

Sommerprüfung 2019

Ausbildungsberuf

Fachinformatiker/-in Anwendungsentwicklung

Prüfungsbezirk

SIHK FIA 101 (AP T2V1)

Herr Roman Berger

Identnummer: 545664

E-Mail: r.berger.1992@web.de, Telefon: +49 2331 402208

Ausbildungsbetrieb: stratEDI GmbH

Projektbetreuer: Herr Thorsten Georg

E-Mail: thorsten.georg@stratedi.de, Telefon: +49 2332 66600-0

Thema der Projektarbeit

Entwicklung einer Datenbank-gestützten Clientanwendung für Mitarbeiter der stratEDI GmbH zur Verwaltung und automatisierten Einbindung eingegangener Kunden-Testdaten.

1 Thema der Projektarbeit

Entwicklung einer Datenbank-gestützten Clientanwendung für Mitarbeiter der stratEDI GmbH zur Verwaltung und automatisierten Einbindung eingegangener Kunden-Testdaten.

2 Geplanter Bearbeitungszeitraum

Beginn: 25.03.2019

Ende: 26.04.2019

3 Projektbeschreibung

Aktueller Stand

Als Kommunikationszwischenstelle für den Austausch von elektronischen Geschäftsnachrichten zwischen Händlern und Lieferanten erhält die stratEDI GmbH täglich Testdaten von Kunden, welche in den Testbetrieb zur weiteren Verarbeitung gestellt werden müssen.

Dazu werden erstmals die gesendeten Testdateien von einem zugestellten Mitarbeiter aus dem Email-Anhang bzw. dem Speicherordner für Emails entnommen und ggf. mittels eines externen Programms auf ihre syntaktische Richtigkeit geprüft.

Ein solches Programm verwendet je ein speziell auf das Format der Kunden-Testdatei ausgerichtetes Mapping, welches für den Abgleich mit der Testdatei verwendet wird.

Danach erfolgt die semantische Prüfung, bei der ein Mitarbeiter die Testdatei öffnet und die Sinnigkeit der angegebenen Daten überprüft.

Kam es bei einer der beiden Phasen zu Fehlern, dann versendet der Mitarbeiter an den Kunden eine Email mit den erfassten Fehlern und der Aufforderung einer korrigierten Testdatei.

Wurden keine Fehler entdeckt, dann erfolgt die Einbindung der Testdatei, welche zuerst in einen entsprechenden Ordner verschoben bzw. kopiert wird.

Darauf folgt die manuelle Bearbeitung einer der vorgesehen Textdateien, welche um den Namen der Testdatei und weiterer Meta-Daten, abhängig vom vorhandenen Format der Testdatei, an entsprechender Stelle erweitert wird.

Diese Textdateien enthalten alle bereits eingebundenen Testdaten und werden im späteren Verlauf für weitere Programme benötigt, durch die der Testbetrieb weiter fortgeführt wird.

Der Mitarbeiter navigiert dafür zuerst in das entsprechende Verzeichnis mit dem erforderlichen Programm und öffnet dieses Programm mittels eines Editors (WordPad, Notepad, etc.)

Als nächstes scrollt der Mitarbeiter in der jeweiligen Textdatei bis an die entsprechende Stelle für die einzubindenden Testdateien des Kunden und muss dann händisch eine Zeile mit entsprechenden Parametern für die Testdatei einfügen.

Danach muss noch ein Vermerk am Anfang der Datei in Form eines Mitarbeiter-Kürzels hinterlegt werden, mit welcher der zuletzt zuständige Mitarbeiter festgehalten wird.

Weiterhin muss im Intranet des Unternehmens noch ein Eintrag im „Workflow“ hinterlassen werden welcher besagt, dass eine Testdatei in den Testbetrieb gestellt wurde.

Zum Schluss wird nach einem reibungslosen Ablauf dem Kunden eine Nachricht geschrieben, welche die Einbindung seiner Testdatei bestätigt.

Mängel

Wie zuvor aufgezeigt erfolgen bis zur eigentlichen Einbindung der Testdatei viele manuelle Zwischenschritte und händische Bearbeitungen von Quelltexten, welche ebenfalls erst unter einer größeren Menge von weiteren Quelltexten aus diversen Unterverzeichnissen rausgesucht werden müssen.

Händische Bearbeitungen erweisen sich gerade bei einer großen Zahl zu bearbeitender Testdaten als sehr redundant und zeitaufwändig, Zeit, welche in der Kernaufgabe, der Überprüfung und Sicherstellung von validen Testdateien, sinnvoller genutzt werden könnte. Auch können durch die händische Bearbeitung und durch etwaige Konfusion aufgrund der recht unübersichtlichen Verzeichnis- und Datenstruktur Fehler entstehen, womit auch ein Sicherheitsrisiko über die Funktionsrichtigkeit in der Einbindung besteht und gegebenenfalls zeitliche Blockaden entstehen.

Soll-Stand

Im Anbetracht der vorherig aufgezeigten Mängel, soll die Entwicklung eines Projekts gestartet werden, welches die aktuellen Arbeitsschritte möglichst kompakt zusammenfasst und redundante Arbeitsschritte automatisiert.

Statt dass die Bearbeitung der Testdaten von einem Mitarbeiter über mehrere Instanzen ausgeführt wird, soll eine zentralisierte Anwendung die Verknüpfungen zu den Instanzen aufbauen und verwalten.

Besagte Instanzen, z.B. die Mapping-Programme werden dann über das Projekt automatisch ermittelt und angesprochen.

Im Fall der Syntax-Validierung wird die Kunden-Testdatei dem Programm übermittelt und dann mittels einer Test-Konvertierung in das InHouse-Format die Syntax validiert.

Von dem Mitarbeiter wird schlussendlich nur die Betätigung einer Schaltfläche für die Ausführung jener komplexer Arbeitsschritte gefordert.

Zur Realisierung der oben genannten Prozedur wird der Auszubildende ein Datenbankmodell konzeptionieren, mit welchem eine Verknüpfung zwischen dem Format einer Kunden-Testdatei und dem entsprechendem Mapping-Programm hergestellt werden kann.

Auch wird der Auszubildende entsprechende Diagramme und Programmläufe mittels UML-Diagrammen, etc. konzeptionieren, die einen genauen Überblick über die Funktionalitäten und den Implementierungsbedarf des Projekts verschaffen.

Jene Anwendung soll über eine leicht zu bedienende grafische Benutzeroberfläche verfügen und die einzelnen Bearbeitungsschritte leicht und verständlich visualisieren.

Das wiederholte Zusammensuchen der verschiedenen Komponenten für die Validierung und Testaufnahme von Testdaten soll entfallen, da die Anwendung alle benötigten Komponenten zusammenstellt und den Mitarbeitern stattdessen klare, strukturierte Schritt-für-Schritt-Bearbeitung in einer einzigen Domäne zur Verfügung stellt.

Daher wird auch zu der Benutzeroberfläche vom Auszubildenden ein Konzept angefertigt, in welchem die Benutzeroberfläche unter Einhaltung ergonomischer Softwarerichtlinien ausgestaltet wird. Das wiederum wird auch vom Azubi in der Implementierungsphase realisiert.

Weiterhin soll das Eintragen der Testdatei in die entsprechende Quelltextdatei ebenfalls automatisiert, und mittels eingespeicherter Parametervorgaben für verschiedene Formate von Kunden-Testdateien realisiert werden.

Änderungsvermerke in der Quelltextdatei für die Testdaten-Einbindung und für den Workflow im Intranet sollen automatisiert aufgezeichnet werden.

Durch richtige Ausführung durch den Mitarbeiter erweitert sich so die entsprechende Quelltextdatei automatisch um die neuen Einträge.

Für diesen Schritt muss auch wieder ein Datenbank-Konzept vom Auszubildenden angefertigt werden, welches die benötigten Parameter zu einem Format je Kunden festhält und entsprechende Datensätze für die erforderlichen Verwendungszwecke aufbereitet.

Auch muss ein technisches Konzept und ggf. eine Ausarbeitung des Datenmodells für die Quelltextdateien angefertigt werden, welches die automatisierte Einbindung von Kunden-Testdaten ermöglicht.

Ziel des Projekts ist es am Ende für die Mitarbeiter eine schnellere und gezieltere Bearbeitung von Testdaten zu ermöglichen, indem redundante Bearbeitungsschritte minimiert bis bestenfalls umgangen werden und den Fokus einzig auf die syntaktische und semantische Überprüfung der aktuellen Testdatei zu begrenzen.

Die Anwendung soll als begleitendes Werkzeug den ehemals langwierigen und teils unorganisierten Arbeitsablauf ersetzen und dem Mitarbeiter eine qualitativ bessere und produktivere Arbeitsweise gewähren.

Infolgedessen werden schnellere Feedbacks für Kunden und Zeiteinsparungen erzielt, welche geringere Kosten mit sich ziehen.

Dem Auszubildenden obliegt es das Projekt mit dem neuen Konzept und die alte Arbeitsweise in wirtschaftlicher und technischer Weise gegeneinander abzuwägen, was über die Aufstellung eines Kostenplans und letztlich über die Absprache mit dem Mitarbeiter-Team entschieden wird. Sollte das neue Konzept sich bewähren, übernimmt der Auszubildende die alleinige Implementierung des Projekts und stellt die Weichen für die Inbetriebnahme nach der abgeschlossenen Ausarbeitung bereit.

Anforderungen

Es ergeben sich folgende Anforderungen an die zu entwickelnde Anwendung:

- Die Anwendung muss leicht und flexibel ausführbar sein
- Leichte Bedienbarkeit für den Nutzer
- Alle Arbeitsschritte müssen abgedeckt und visuell repräsentiert werden
- Alle Arbeitsschritte müssen zurückverfolgbar sein
- Die Anwendung muss für weitere externe Programme erweiterbar sein
- automatisierte Einträge müssen innerbetrieblichen Konventionen folgen
- keine Einschränkungen für vorherige Arbeitsweisen sollen entstehen

4 Projektumfeld

Arbeitsplatz

Die Entwicklung der Anwendung wird als firmeninternes Projekt im Hause der stratEDI GmbH mit dem Sitz in Gevelsberg durchgeführt.

Das Unternehmen wurde im Jahr 1994 gegründet und hat sich auf die Kundenanbindung mittels EDI und der Realisierung von cloudbasierten Clearing-Serviceleistungen spezialisiert.

Stets seine eigenen Standards zu übertreffen, baut und modernisiert die stratEDI GmbH seine Service-Leistungen laufend weiter aus und liefert seinen Kunden maßgeschneiderte Lösungen für alle Anforderungen des EDIs.

Verwendete Tools

Entwickelt wird die Anwendung mit dem Framework JavaFX unter der Programmiersprache Java.

Als Entwicklungsumgebung für Java dient Apache Netbeans 10.

Für das Management von Libraries und das Erstellen von Releases wird das Build-Management-Tool Maven von Apache verwendet.

Zum Speichern der Daten wird die auch im üblichen Normalbetrieb verwendete MariaDB genutzt.

Für die Erstellung und Verwaltung von Datenbanken wird HeidiSQL verwendet.

Zur Ausführung der Anwendung wird eine kompakte, eigenständige Laufzeit für das Windows-Betriebssystem mittels jLink erstellt.

Für die Validierung und Konvertierung von Testdateien werden die stetig weiterentwickelten, unternehmensinternen Mapping-Programme verwendet, welche einen wesentlichen Bestandteil des Programms ausmachen.

Entwicklungsprozess

Die Projektdurchführung, darunter die Konzeptionierung und Implementierung, wird vollständig vom Auszubildenden übernommen und als ein in sich abgeschlossenes Projekt bearbeitet.

Dabei soll das Projekt agil entwickelt werden, d.h. es werden in kurzen Releasezyklen Feedback und Änderungswünsche von entsprechenden Mitarbeitern einbezogen, wodurch Zeit bei der Entwicklung eingespart wird.

Im gleichen Zeitraum wird den Mitarbeitern, welche auch die zukünftigen Nutzer des Produkts sein werden, das Wissen für die Benutzung der Anwendung vermittelt, wodurch etwaige Schulungen für die spätere Nutzung entfallen.

5 Projektphasen mit Zeitplanung

Folgende Aufgabenphasen ergeben sich für die Entwicklung des Projekts:

Analyse:	6 Std.
- Anforderungen und Absprache mit Mitarbeitern	2 Std.
- Erstellung der Anforderungsliste	2 Std.
- Wirtschaftlichkeitsprüfung/Amortisationsrechnung	2 Std.
Planung/Entwurf:	9 Std.
- Netzplan erstellen	1 Std.
- Erstellung von Klassendiagrammen	1 Std.
- Erstellung des Aktivitätsdiagramm	1 Std.
- Erstellung des Use-Case-Diagramm	1 Std.
- Erstellung von Datenbankmodellen	2 Std.
- Planung des GUI-Layouts	1 Std.
Implementierung:	36 Std.
- Datenbank-Verknüpfungen	3 Std.
- Datenbank für Mitarbeiter	1 Std.
- Datenbank für Meta-Daten von Testdaten	2 Std.
- Ausarbeitung der GUI	6 Std.
- Hauptfenster	1 Std.
- Einstellungsfenster	2 Std.
- Bearbeitungsfenster	3 Std.
- Funktionalitäten implementieren	22 Std.
- Mitarbeiter-Profil einstellen	1 Std.
- Einbezug und Anzeige zugewiesener Testdaten	2 Std.
- Schnittstellen für externe Programme und Mailsoftware	2 Std.
- Syntaktischen Workflow implementieren	3 Std.
- Semantischen Workflow implementieren	4 Std.
- Prozesse der Testdaten-Einbindung	10 Std.
— Eingliederung in Quelldatei	4 Std.
— Änderungsvermerke in Intranet und Dateien	2 Std.
— Kundenfeedback	2 Std.
- Fehlerkorrektur	5 Std.
- Integration in den Geschäftsbetrieb	1 Std.
Dokumentation:	18 Std.
- Erstellung der Projektdokumentation	12 Std.
- Erstellung der Kundendokumentation	6 Std.
Gesamt:	70 Std.

6 Dokumentation zur Projektarbeit

Für die Dokumentation des Projekts werden die folgenden Unterlagen im Verlauf der Entwicklungsarbeiten erstellt:

- UML-Diagramme
 - Klassendiagramme
 - Aktivitäts-Diagramm
 - Use-Case-Diagramm
- Datenbankmodelle

- Layoutentwurf
- Nutzwertanalyse
- Kostenplan
- Amortisationsrechnung
- Programmablaufplan
- Projektdokumentation
- Benutzerhandbuch

7 Anlagen

siehe Anlage 1

8 Präsentationsmittel

Folgende Hilfsmittel werden für die Präsentation des Projekts mitgebracht:

- Beamer
- Notebook

9 Hinweis!

Ich bestätige, dass der Projektantrag dem Ausbildungsbetrieb vorgelegt und vom Ausbildenden genehmigt wurde. Der Projektantrag enthält keine Betriebsgeheimnisse. Soweit diese für die Antragstellung notwendig sind, wurden nach Rücksprache mit dem Ausbildenden die entsprechenden Stellen unkenntlich gemacht.

Mit dem Absenden des Projektantrages bestätige ich weiterhin, dass der Antrag eigenständig von mir angefertigt wurde. Ferner sichere ich zu, dass im Projektantrag personenbezogene Daten (d. h. Daten über die eine Person identifizierbar oder bestimmbar ist) nur verwendet werden, wenn die betroffene Person hierin eingewilligt hat.

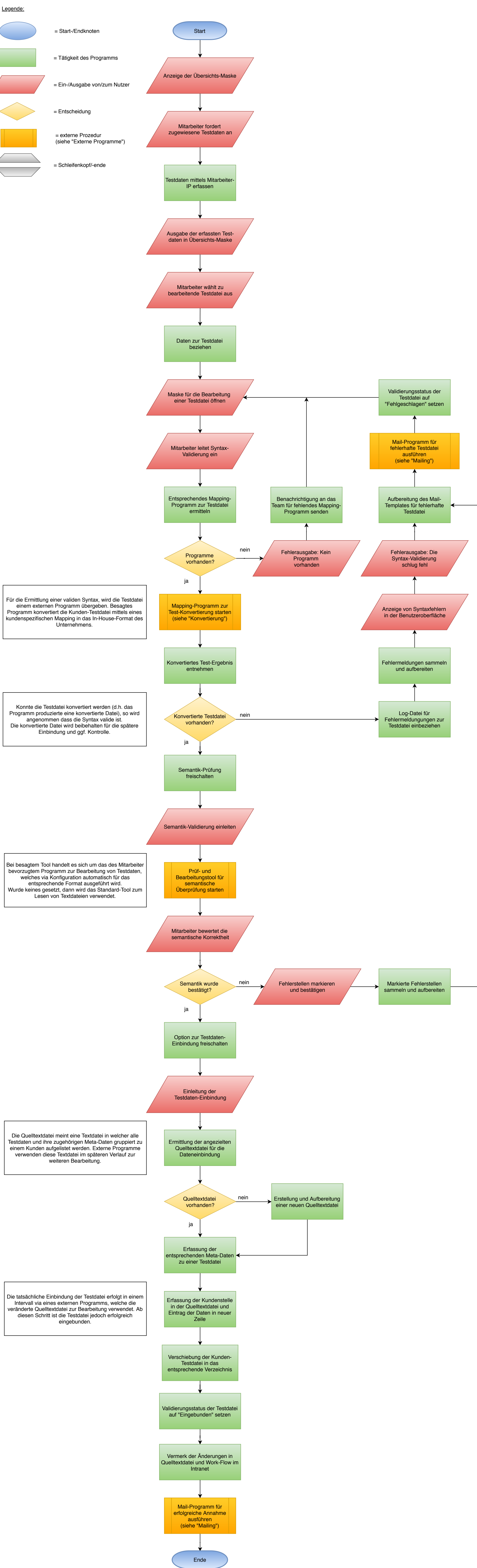
Bei meiner ersten Anmeldung im Online-Portal wurde ich darauf hingewiesen, dass meine Arbeit bei Täuschungshandlungen bzw. Ordnungsverstößen mit „null“ Punkten bewertet werden kann. Ich bin weiter darüber aufgeklärt worden, dass dies auch dann gilt, wenn festgestellt wird, dass meine Arbeit im Ganzen oder zu Teilen mit der eines anderen Prüfungsteilnehmers übereinstimmt. Es ist mir bewusst, dass Kontrollen durchgeführt werden.

Anlage 1

PAP: automatisierung der manuellen Arbeitsschritte zur Annahme und Einbindung von Kunden-Testdaten in den Testbetrieb

Legende:

-  = Start-/Endknoten
-  = Tätigkeit des Programms
-  = Ein-/Ausgabe von/zum Nutzer
-  = Entscheidung
-  = externe Prozedur (siehe "Externe Programme")
-  = Schleifenkopf/-ende



Externe Programme

