

Pflichtenheft

Gliederung

1. Zielbestimmung
2. Produkteinsatz
3. Produktübersicht
4. Produktfunktionen
5. Produktdaten
6. Produktleistungen
7. Qualitätsanforderungen
8. Benutzeroberfläche
9. Nicht funktionale Anforderungen
10. Technische Produktumgebung
11. Spezielle Anforderungen an die Entwicklungsumgebung
12. Gliederung in Teilprodukte
13. Ergänzungen

1. Zielbestimmung

Das Ziel ist die Entwicklung von Dienstleistungen mittels Service Editors zu ermöglichen.

1.1 Musskriterien

- Es muss möglich sein, auf der Basis eines Metamodells einzelne Komponenten und deren Komposition zu modellieren.
- Das Zusammenspiel zwischen den funktionalen und nichtfunktionalen Aspekten muss berücksichtigt werden.
- Die einzelnen Komponenten & Modelle des Dienstleistungsentwurfs müssen erstellt und bearbeitet werden können.

1.2 Wunschkriterien

- Es sollte möglich sein ein Metamodell zu laden.
- Die Erweiterung der Anwendung um Marketingkonzept-Modul sollte möglich sein.
- Die Erweiterung der Anwendung um weitere Funktionalitäten sollte möglich sein.

1.3 Abgrenzungskriterien

- Es wird keine Anmeldefunktion eingebunden.

2. Produkteinsatz

Das Produkt wird für den Lehrstuhl „Anwendungsspezifische Informationssysteme“ am Institut für Informatik der Universität Leipzig im Rahmen vom SWT-Praktikum entwickelt, sollte aber im Dienstleistungsbereich verwendet werden.

2.1 Anwendungsbereiche

- Entwicklung von Dienstleistungen

2.2 Zielgruppen

- Mitarbeiter eines Dienstleistungsunternehmens, die neue Dienstleistungen entwerfen.

2.3 Betriebsbedingungen

- Büroumgebung

3. Produktübersicht

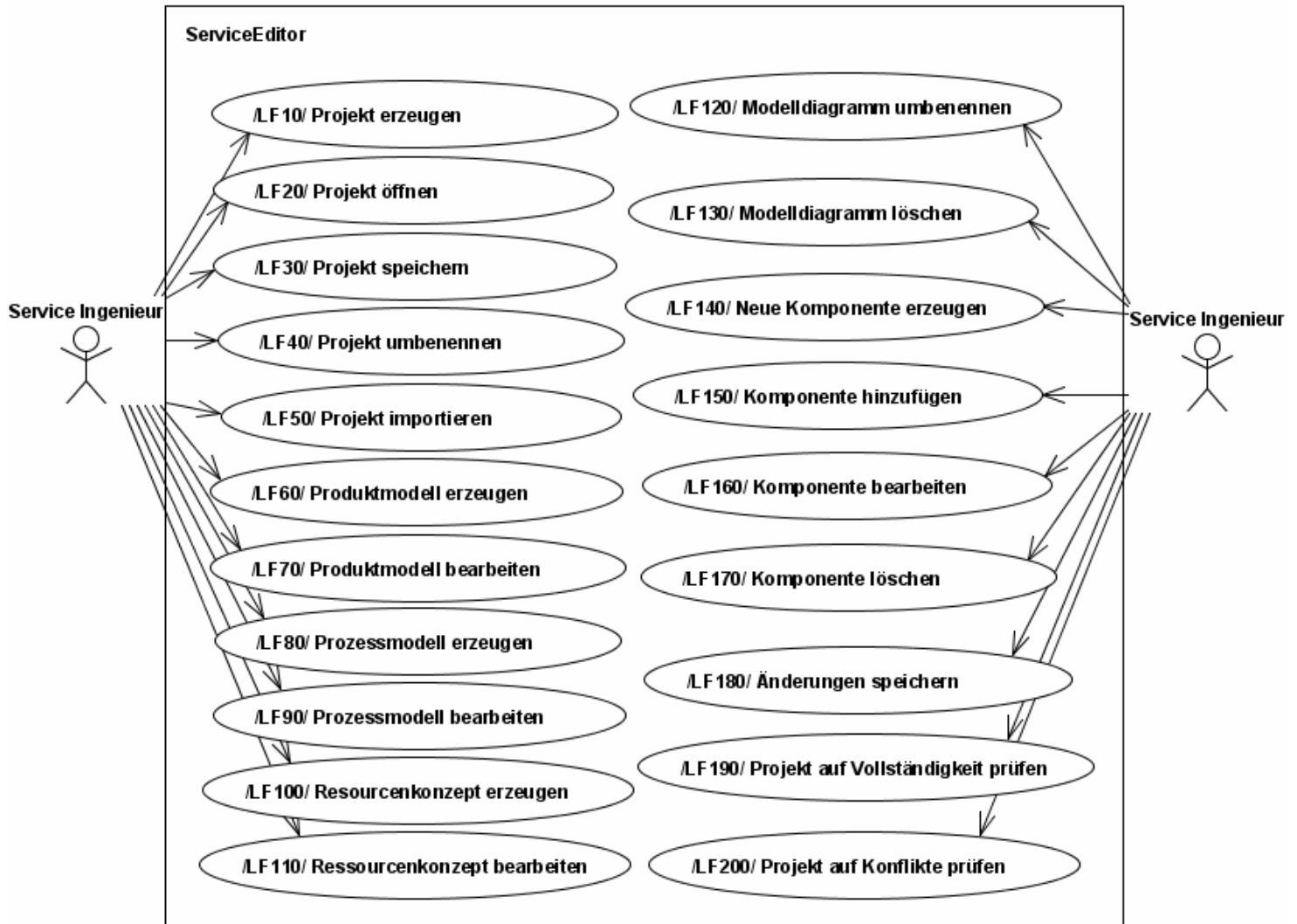


Diagramm: Usecase Diagramm
 Erstellt mit: Fujaba Tool Suite 4.3.1

4. Produktfunktionen

/LF10/	Geschäftsprozess: Akteure: Beschreibung:	Projekt erzeugen Service Ingenieur Nach dem Starten des Editors kann ein neues Projekt angelegt werden
/LF20/	Geschäftsprozess: Akteure: Beschreibung:	Projekt öffnen Service Ingenieur Nach dem Starten des Editors kann ein vorhandenes Projekt geöffnet werden
/LF30/	Geschäftsprozess: Akteure: Beschreibung:	Projekt speichern Service Ingenieur Nach dem Ändern des Projekts kann dieses gespeichert werden
/LF40/	Geschäftsprozess: Akteure: Beschreibung:	Projekt umbenennen Service Ingenieur Nach dem Starten des Editors kann ein vorhandenes und geöffnetes Projekt umbenannt werden
/LF50/	Geschäftsprozess: Akteure: Beschreibung:	Projekt importieren Service Ingenieur Nach dem Starten des Editors kann ein vorhandenes Projekt importiert werden
/LF60/	Geschäftsprozess: Akteure: Beschreibung:	Produktmodell erzeugen Service Ingenieur Nach dem Anlegen oder Öffnen des Projekts kann ein Produktmodell erzeugt werden
/LF70/	Geschäftsprozess: Akteure: Beschreibung:	Produktmodell bearbeiten Service Ingenieur Nach dem Öffnen des Projekts kann ein vorhandenes Produktmodell bearbeitet werden
/LF80/	Geschäftsprozess: Akteure: Beschreibung:	Prozessmodell erzeugen Service Ingenieur Nach dem Anlegen oder Öffnen des Projekts kann ein Prozessmodell erzeugt werden
/LF90/	Geschäftsprozess: Akteure: Beschreibung:	Prozessmodell bearbeiten Service Ingenieur Nach dem Öffnen des Projekts kann ein vorhandenes Prozessmodell bearbeitet werden
/LF100/	Geschäftsprozess: Akteure: Beschreibung:	Ressourcenkonzept erzeugen Service Ingenieur Nach dem Anlegen oder Öffnen des Projekts kann ein Ressourcenkonzept erzeugt werden
/LF110/	Geschäftsprozess: Akteure: Beschreibung:	Ressourcenkonzept bearbeiten Service Ingenieur Nach dem Öffnen des Projekts kann ein vorhandenes Ressourcenkonzept bearbeitet werden
/LF120/	Geschäftsprozess: Akteure: Beschreibung:	Modelldiagramm umbenennen Service Ingenieur Nach dem Öffnen des Projekts kann ein vorhandenes Modelldiagramm umbenannt werden

/LF130/	Geschäftsprozess: Akteure: Beschreibung:	Modelldiagramm löschen Service Ingenieur Nach dem Öffnen des Projekts kann ein vorhandenes Modelldiagramm gelöscht werden
/LF140/	Geschäftsprozess: Akteure: Beschreibung:	Neue Komponente erzeugen Service Ingenieur Nach dem Starten des Editors, Öffnen des Projektes und des entsprechenden Modells kann eine neue Komponente (gemäß dem Metamodell) erzeugt werden
/LF150/	Geschäftsprozess: Akteure: Beschreibung:	Komponente hinzufügen Service Ingenieur Nach dem Starten des Editors, Öffnen des Projektes und des entsprechenden Modells kann eine neue erzeugte Komponente dem Modell hinzugefügt werden
/LF160/	Geschäftsprozess: Akteure: Beschreibung:	Komponente bearbeiten Service Ingenieur Nach dem Auswählen einer Komponente (durch Anklicken) kann diese bearbeitet werden
/LF170/	Geschäftsprozess: Akteure: Beschreibung:	Komponente löschen Service Ingenieur Nach dem Auswählen einer Komponente (durch Anklicken) kann diese gelöscht werden
/LF180/	Geschäftsprozess: Akteure: Beschreibung:	Änderungen speichern Service Ingenieur Nach den Änderungen an einer Komponente, einem Diagramm oder dem Gesamtprojekt können diese gespeichert werden.
/LF190/	Geschäftsprozess: Akteure: Beschreibung:	Projekt auf Vollständigkeit prüfen Service Ingenieur Nach Fertigstellung des Projekts kann dieses auf Vollständigkeit geprüft werden. Dabei wird es nach den Regeln des Service Engineerings geprüft
/LF200/	Geschäftsprozess: Akteure: Beschreibung:	Projekt auf Konflikte prüfen Service Ingenieur Während und nach dem Fertigstellen des Projekts kann dieses auf Konflikte geprüft werden. Besondere Aufmerksamkeit gilt dabei der Ressourcenzuteilung.

5. Produktdaten

Da die Datenhaltung mit XML realisiert wird, hängt die maximale Anzahl der Daten von der Kapazität der Kundenfestplatte ab.

/LD10/ Projektdaten

6. Produkteleistungen

/LL10/ Das Laden der Projektdaten sollte 20 Sekunden nicht überschreiten.

/LL20/ Die Reaktionszeit auf die Benutzereingaben sollte 5 Sekunden nicht überschreiten

/LL30/ Das Aufbauen (grafisch) der Diagramme sollte 10 Sekunden nicht überschreiten.

7. Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
<u>Funktionalität</u>	x			
Angemessenheit	x			
Richtigkeit	x			
Interoperabilität				x
Ordnungsmäßigkeit	x			
Sicherheit	x			
<u>Zuverlässigkeit</u>		x		
Reife		x		
Fehlertoleranz		x		
Wiederherstellbarkeit		x		
<u>Benutzbarkeit</u>	x			
Verständlichkeit	x			
Erlernbarkeit	x			
Bedienbarkeit	x			
<u>Effizienz</u>			x	
Zeitverhalten			x	
Verbrauchsverhalten				x
<u>Änderbarkeit</u>			x	
Analysierbarkeit				x
Modifizierbarkeit			x	
Stabilität			x	
Prüfbarkeit			x	
<u>Übertragbarkeit</u>			x	
Anpassbarkeit			x	
Installierbarkeit			x	
Konformität			x	
Austauschbarkeit			x	

8. Benutzungsschnittstelle

- /B10/ Die Benutzeroberfläche sollte einfach und intuitiv benutzbar sein.
- /B20/ Die Oberfläche ist auf Mausbedienung auszulegen, es sollte aber auch möglich sein, das Programm auch ohne Maus zu bedienen.
- /B30/ Die Oberfläche muss durch aktuell gängige Browser darstellbar sein.
- /B40/ Die Oberfläche sollte in drei Hauptbereiche gegliedert sein: Projekt-, Diagramm- und Komponentenbereich. Des Weiteren sollte ein spezieller Bereich für Status- und Fehleranzeigen vorgesehen werden.

9. Nicht funktionale Anforderungen

- / -

10. Technische Produktumgebung

Das Produkt läuft auf einem Rechner mit grafischer Benutzeroberfläche.

10.1 Software

- Die Anwendung ist auf einer Java Plattform zu entwickeln.

10.2 Hardware

- Für die Anwendung steht ein PC als Webserver (Apache Tomcat Server 5.0) zur Verfügung.

10.3 Orgware

- Als Betriebssystem wird Windows XP eingesetzt.

10.4 Produktschnittstellen

- Es sind keine Produktschnittstellen abgesehen von der Weboberfläche vorgesehen.

11. Spezielle Anforderungen an die Entwicklungsumgebung

Es ist keine spezielle Entwicklungsumgebung erforderlich.

12. Gliederung in Teilprodukte

- / -

13. Ergänzungen

- Die Umsetzung des Marketingkonzeptes klammern wir bewusst aus, da es, unserer Meinung nach, den Rahmen des Projekts sprengen würde. Dieses Modul kann in späteren Projekten hinzugefügt werden.
- Es sollte möglich sein, in späteren Projekten, die Anwendung zu erweitern. Dies planen wir durch das Einsetzen von „Platzhaltern“ zu erleichtern. Die Funktionalität dieser Platzhalterklassen / -methoden kann dann problemlos hinzugefügt werden.
- Als Wunschkriterium wird die Möglichkeit angesehen, ein Metamodell laden zu können, um so die Abhängigkeit des Programms von einem statischen Metamodell zu beseitigen. Derzeit gehen wir noch von einem statischen Metamodell aus, behalten es uns aber vor, je nach Zeitaufwand auch das Laden eines beliebigen Metamodells zu ermöglichen. Das Erfüllen von Sollvorgaben steht aber an erster Stelle.