Software-Analysemethoden
SoSe 2003

Aufgabe 1: Softwareentwicklungsmodelle

Für die folgenden beiden Projekte wird nach einem passenden Modell für die Software-Entwicklung gesucht:

- die Steuerung eines Antiblockiersystems in einem Auto;
- ein Flugbuchungssystem, das sowohl im Reisebüro für die Eingabe von Kunden- und Reisedaten eingesetzt werden soll als auch auf Basis dieser Daten das Einchecken von Passagieren am Flughafen implementiert.

Charakterisieren Sie diese beiden Anwendungen kurz und geben Sie ein passendes Softwareentwicklungsmodell dafür an. Begründen Sie Ihre Auswahl.

Aufgabe 2: Qualitätssicherung

Qualitätsmerkmale von Software sind

Betriebssicherheit	Verständlichkeit	Portierbarkeit
Sicherheit	Testbarkeit	Benutzerfreundlichkeit
Zuverlässigkeit	Anpassungsfähigkeit	Wiederverwendbarkeit
Flexibilität	Modularität	Leistungsfähigkeit
Stabilität	Komplexität	Erlernbarkeit

Erläutern Sie wie die Beurteilung der Software-Qualität auf Grundlage dieser Merkmale erfolgen kann und skizzieren Sie kurz für jedes Merkmal wie es für ein gegebenes Softwareprodukt geprüft werden kann. Mögliche Ansätze für die Prüfung sind dabei prinzipiell Test, Analyse und Inspektion.

Aufgabe 3: Coding Rules und Qualitätsstandards

Ein Entwickler, der ein guter Programmierer ist, erstellt Software, die üblicherweise nur eine geringe Anzahl von Fehlern enthält; er ignoriert dabei allerdings die nach Firmenstandard und Projektstandard vorgegebenen Qualitätsstandards und Coding Rules.

Wie sollten die verantwortlichen Manager (Projektleiter und Abteilungsleiter) auf dieses Verhalten reagieren? Begründen Sie Ihre Entscheidung.

Aufgabe 4: Code-Inspektionen

Untersuchen Sie, welche der in der Astrium Software Produktsicherungsinstruktion "Software Coding Verification Assurance Standard" im Anhang B aufgeführten Prüfpunkte für C-Programme durch einen Einsatz von Splint überprüfbar sind.

Als Ergebnis ist eine Tabelle zu erstellen, die für die relevanten Prüfaspekte angibt, ob sie mit Splint prüfbar sind und wenn ja, mit welchen Optionen oder Kombinationen von Optionen und Annotationen.

Aufgabe 5: Metriken

Wartbarkeit ist ein wichtiges Merkmal für Software, die für eine langjährige Betriebsphase vorgesehen sind. Leiten Sie mit Hilfe des Goal-Question-Metrics Ansatzes ab, welche Daten Sie für eine Analyse dieser Eigenschaft mittels Metriken benötigen. Dabei sollen für das Ziel "Wartbarkeit" geeignete Fragen identifiziert werden, deren Beantwortung Auskunft über Einhaltung bzw. Verletzung der Eigenschaft geben und diesen Fragen wiederum Daten zugeordnet werden, deren Messung zur Beantwortung der Fragen notwendig sind.

Anmerkung: Wesentliche Aspekte von Wartbarkeit sind Lesbarkeit, Modularität, Portierbarkeit, Fehlerfreiheit.

Beschränken Sie sich dabei auf Produktmetriken und unterscheiden Sie solche Daten, die absolut gesehen Auskunft geben und solche, deren Verlauf Rückschlüsse auf die Eigenschaften zulassen. Begründen Sie Ihre Bewertung und gehen Sie auch darauf ein wie zuverlässig die Metriken als Nachweis für die Eigenschaft sind.

Aufgabe 6: Fehlerbaum-Analyse

Die folgende Beschreibung ist Bestandteil der Anforderungen für eine Implementierung eines Systems zum Steuern einer Fahrstuhlanlage.

Allgemeine Beschreibung

Die folgenden Aufgaben sind Bestandteil der Entwicklung eines Fahrstuhlsteuersystems. Als Grundannahme gehen wir dabei von einem System mit 2 Fahrstühlen aus, die jeweils 10 Stockwerke bedienen. Das Gesamtsystem teilt sich dabei in drei Komponenten:

- die Steuerung und Überwachung der Anzeigen und Kontrollen in jedem der Stockwerke
- die Steuerung und Überwachung der Anzeigen und Kontrollen in jeder Kabine
- die Umsetzung des zentralen Steuermechanismus, der die generelle Bewegung und Position der Fahrkabinen überwacht und steuert.

In jedem Stockwerk befinden sich

- je ein Knopf pro Fahrrichtung für die Fahrstuhlanforderung
- Anzeigen für jeden Fahrstuhl über die aktuelle Position und Fahrtrichtung
- je zwei Sensoren pro Fahrstuhl, die oberhalb und unterhalb der Tür im Fahrstuhlschacht angebracht sind und bei gleichzeitiger Aktivierung anzeigen, dass die Fahrkabine in Position steht.
- eine Tür, die automatisch mit der Kabinentür geöffnet und geschlossen wird. Diese Tür besitzt keine eigene Steuerung, aber einen Sensor zur Überwachung, ob sie geöffnet oder geschlossen ist.

In jeder Fahrkabine befinden sich

- je ein Knopf pro Stockwerk zur Angabe des gewünschten Fahrziels
- eine Anzeige für die aktuelle Position und Fahrtrichtung
- ein Knopf für das Schnellschließen der Tür
- ein Knopf für das Offenhalten der Tür
- insgesamt 6 Sensoren (je drei pro Seite) in der Tür zur Erkennung von Hindernissen beim Türschließen
- ein Timer, der nach Angabe des Fahrtziels oder bei vorhandenen Zieleingaben nach Öffnen der Türen als Beantwortung auf eine Anforderungen gestartet wird und bei Ablauf das Signal an die Kabinentür zum Schließen gibt; dieser Timer wird bei Betätigung des Öffne-Knopfes neu gestartet, bei Betätigung des Schließe-Knopfes gestoppt (auf 0 gesetzt).
- ein zweiter Timer, der nach Ablauf des ersten Timers aktiviert wird. Wenn nach Ablauf des zweiten Timers die Tür nicht geschlossen ist, wird ein Fehler an die zentrale Einheit gemeldet. Der zweite Timer wird zurückgesetzt, falls die Tür wieder geöffnet wird, was an einer Reaktivierung des ersten Timers erkennbar ist.

Die zentrale Steuereinheit hat folgende Aufgaben:

- Sie speichert und überwacht die aktuelle Position jeder Fahrkabine und stellt sie den Anzeigeelementen in den Stockwerken und Kabinen zur Verfügung. Als Input wird die Information der Positionssensoren in den Stockwerken verwendet.
- Sie kontrolliert Fahrrichtung und Richtungsänderungen der Kabinen; Ziel ist ein bestmögliches Scheduling in Abhängigkeit von Position und Nutzeranfragen.

- Sie überwacht mögliches Fehlverhalten des Systems, etwa Türblockierungen.

Bei der Realisierung des Systems sind eine Reihe von Regeln zu beachten, die das Gesamtverhalten bestimmen:

- Eine Fahrkabine befindet sich im Stockwerk n wenn oberer und unterer Sensor der Stockwerkserkennung im Stockwerk n aktiviert sind. Das heisst insbesondere, dass sich der Fahrstuhl solange in diesem Stockwerk befindet solange keine andere Stockwerksposition erreicht ist.
- Die Anzeigen geben die letzte gesicherte Position der Kabinen wieder (sowohl innerhalb der Kabine als auch in den Stockwerken)
- die Türen der Kabine (und damit auch die des Stockwerks) öffnen sich ausschließlich dann, wenn sich die Kabine im zugehörigen Stockwerk befindet.
- Der Fahrstuhl kann erst dann fahren, wenn innere und äußere Tür geschlossen sind.
- Das Schließen der Tür wird entweder durch den Signalgeber (Timer) nach Ablauf der Wartezeit aktiviert (Zeitsignal) oder durch den Interrupt, der vom Schnellschließen-Knopf ausgelöst wird.
- Beim Schließen der Tür geht Sicherheit vor: beim Erkennen eines Hindernises wird ein Schnellschließen ignoriert bzw. abgebrochen.
- Das Fahrverhalten soll möglichst effizient sein; d.h. dass unnötige Richtungsänderungen vermieden werden sollen und dass auf eine Anforderung aus dem Stockwerk der jeweils nähere Fahrstuhl die Anfrage annimmt. Dabei soll die erste Bedingung eine höhere Priorität haben als die zweite.

Übersicht über die Software-Architektur

Die Steuersoftware wird verteilt realisiert. Folgende Hauptkomponenten sind dabei vorgesehen:

- zentrale Steuerung, die die Liste der Anforderungen und den Gesamtstatus des Systems verwaltet, die Motorsteuerung für die Kabinenbewegung kontrolliert, die relevanten Daten für die Anzeige in den Fahrstühlen und den Stockwerken an diese weiterleitet und die globale Fehlererkennung realisiert,
- eine separate Steuerung für jede der Kabinen inklusive der Türsteuerung, Anzeigenverwaltung, Anforderungsannahme und Weiterleitung an die zentrale Steuerung
- eine Einheit für die Überwachung der Stockwerke, die die Stockwerksanforderungen entgegennimmt und an die zentrale Steuereinheit weiterleitet, sowie die Stockwerkssensoren allgemein überwacht, Fehler an die zentrale Steuerung weiterleitet, und die Daten aus der zentralen Einheit für die Anzeigen umsetzt.

Der Türsteuermechanismus selber besteht wieder aus einer Reihe von Komponenten, die parallel laufen:

- einer Komponente zur Stockwerkserkennung, der die Sensoren des Stockwerks abfragt und Informationen der zentralen Steuereinheit verwendet;
- eine Motorsteuerung, die die Motoren steuert, die die Türen öffnen und schließen;
- eine Komponente zur Hinderniserkennung, die regelmäßig die Türsensoren abfragt, wenn die Tür im Zustand Tür schliessenïst und bei Erkennung eines Hindernisses der Türkontrolleinheit eine Nachricht schickt;
- eine zentrale Kontrolleinheit für die Tür, die die eingehenden Daten aus den Einheiten interpretiert und entsprechend des Türzustands interpretiert, die Motorsteuerung für die Tür per Nachricht aktiviert bzw deaktiviert, das Dateninterface zur zentralen Steuereinheit beinhaltet, und die übrigen Komponenten koordiniert.

Zustandsautomat für Türsteuerung

Zum besseren Verständnis kann das Zustandsdiagramm für den Türmechanismus hinzugezogen werden, das separat als .pdf-Datei verfügbar ist (siehe www-Seiten zur Vorlesung)

Aufgabenstellung

1. Im Rahmen der Entwicklung sollen auch Sicherheitsanalysen durchgeführt werden, unter anderem eine Fehlerbaumanalyse bzgl. der Sicherheit des Türschließmechanismus. Dabei ist eine kritische Situation das Schließen einer Kabinentür obwohl sich eine Person im Türbereich befindet.

   Stellen Sie einen Fehlerbaum auf, der diese kritische Situation analysiert. Die Analyse sollte bis auf die Komponentenebene der Türsteuerung gehen und auch deren Eingaben, Ausgaben, sowie die Sensoren und Aktuatoren (z.B. Motorsteuerung) mit einbeziehen.

   Anmerkungen: Vermeiden Sie die Verwendung von normalenËreignissen im Fehlerbaum; ein Ereignis "Tür schließt sich" im Zusammenhang mit"Person im Türbereich" beispielsweise beinhaltet auch alle anderen normalen Ereignisse, die nicht direkt mit dem Fehlverhalten zusammenhängen und führt zu einem unnötig umfangreichen Fehlerbaum. Alternative: "Tür schließt obwohl Person im Türbereich".

2. Bestimmen Sie nach dem in der Vorlesung vorgestellten Verfahren die minimalen Cut-Sets Ihres Fehlerbaums.

3. Welche Zusatzkomponenten oder zusätzlichen Anforderungen für die Komponenten sind erforderlich um die Fehlerquellenzu vermeiden, die Ihre Analyse identifiziert? Welche Ereignisse sind nicht aus der Software heraus zu kontrollieren oder zu vermeiden? Begründen Sie Ihre Bewertung.