dem Praktikanten nicht so gut gelungen. Sie enthält sowohl Mängel in der Modellierung als auch funktionale Ungereimtheiten. Welche?



Lösung a) und b)

Entry /
Exit /
Do /

Automat_on	

Lösung c)

Aufgabe 2: Verständnisfragen (ca. 25 min, 36 Punkte)

Frage 2.1: Kurzfragen

(7,5 Punkte)

Stimmen Sie folgenden Aussagen zu? Falls nicht, begründen Sie bzw. stellen Sie die Aussage richtig.

Aussage	JA	NEIN
Software Engineering und Programmieren ist das Gleiche.		
Die FMEA ist ein Verfahren zur Aufwandsschätzung.		
Logische Basiskonzepte beschreiben die zur Laufzeit unveränderlichen Aspekte wie Architektur und Datenmodelle.		
ERM Diagramme eignen sich zum Modellieren von Datenstrukturen.		
Bei agilen Softwareentwicklungsmethoden sind die Entwickler an keine Anforderungen gebunden. Die Arbeitsweise ähnelt dem Code and Fix Verfahren.		
Funktionale Anforderungen geben an, was das System können soll. Nichtfunktionale Anforderungen betreffen Eigenschaften, die das System zusätzlich zur Funktionalität aufweisen soll.		
Validierung prüft, ob ich das richtige System entwickelt habe. Verifikation überprüft ob es fehlerfrei funktioniert.		

Frage 2.2: Projektmanagement**(4,5 Punkte)**

- a) Was versteht man unter dem magischen Dreieck des Projektmanagements bzw. Software-Engineerings. (1,5 Punkte)

- b) In einem Projektteam gibt es verschiedene Projektrollen. Was sind die Aufgaben des Projektleiters und welche des Requirements Engineers (mindestens 3 Stichpunkte pro Rolle)? (3 Punkte)

Frage 2.3: Risikomanagement**(3 Punkte)**

- a) Nach welchen drei Kriterien werden bei einer FMEA die Risiken bewertet. Wie wirken sich diese auf das Gesamtrisiko aus?

Frage 2.4: Aufwandsschätzung

(4 Punkte)

a) Was versteht man unter der Expertenschätzung ?

b) Benennen und beschreiben Sie ein weiteres Schätzverfahren Ihrer Wahl.

Frage 2.5: Testing**(6 Punkte)**

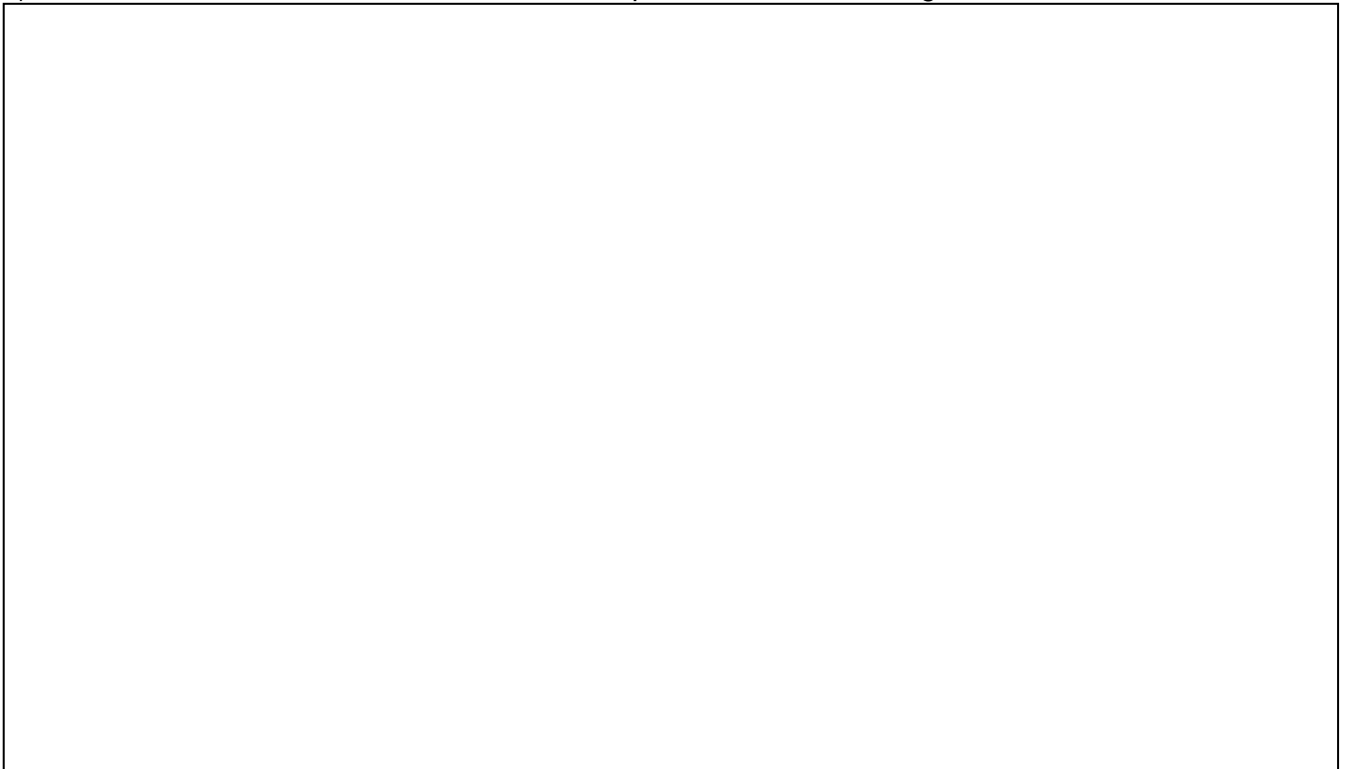
- a) Im Whitebox-Testing spricht man oft von Anweisungsüberdeckung (C_0), Entscheidungsüberdeckung (C_1), Bedingungsüberdeckung (C_2) und Pfadüberdeckung. Erklären Sie um was es sich hier handelt und differenzieren Sie diese Begriffe.

- b)** Was versteht man unter Äquivalenzklassenbildung und Grenzwertanalyse?

Frage 2.6: Requirements Engineering

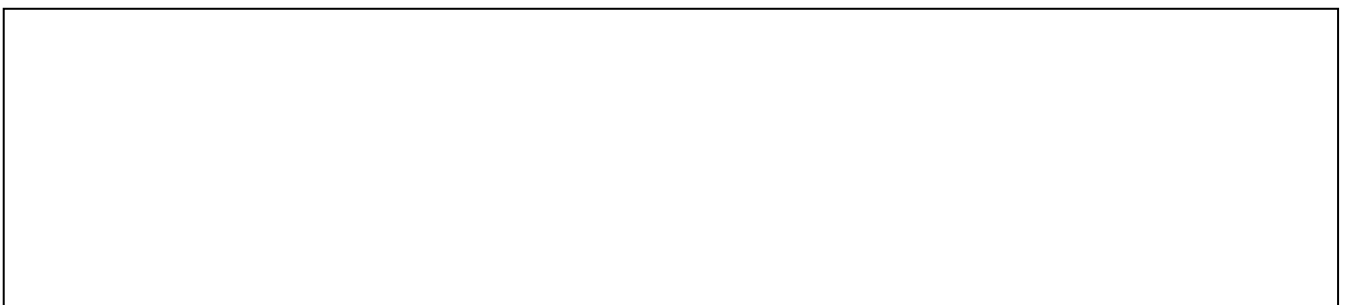
(8 Punkte)

a) Skizzieren Sie eine Schablone für natürlich sprachliche Anforderungen.



b) Was ist der Unterschied zwischen funktionalen und nichtfunktionalen Anforderungen? Klassifizieren Sie die folgenden Anforderungen jeweils als funktional oder nicht funktional.

- /A1/ Das System muss IEC 61131-konform sein.
- /A2/ Die Anwendung kommuniziert über TCP/IP mit einem vorhandenen Datenbankserver.
- /A3/ Die Software muss einfach auf unterschiedliche Plattformen übertragen werden können.
- /A4/ Die Software muss Daten für maximal 1000 Benutzer verwalten.
- /A5/ Die sicherheitskritischen Teilfunktionen müssen sehr zuverlässig sein.
- /A6/ Die Bedienung erfolgt über einen berührungsempfindlichen Bildschirm (Touchscreen).
- /A7/ Der Benutzer kann die Prozesstemperatur stufenlos einstellen.
- /A8/ Das System muss objekt-orientiert entwickelt werden.



Frage 2.7: Prozessdokumente/Vorgehensmodelle**(6 Punkte)**

- a) Erläutern Sie kurz die wesentlichen Unterschiede zwischen System Architecture Specification und System Requirements Specification.

- b) Sie werden als Consultant in eine Software-Entwicklungsabteilung eines Herstellers von Embedded Steuerungen gerufen. Sie sollen den Entwicklungsprozess beurteilen und Vorschläge unterbreiten.

Beim Studium der Projektakten stoßen Sie auf folgende Informationen über vergangene Projekte:

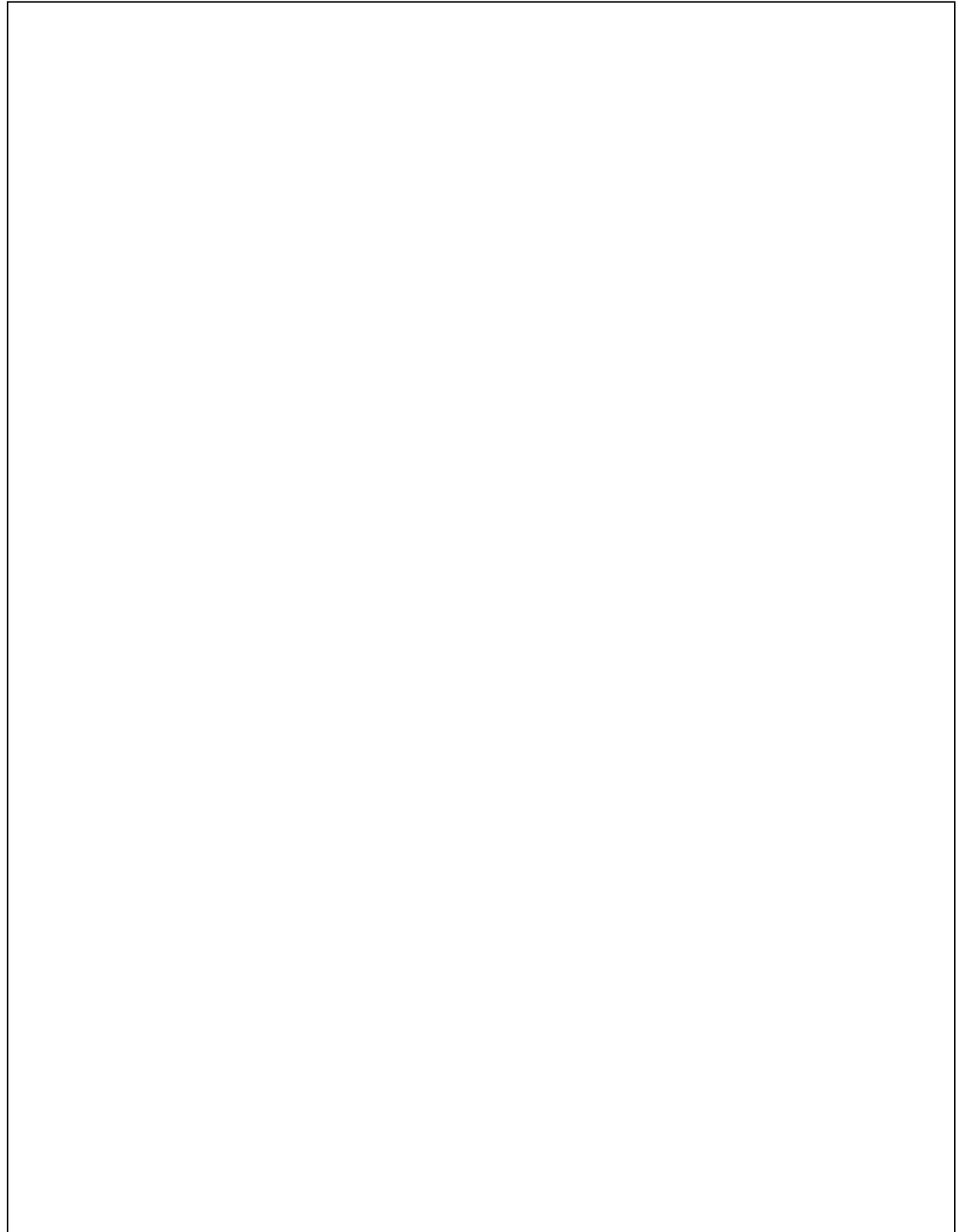
- Die Kunden reklamieren enorme Qualitätsmängel in der Software. Ebenso fühlen sie sich nicht in ihren Anliegen verstanden, was sie als Funktionalitäten in den Geräten wünschen.
- In den Büchern finden Sie folgende Budgetaufteilung

Anforderungserfassung	5%
Analyse	-
Design	15 %
Codierung	70 %
Test	5%

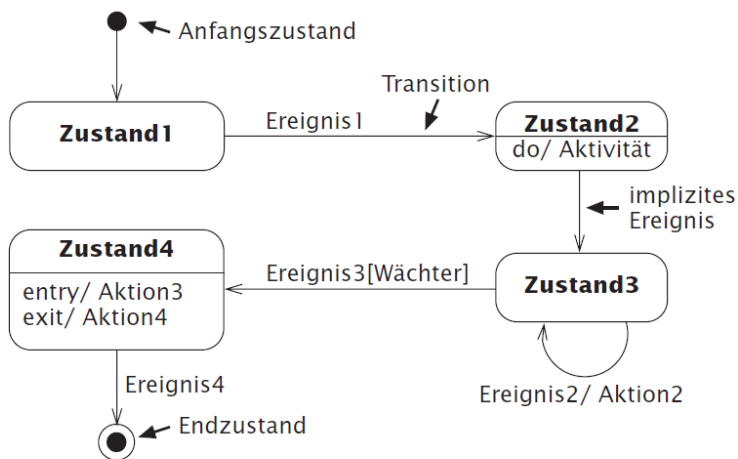
- Die Dokumentation ist von großer Wichtigkeit, da auch Offshore entwickelt wird. Sie ist jedoch leider kaum vorhanden.
- Gegen Ende des Projektes werden immer noch Anforderungen neu definiert.

Frage:

Sie wollen der Firmenleitung die möglichen Gründe für die Probleme aufzeigen und ein Verfahrensmodell vorschlagen. Welches würden Sie wählen? Bitte begründen. Wie würde das Modell die Probleme beseitigen?



UML: Notation Zustandsdiagramm



UML Sequenzdiagramm; Notation von Lebenslinien und Nachrichten

