

Recherchenbericht

1. Begriffe

1.1 IT-basierte Dienstleistung

Dienstleistungen nennt man IT-basiert, wenn für eine effiziente Erbringung der Dienstleistung der Einsatz von Informationstechnik bei wertschöpfungsrelevanten Prozessschritten zwingend erforderlich ist. Das alleinige Vorhandensein von Computersystemen mit Betriebssystem und Ergänzungsprogrammen (z.B. Office-Programme, Browser, etc.) reicht nicht aus, um eine Dienstleistung als IT-basiert anzusehen.

Quelle: http://www.softwarefoerderung.de/projekte/servcase/folien_SERVCASE.pdf

Begründung: ein Begriff aus der Aufgabenstellung

1.2 Service Engineering

Service Engineering beschäftigt sich mit der systematischen Entwicklung und Gestaltung von Dienstleistungen unter Verwendung geeigneter ingenieurwissenschaftlicher Methoden, Vorgehensweisen und Werkzeuge.

Quelle: http://ais.informatik.unileipzig.de/studium/vorlesungen/2004_ss/se_it_dl/

Begründung: resultiert aus dem Begriff 1.1 IT-basierte Dienstleistungen

1.3 Produktmodelle

Ein Produktmodell beschreibt, was eine Dienstleistung leistet (nicht wie)

- Beschreibung der Leistung
- Festlegung der Stammdaten
- ggf. Definition von Modulen

Quelle: http://ais.informatik.unileipzig.de/studium/vorlesungen/2004_ss/se_it_dl/

Begründung: resultiert aus dem Begriff 1.2 Service Engineering

1.4 Prozessmodelle

Ein Prozessmodell beschreibt, wie die Ergebnisse einer Dienstleistung zustande kommen.

- Festlegung der Prozessschritte
- Definition von Schnittstellen

Quelle: http://ais.informatik.unileipzig.de/studium/vorlesungen/2004_ss/se_it_dl/

Begründung: resultiert aus dem Begriff 1.2 Service Engineering

1.5 Ressourcenkonzepte

Ressourcenkonzepte dienen der Planung des Ressourceneinsatzes, der für die spätere Erbringung der Dienstleistung erforderlich ist:

- Mitarbeiter
- Betriebsmittel
- IuK-Infrastruktur

Quelle: http://ais.informatik.unileipzig.de/studium/vorlesungen/2004_ss/se_it_dl/

Begründung: resultiert aus dem Begriff 1.2 Service Engineering

1.6 Computer Aided Engineering (CAE)

Der Begriff Rechnergestützte Entwicklung oder englisch Computer Aided Engineering fasst alle Möglichkeiten der Computerunterstützung von Arbeitsprozessen der Ingenieure zusammen.

Der Begriff ist ähnlich zu verstehen wie Computer Aided Design, nur ist beim CAE ebenfalls die Konzeption und Analyse oder Simulation inbegriffen.

Quelle: http://de.wikipedia.org/wiki/Computer_Aided_Engineering

Begründung: ein Begriff aus der Aufgabenstellung

1.7 ServCASE

ServCASE: Computer Aided Engineering für IT-basierte Dienstleistungen:

Das Ziel des Verbundvorhabens besteht in der Entwicklung und Umsetzung eines Referenzmodells für das Co-Design von Software und Services. Insbesondere umfasst das Referenzmodell Vorgehensweisen, Methoden und Werkzeuge für die integrierte Entwicklung von Software- und Dienstleistungsprodukten. Um einen hohen praktischen Nutzen zu erreichen, wird das Referenzmodell in Form einer toolgestützten Entwicklungsplattform umgesetzt.

Neben der übergeordneten Zielsetzung des Verbundvorhabens werden folgende Teilziele angestrebt:

- Entwicklung eines konfigurierbaren Vorgehensmodells unter Würdigung der bekannten Klassen von Vorgehensmodellen (Phasenmodelle, iterative Modelle, Prototyping, regenerative- und Komponentenmodelle),
- Entwicklung von Methoden und Werkzeugen des Service Engineering mit einem Fokus auf Produktmodelle, Prozessmodelle, Ressourcenmodelle und Kundeninteraktionsmodelle,
- Entwicklung eines Reifegradmodells (Maturity Model) für das Co-Design von Software und Services,
- Zusammenführung der Teilmodelle in ein umfassendes Referenzmodell und Bereitstellung einer integrierten Entwicklungsplattform, Demonstration von Machbarkeit und Anwendbarkeit in Betriebsprojekten und Transfer der Ergebnisse in die Praxis.

Quelle: http://www.softwarefoerderung.de/projekte/servcase/beitrag_SERVCASE.pdf

Begründung: ein Begriff aus der Aufgabenstellung

1.8 Serviceorientierte Architektur (Service Oriented Architecture)

Der Begriff Serviceorientierte Architektur (SOA) oder englisch Service Oriented Architecture, auch dienstorientierte Architektur, ist ein Managementkonzept und setzt erst in zweiter Linie ein Systemarchitektur-Konzept voraus:

- Das Managementkonzept strebt eine an den gewünschten Geschäftsprozessen ausgerichtete IT-Infrastruktur an, die schnell auf veränderte Anforderungen im Geschäftsumfeld reagieren kann.
- Das Systemarchitektur-Konzept sieht die Bereitstellung fachlicher Dienste und Funktionalitäten in Form von Services vor.

Ein Service sei in diesem Kontext als eine Funktionalität definiert, die über eine standardisierte Schnittstelle in Anspruch genommen werden kann. Ein Synonym also für die lange bekannte Software-Komponente.

Quelle: http://de.wikipedia.org/wiki/Service_Oriented_Architecture

Begründung: ein Begriff aus der Aufgabenstellung

1.9 Komponentenmodell

Komponentenmodelle sind konkrete Ausprägungen des Paradigmas der komponentenbasierten Entwicklung. Neben der genauen Form und den Eigenschaften einer Komponente muss es einen Interaction-Standard und einen Composition-Standard festlegen.

Ein Komponentenmodell kann außerdem Implementierungen verschiedener Hersteller besitzen.

Folgende Komponentenmodelle sind weit verbreitet:

- Enterprise Java Beans
- Distributed Component Object Model
- CORBA Component Model

Quelle: <http://de.wikipedia.org/wiki/Komponentenmodell>

Begründung: ein Begriff aus der Aufgabenstellung

1.10 Metamodell

Modelle, die beschreiben, wie Modelle gebaut werden, nennt man Metamodelle.

Metamodelle werden im Bereich des Software Engineering und des IT gestützten Managements (Strategiemanagement, Geschäftsprozessmanagement, IT-Architekturmanagement etc.) zur Beschreibung von Modellierungssprachen eingesetzt. Auch können metamodellbasiert Abbildungen zwischen einzelnen Modellen verschiedener Sprachen definiert werden. Beispielsweise kann ein Entity-Relationship-Modell in ein Klassenmodell transformiert werden. Der Standard der OMG zur metamodellbasierten Modell- zu Modell-Transformation ist QVT.

Quelle: <http://de.wikipedia.org/wiki/Metamodell>

Begründung: ein Begriff aus der Aufgabenstellung

1.11 ISO/ IEC 9126

ISO/ IEC 9126 stellt eins von vielen Modellen dar, um Softwarequalität sicherzustellen. Es bezieht sich ausschließlich auf die Produktqualität und nicht die Prozessqualität.

Folgende Qualitätsmerkmale werden aufgeführt:

Funktionalität, Angemessenheit, Richtigkeit, Interoperabilität, Sicherheit, Konformität, Zuverlässigkeit, Reife, Fehlertoleranz, Robustheit usw.

Quelle: http://de.wikipedia.org/wiki/ISO_9126

Begründung: ein Begriff aus der Aufgabenstellung

1.12 SPDL

- Service Product Description Language, basiert auf OWL
- Ziel: Sammeln alles Wissens, dass im Lebenszyklus von Dienstleistungsprodukten benötigt wird, maximal mögliche Integration mit Webservices
- Methoden: Entwickeln von domänenspezifischen Ontologien, rekursiv-modulare Modellierung von Service-Produkten
- Ergebnis: ein Dokument beschreibt Service-Produkte, Prozesse, Modelle, generell alle zum Service relevanten Informationen

Quelle: http://ais.informatik.unileipzig.de/studium/vorlesungen/2004_ss/se_it_dl/

Begründung: resultiert aus dem Begriff 1.2 Service Engineering

2. Konzepte

2.1 Apache Struts

Struts ist ein Open-Source-Framework für die Präsentationsschicht von Java-Webanwendungen. Als eines der bekanntesten Jakarta-Projekte ist es im Jahr 2005 zu einem „Apache Toplevel Project“ avanciert. Struts beschleunigt die Entwicklung von Webanwendungen wesentlich, indem es HTTP-Anfragen in einem standardisierten Prozess verarbeitet. Dabei bedient es sich standardisierter Technologien wie JavaServlets, Java Beans, Resource Bundles und XML sowie verschiedener Jakarta-Commons-Pakete. Für den Entwickler bedeutet das, dass viele applikationsrelevante Funktionen bereits implementiert und einsatzbereit sind. Struts wird bereits in sehr vielen Webanwendungen eingesetzt und gemeinhin als solides Framework angesehen. Struts wurde von Craig McClanahan [1] im Jahr 2000 entwickelt. Seitdem arbeitet eine ständig wachsende Entwicklergemeinde an der Verbesserung des Frameworks. Aktuelle Version ist 1.3. Es gibt auch eine alternative Version des Struts-Frameworks, welche auf Basis der Skriptsprache PHP 5 implementiert wurde: struts4php.

Quelle: <http://de.wikipedia.org/wiki/Struts>

2.2 Apache Tomcat

Apache Tomcat stellt eine Umgebung zur Ausführung von Java-Code auf Webservern bereit, die im Rahmen des Jakarta-Projekts der Apache Software Foundation entwickelt wird. Es handelt sich um einen in Java geschriebenen Servlet Container, der mithilfe des JSP-Compilers Jasper auch JavaServer Pages in Servlets übersetzen und ausführen kann. Dazu kommt ein kompletter HTTP-Server.

Der HTTP-Server des Tomcat wird vor allem zur Entwicklung eingesetzt, während in Produktion zumeist ein Apache Web-Server zum Einsatz kommt. Dazu wird in Apache ein Plugin eingebunden, das nur Requests für dynamische Inhalte an Tomcat weiterleitet. Das Plugin spricht Tomcat dann normalerweise über das Apache JServ Protocol an.

Quelle: http://www.computerbase.de/lexikon/Apache_Tomcat

2.3 Extensible Markup Language (XML)

Die Extensible Markup Language (engl. für „erweiterbare Auszeichnungssprache“), abgekürzt XML, ist ein Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur, der vom World Wide Web Consortium (W3C) definiert wird. XML definiert dabei die Regeln für den Aufbau solcher Dokumente. Für einen konkreten Anwendungsfall ("XML-Anwendung") müssen die Details der jeweiligen Dokumente spezifiziert werden. Dies betrifft insbesondere die Festlegung der Strukturelemente und ihre Anordnung innerhalb des Dokumentenbaums. XML ist damit ein Standard zur Definition von beliebigen in ihrer Grundstruktur jedoch stark verwandten Auszeichnungssprachen. Eine Sprache zur Definition anderer Sprachen nennt man Metasprache. XML ist eine vereinfachte Teilmenge von SGML. Die Namen der Strukturelemente (XML-Elemente) für eine XML-Anwendung lassen sich frei wählen. Ein XML-Element kann ganz unterschiedliche Daten enthalten und beschreiben, als prominentestes Beispiel Text, aber auch Grafiken oder abstraktes Wissen. Ein Grundgedanke hinter XML ist es, Daten und ihre Repräsentation zu trennen, also beispielsweise Wetterdaten

einmal als Tabelle und einmal als Grafik auszugeben, aber für beide Anwendungen die gleiche Datenbasis im XML-Format zu nutzen.

Quelle: http://de.wikipedia.org/wiki/Extensible_Markup_Language

2.4 Java

Java ist eine objektorientierte, plattformunabhängige Programmiersprache und als solche ein eingetragenes Warenzeichen der Firma Sun Microsystems. Üblicherweise benötigen Java-Programme zur Ausführung eine spezielle Umgebung (Java Virtual Machine). Der Vorteil ist, dass nur diese Umgebung an verschiedene Computer und Betriebssysteme angepasst werden muss. Sobald dies geschehen ist laufen auf der Plattform alle Java-Programme ohne Anpassungsarbeiten.

Quelle: [http://de.wikipedia.org/wiki/Java_\(Programmiersprache\)](http://de.wikipedia.org/wiki/Java_(Programmiersprache))

3. Beschreibung der Rahmenapplikation

Verbale Beschreibung der Leistungsmerkmale und der Anwendungsfälle

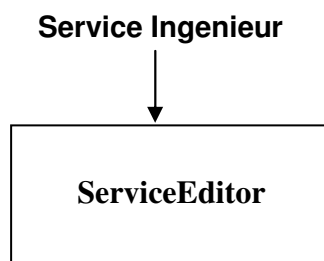
1. Zielbestimmung

Der Kunde soll mit Hilfe des Editors in die Lage versetzt werden, auf der Basis eines Metamodells einzelne Komponenten und deren Komposition zu modellieren. Dabei soll das Zusammenspiel zwischen den funktionalen und nichtfunktionalen Aspekten berücksichtigt werden. Der Kunde soll mit dem Editor die komplette Dienstleistung entwerfen und deren einzelnen Komponenten & Modelle erstellen und bearbeiten können. Es soll nach den Prinzipien des Service Engineering vorgegangen werden (s. Glossar Punkt 1.2 bis 1.5).

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 verwendbar sein.

3. Produktübersicht



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 Öffnen des Projekts kann ein vorhandenes Modelldiagramm umbenannt werden

/LF130/	Geschäftsprozess: Akteure: Beschreibung:	Modelldiagramm löschen Service Ingenieur Nach 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
/LF140/	Geschäftsprozess: Akteure: Beschreibung:	Komponente bearbeiten Service Ingenieur Nach dem Auswählen einer Komponente (durch Anklicken) kann diese bearbeitet werden
/LF150/	Geschäftsprozess: Akteure: Beschreibung:	Komponente löschen Service Ingenieur Nach dem Auswählen einer Komponente (durch Anklicken) kann diese gelöscht werden
/LF160/	Geschäftsprozess: Akteure: Beschreibung:	Projekt auf Vollständigkeit prüfen Service Ingenieur Nach Fertigstellung des Projekts kann dieses auf Vollständigkeit geprüft werden. Dabei wird es nach den Regel des Service Engineering geprüft
/LF170/	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

Datenhaltung wird mit XML realisiert

Die Angaben zum Umfang der Daten werden noch präzisiert

6. Produktleistungen

Wird noch präzisiert

7. Qualitätsanforderungen

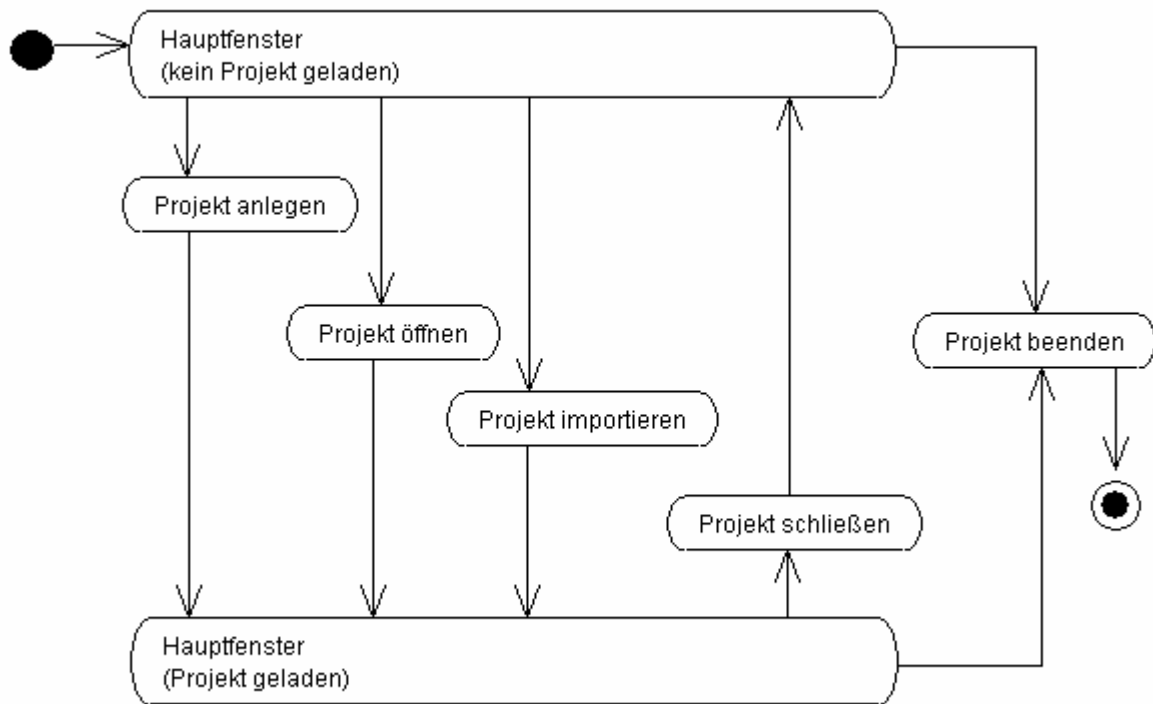
Wird noch präzisiert

8. Ergänzungen

Wird noch präzisiert

Darstellung äußerlich sichtbarer Aspekte der inneren Logik des ServiceEditors mittels Aktivitätsdiagramme

1. Hauptfenster



2. Diagrammfenster

