
Software Engineering 2

Pflichtenheft

**Projekt:
Entwicklung eines Software-Systems
zur Simulation der Steuerung eines Liftes**

Inhaltsverzeichnis

1	Einführung	3
2	Dokumentation der Anforderungen	4
2.1	Kontextdiagramm	5
2.2	Satzschablonen	6
2.3	Anwendungsfälle - Benutzersicht.	9
2.4	Anwendungsfälle - Passagiersicht	9
2.5	Qualitätsanforderungen.	10
2.6	Rahmenbedingungen	10
3	Zustandsdiagramm des Liftes	11
4	Glossar	12
4.1	Allgemeiner Glossar.	12
4.2	Projektbezogener Glossar	12

1 Einführung

Im Rahmen der Belegarbeit im Fach SOFTWARE ENGINEERING 2 ist ein Software-System zu implementieren, welches die Steuerung eines Liftes simuliert. Dieses Software-System soll in der Zukunft als Anschauungsmaterial im Lehrbetrieb verwendet werden. Studierenden soll damit ermöglicht werden, die Zusammenhänge zwischen real existierenden Automaten und der Thematik der Zustandsdiagramme zu erfahren.

In diesem Zusammenhang ergeben sich zusätzlich zu den Anforderungen an das Teilsystem *Liftsteuerung*, spezielle Anforderungen an das Teilsystem *Visualisierung* aus der Sicht des Lehrbetriebes. Im folgenden werden diese beiden Teilsysteme daher an verschiedenen Stellen getrennt voneinander betrachtet und beschrieben.

Das vorliegende Pflichtenheft dient der Beschreibung und Vereinbarung von Anforderungen an das Gesamtsystem und besitzt Vertragscharakter. Die darin enthaltenen Anforderungen wurden auf Basis von Kundengesprächen und Kundenvereinbarungen sowie der Aufgabenstellung formuliert¹.

¹ Meeting Minutes und Audiomitschnitte der beiden Kundengespräche sind unter folgendem Link zu finden: <http://goo.gl/UVHn2G>

2 Dokumentation der Anforderungen

Anforderungen an ein Software-Produkt werden im Allgemeinen zunächst in funktionale und nicht-funktionale Anforderungen unterteilt. Erstere decken dabei die Fähigkeiten und die Beschaffenheiten ab, die der Benutzer der Software zur Problemlösung oder zur Erreichung seines Zieles benötigt. Nicht-funktionale Anforderungen unterteilen sich weiterhin in Rahmenbedingungen und Qualitätsanforderungen.

Im Folgenden werden die funktionalen Anforderungen an das Software-System aus den bereits angesprochen zwei Perspektiven betrachtet. Perspektive *A*) bezieht sich auf das Teilsystem *Visualisierung* und betrachtet es aus der Sicht des Benutzers. Diese Sicht wird im folgenden *Benutzersicht* genannt. Unter Perspektive *B*) wird das Teilsystem *Liftsteuerung* aus der Sicht der Passagiere betrachtet. Diese Sicht wird im folgenden *Passagiersicht* genannt.

2.1 Kontextdiagramm

Um die Verschachtelung der beiden Perspektiven sowie die Schnittstellen des Systems zur Umwelt übersichtlich darzustellen eignet sich ein Kontextdiagramm.

TODO: KONTEXTDIAGRAMM EINFÜGEN

2.2 Satzschablonen

Im Folgenden werden die Anforderungen an das Software-System mit Hilfe von Satzschablonen² dokumentiert.

Deren Auflistung unterscheidet dabei zwischen selbstständigen Systemaktivitäten³ und Benutzerinteraktionen⁴. Weiterhin wird eine Nummerierung vorgenommen, welche in allen Teilen der Software-Dokumentation konsistent benutzt werden.

Selbstständige Systemaktivitäten

- **ELV-001:**
Die Liftsimulation muss den Fahrstuhl nach oben fahren lassen.
- **ELV-002:**
Die Liftsimulation muss den Fahrstuhl nach unten fahren lassen.
- **ELV-003:**
Die Liftsimulation muss das Öffnen der Fahrstuhltür anzeigen.
- **ELV-004:**
Die Liftsimulation muss das Schließen der Fahrstuhltür anzeigen.
- **ELV-005:**
Die Liftsimulation sollte den aktuellen Zustand des Fahrstuhls anzeigen.
- **ELV-006:**
Die Liftsimulation sollte Zustandsübergänge des Fahrstuhls anzeigen.

2 Eine Satzschablone ist ein Bauplan für die syntaktische Struktur einer einzelnen Anforderung. Der Einsatz der Satzschablone unterstützt den Autor einer Anforderung darin, die syntaktische Eindeutigkeit der Anforderung zu erreichen

3 Diese Aktionen werden von der Liftsteuerung selbstständig ausgeführt. Über verschiedene Schnittstellen interagieren Benutzer, Passagier und Sensoren mit dem System.

4 Über Benutzerinteraktionen kann der Benutzer des Software-Systems mit der Liftsteuerung interagieren.

- **ELV-007:**
Die Liftsimulation muss den Wechsel einer Etage anzeigen.
- **ELV-008:**
Die Liftsimulation sollte die Fahrstuhltür selbständig schließen, wenn länger als 3 Sekunden keine Benutzerinteraktion durchgeführt wurde.
- **ELV-009:**
Die Liftsimulation muss eine Überlastung des Fahrstuhls durch zu viele Passagiere anzeigen.
- **ELV-010:**
Die Liftsimulation muss im Falle einer Überlastsituation⁵ in den Zustand *Überlast* wechseln. Ausgehend von diesem Zustand ist es dann nur noch möglich in den vorherigen Zustand zu wechseln, sofern die Überlastsituation durch das Verlassen von Passagieren aufgehoben wurde.

Benutzerinteraktionen

- **ELV-011:** Die Liftsimulation muss dem Anwender die Möglichkeit bieten einen *Fahrtwunsch*⁶ für einen Passagier einzugeben.
- **ELV-012:** Die Liftsimulation muss dem Anwender die Möglichkeit bieten einen Passagier in den Fahrstuhl einsteigen zu lassen.
- **ELV-013:** Die Liftsimulation muss dem Anwender die Möglichkeit bieten einen Passagier aus dem Fahrstuhl aussteigen zu lassen.
- **ELV-014:** Die Liftsimulation muss dem Anwender die Möglichkeit geben, einen *Fahrstuhlruf*⁷ in jedem Stockwerk absetzen zu können.

⁵ Eine Überlastsituation tritt ein, sobald mehr sich als 8 Passagiere im Fahrstuhl befinden.

⁶ Ein *Fahrtwunsch* ist die Eingabe der Zieletage eines Passagiers über die innere Schaltfläche des Liftes.

⁷ Ein *Ruf* wird durch das Betätigen eines Etagenknopfes abgesetzt.

- **ELV-015:** Die Liftsimulation sollte dem Anwender die Möglichkeit bieten eine *Vorrangschaltung*⁸ auswählen zu lassen.

⁸ Der Benutzer kann einen *Monteur* in den Lift einsteigen lassen, welcher die Möglichkeit besitzt einen *Priorisierten Fahrtwunsch* einzugeben.

2.3 Anwendungsfälle - Benutzersicht

In dieser Sicht gibt es einen « *abstrakten* » Anwendungsfall *Passagier/Monteur steuern*. Dieser gliedert sich in die unabhängigen Anwendungsfälle:

- *Passagier-einsteigen* (ELV-012)
- *Passagier-aussteigen* (ELV-013)

TODO: DIAGRAMM EINFÜGEN

Die Eingangs- und Ausgangsdaten, sowie einige Bemerkungen zu den Anwendungsfällen sind in folgender Tabelle zusammengefasst.

Funktion	Eingangsdaten	Ausgangsdaten	Bemerkungen	abstrakter AWD
<i>Passagier einsteigen</i>	Betätigung der entsprechenden Schaltfläche	Visuelle Bestätigung der Eingabe	Betreten mehr als 8 Personen den Lift kann eine Überlastsituation auftreten	Passagier / Monteur steuern
<i>Passagier aussteigen</i>	Betätigung der entsprechenden Schaltfläche	Visuelle Bestätigung der Eingabe		

2.4 Anwendungsfälle - Passagiersicht

Aus dieser Perspektive ergeben sich drei voneinander unabhängige Anwendungsfälle:

- *Fahrtwunsch - Passagier* (ELV-011)
- *Fahrstuhlruf* (ELV-014)
- *Fahrtwunsch - Monteur* (ELV-015)

TODO: DIAGRAMM EINFÜGEN

Die Eingangs-, Ausgangsdaten, sowie Bemerkungen sind wiederum in folgender Tabelle zusammengefasst.

Funktion	Eingangsdaten	Ausgangsdaten	Bemerkungen
<i>Fahrtwunsch Passagier</i>	Etagenwahl (innen)	Leuchten der Wunschetage	Übergabe des Wunsches an Lift- steuerung
<i>Fahrstuhlruf</i>	Drücken eines der beiden Rufknöpfe (außen)	Leuchten der Ruftas- te	Übergabe des Wunsches an Lift- steuerung
<i>Fahrtwunsch Monteur</i>	Drücken des Schlüsselsymbols und der Rufetage	Leuchten der des Schlüsselsymbols und der Wunsche- tage	Übergabe des Wunsches an Lift- steuerung

2.5 Qualitätsanforderungen

Benutzerfreundlichkeit und die intuitive Bedienbarkeit des Software-Systems haben vor allem im Lehrbetrieb große Wichtigkeit. Aktivitäten des Systems sollten erst nach der Interaktion des Benutzers beginnen und nicht automatisch starten. So ist sicher gestellt, dass der Benutzer in jeder Situation genügend Zeit hat, um das vergangene und zukünftige Verhalten des Systems nachvollziehen und durchdenken zu können.

Weiterhin wird sichergestellt, dass jede Interaktion des Benutzers mit dem System eine Rückmeldung an den Benutzer gibt. Hier werden vor allem Methoden der visuellen Rückmeldung Anwendung finden.

2.6 Rahmenbedingungen

3 Zustandsdiagramm des Liftes

4 Glossar

4.1 Allgemeiner Glossar

Anwendungsfall Eine abstrakte Darstellung einer vom Software-System angebotenen Funktionalität (Aktivität). Er kapselt eine Menge von Aktionen, die sequentiell, bedingungsabhängig oder zyklisch abgearbeitet werden. Ein Anwendungsfall wird in Folge von Dateneingaben oder zeitlichen Ereignissen ausgelöst und führt in der Regel zu einem von außen sichtbarem Ergebnis.

Anwendungsfalldiagramm Das Anwendungsfalldiagramm, kurz AWD, stellt die funktionalen Anforderungen (Aktivitäten) aus Sicht des Anwenders dar. Diese Aktivitäten werden zu den Beteiligten aus dem Kontext (Akteuren) in Beziehung gesetzt.

Akteur Ein Akteur ist die abstrakte Darstellung einer externen Instanz, die mit dem System kommuniziert.

4.2 Projektbezogener Glossar