



Abschlussprüfung Sommer 2022

Fachinformatiker für Anwendungsentwicklung  
Dokumentation zur betrieblichen Projektarbeit

## Scraper Dashboard

**Verbesserung der Datenqualität im CRM-System Salesforce unter  
Nutzung eines auf Python basierenden REST Client.**

Abgabetermin: Neuhausen auf den Fildern, den 07.06.2022

**Prüfungsbewerber:**

Florin Gusa  
Gutenbergstraße 18  
75428 Illingen



**Ausbildungsbetrieb:**

Fanuc Europe GmbH  
Bernhäuser Str. 36  
73765 Neuhausen auf den Fildern

## Inhaltsverzeichnis

## Abbildungsverzeichnis

## Tabellenverzeichnis

# 1 Einleitung

## 1.1 Projektumfeld

- Die folgende Projektdokumentation schildert den Ablauf des IHK-Abschlussprojektes, welches der Autor im Rahmen seiner Ausbildung zum Fachinformatiker mit Fachrichtung Anwendungsentwicklung durchgeführt hat. Ausbildungsbetrieb ist die Fanuc Europe GmbH.

## 1.2 Projektziel

- Im CRM-System der Fanuc Europe befinden sich über 220.000 Kundenkonten, die Daten wie Adresse, Telefonnummer und Website des Kunden enthalten. Zur Automatisierung der Verwaltung dieser Kunden wird die Salesforce CRM-Plattform verwendet. Da es unmöglich ist, Änderungen an den Kundendaten manuell zu verfolgen, ist die Erstellung eines Web-Scraping-Systems geplant. Die Anwendung liest die Kontaktseiten der Kunden automatisiert und speichert sie in einem gemeinsamen Puffer. In diesem Puffer werden dann Tests durchgeführt, die die Daten mit den entsprechenden Daten in Salesforce abgleichen. Diese Aufgabe wurde vom Salesforce-Administrationsteam zugewiesen, dessen Ziel es ist, die Datenqualität innerhalb des CRM-Systems zu erhöhen. Für die Messung dieses Faktors wurde eine aktuelle Schätzung für falsch eingegebene Daten von etwa 20% ermittelt. Dies ist für das Marketing und die Kundenbeziehungen von entscheidender Bedeutung, da eines der Hauptziele des Unternehmens die Steigerung der Kundenzufriedenheit ist. Erwartet ist, dass mit dieser Anwendung 44000 Testergebnisse erstellt werden können, die über eine moderne und intuitive Schnittstelle angezeigt werden, um Salesforce-Administratoren einen klaren Überblick über die Daten zu geben.

## 1.3 Projektbegründung

- Das Unternehmen verfügt oft über Informationen, die in der gesamten Organisation verstreut sind, d.h. teilweise redundant in getrennten Systemen, aber häufig unvollständig. Das hat zur Folge, dass mehrere Unternehmensbereiche scheinbar nur mit den gleichen Daten arbeiten. Die kritische 360-Grad-Perspektive von Kunden oder Geschäftspartnern fehlt somit. Darüber hinaus besteht ein erhebliches Risiko, dass unzureichende Daten als Grundlage für strategische Geschäftsentscheidungen verwendet werden, die dann zur Festlegung von Unternehmenszielen und zur Modellierung von Geschäftsprozessen herangezogen werden. Folglich sind qualitativ hochwertige Daten erforderlich, damit das Unternehmen auf dem Markt erfolgreich sein kann. Der Hauptzweck besteht darin, die manuelle Arbeit zu verringern, die häufig dazu führt, dass ungenaue Daten im System gespeichert werden. Dies führt zu Problemen bei der Datenqualität im Unternehmen. Stattdessen kann ein Administrator einmal pro Woche einen automatisierten Vorgang ausführen, um über Änderungen an den Kundendaten informiert zu werden, was die

Produktivität erhöht und den Zeitaufwand für die Suche nach den relevanten Informationen verringert.

## 1.4 Projektschnittstellen

- Das Hauptaugenmerk liegt auf der Bestimmung von Softwarekomponenten, die sich am effektivsten in die Betriebs- und Systemumgebung des Unternehmens integrieren lassen. In jedem Fall müssen bei der Auswahl dieser Softwarekomponenten die aktuellen CRM-, ERP- oder sonstigen datenhaltenden Systeme berücksichtigt werden. Die aktuellen Daten werden über WSDL-Methoden von Salesforce in eine C#-Anwendung transportiert, die WPF für die Präsentationsschicht nutzt und einem definierten MVVM-Design folgt. Damit wird eine logische Schnittstelle geschaffen, über die alle Subsysteme agieren können. Ziel ist es, die Datenverarbeitung und Web Scraping Services so einfach wie möglich zu gestalten.

## 1.5 Projektabgrenzung

- Eine der grössten Herausforderungen bei der Erstellung dieses Projekts ist die Definition dessen, was geschehen soll. Daher ist es wichtig, den vom CRM-Entwicklungsteam benötigten Output klar zu definieren. Es wurde klargestellt, dass ein System, das Daten innerhalb der CRM-Plattform automatisch überschreibt, nicht erwünscht ist, sondern den Benutzer mit Hilfe von logisch aufgebauten Berichten benachrichtigt.

# 2 Projektplanung

In der Projektplanung soll die notwendige Zeit und die benötigten Ressourcen sowie ein Ablauf der Durchführung des Projektes geplant werden.

## 2.1 Projektphasen

- Für die Umsetzung des Projekts standen dem Autor 70 Stunden zur Verfügung. Diese wurden vor dem Start des Projektes auf verschiedene Phasen verteilt, die während der Softwareentwicklung durchlaufen werden. Eine grobe Zeitplanung sowie die Hauptphasen können der Tabelle 1 entnommen werden: Grobzeitplanung entnommen werden. Dazu lassen sich die einzelnen Hauptphasen noch in kleinere Unterphasen untergliedern.

Projektphase	Geplante Zeit
Startphase	6 h
Projektplanung	14 h
Durchführungsphase	38 h
Erstellen der Dokumentation	12 h
<b>Gesamt</b>	<b>70 h</b>

Tabelle 1: ProjektplanKurz

## 2.2 Abweichungen vom Projektantrag

- Im Projektantrag wurde festgelegt, dass bestimmte Bibliotheken mit Python für das Webscraping-System zusammen mit einem einfachen Codebeispiel verwendet werden. Dies wurde jedoch geändert, da es mehrere dynamische Websites gibt, was bedeutet, dass der Benutzer zuerst mit der Website interagieren muss, um auf sie zuzugreifen. Es gibt auch Fälle, in denen die Elemente für die Webseiten auf der Serverseite verschleiert und unsichtbar sind, was es für den Webscraper schwieriger macht, sie zu lesen und fehleranfällig ist. Die Menge der verwertbaren Stammdaten hat sich aufgrund der begrenzten Komplexität des Webscrapers reduziert. Dies ist auf Probleme bei der manuellen Dateneingabe sowie auf http/https-Probleme zurückzuführen. Stattdessen, werden nur korrekt formatierte Websites (auch nur solche, die das HTTPS verwenden) von der Anwendung eingelesen.

## 2.3 Ressourcenplanung

- In der Übersicht, die in Anhang A.2: Verwendete Ressourcen auf S. ii zu finden ist, sind alle für das Projekt verwendeten Ressourcen aufgeführt. Dazu gehören Hard- und Software-Ressourcen sowie Personal. Bei der Auswahl der verwendeten Software wurde darauf geachtet, dass diese kostenlos (z.B. als Open Source) zur Verfügung steht oder das Unternehmen bereits Lizenzen dafür besitzt. Dadurch konnten die Projektkosten so gering wie möglich gehalten werden.

## 2.4 Entwicklungsprozess

- Bevor das Projekt umgesetzt wurde, musste ein bestimmter geeigneter Entwicklungsprozess gewählt werden. Dieser definiert die Prozedur, nach der die Umsetzung erfolgen soll. Im Verlauf des Projekts entschied sich der Autor für eine agile Methodik, das sogenannte Inkrementelle Vorgehensmodell. Bei der agilen Softwareentwicklung geht es darum, möglichst schnell auf sich ändernde Anforderungen reagieren zu können.

Was ein inkrementelles Vorgehensmodell auszeichnet, ist, dass das zu entwickelnde System nicht im Voraus in allen Details genau geplant und dann in einem einzigen langen Durchgang entwickelt wird. Stattdessen stehen Teile der Software bei neuen Erkenntnissen und Entdeckungen immer

wieder neu im Mittelpunkt. Die Entwicklung findet in kurzen Zeitspannen statt, nach denen jeweils ein neues Leistungsmerkmal erstellt wird, das dem Kunden gezeigt werden kann. Sollte der Kunde einen Anpassungswunsch haben oder eine neue Erkenntnis über die Identifikation von Anomalien gemacht werden, kann darauf bei der nächsten Iteration schnell reagiert werden.

## 3 Analysephase

### 3.1 Ist-Analyse

- Aufgrund der großen Anzahl von Kundenstamm Setzen ist eine manuelle Prüfung der Daten nicht möglich. Aufgrund vorgangenen Analysen geht man von Fehlerhaften Datensätzen im Umfang von 20%. Es bestehen Zweifel, wenn zwei Adressen vorhanden sind, da nicht klar ist, auf welche sie sich beziehen. Die Bezeichnungen "Straßenadresse 1" und "Straßenadresse 2" sind interpretationsbedürftig und sollten nach Gelegenheit geändert werden.

### 3.2 Wirtschaftlichkeitsanalyse

- Das Vertriebsteam hat festgestellt, dass es regelmäßig Rückläufer für ungültige E-Mails erhält. Außerdem erhalten einige Interessenten und Kunden nicht den richtigen Inhalt, weil sie nicht richtig qualifiziert sind. Man stellt fest, dass die E-Mail-Adresse das falsche Format hat und der Kundentyp falsch ist. Derzeit ist es erforderlich, dass ein Vertriebsmitarbeiter direkt vom Kunden über Änderungen in den Kundenkontaktdaten informiert wird, oder dass der Kunde im Rahmen der Kommunikation darauf aufmerksam gemacht wird. Dies führt jedoch zu Verzögerungen, Unannehmlichkeiten und Missverständnissen für die Kunden, was wiederum den wirtschaftlichen Faktor des Unternehmensimages beeinträchtigt. Die wirtschaftliche Berücksichtigung und die Entscheidung, ob die Realisierung des Projekts gerechtfertigt ist, wird in den folgenden Abschnitten vorgenommen.

#### 3.2.1 „Make or Buy“-Entscheidung

- Da es sich bei den eingehenden Dokumenten um firmenspezifische Prozesse der Fanuc Europe GmbH handelt, ist eine Lösung durch ein zugekauftes Produkt nicht möglich. Daher muss eine Lösung durch die Fanuc Europe GmbH entwickelt werden.



### 3.2.2 Projektkosten

- Die Projektkosten, die durch die Erstellung dieser Anwendung entstanden sind, werden im Folgenden beschrieben. Es ist zu beachten, dass die Zahlen aus Datenschutzgründen angepasst wurden und daher nicht korrekt sind. Darin enthalten sind auch fixe Kosten wie Strom, Miete und Lizenzen, aber auch Zeitvergütungen für Mitarbeiter, die diese Anwendung bei der Erstellung und auch bei der Dokumentation unterstützt haben.

Der Auszubildende erhält einen Stundensatz von 15 €, während das reguläre Personal mit 35 € entlohnt wird. Hinzu kommen Kosten in der Höhe von 10 € pro Stunde für die Nutzung der Ressourcen.

Vorgang	Zeit	Personal	Ressourcen
Entwicklungskosten der Projektarbeit	70 h	1050 €	1050 €
1579,20 €			
Hilfestellung der Dokumentation	2 h	70 €	20 €
Abnahmetest	1 h	35 €	
10 €			
Hilfestellung bei Problemen	2 h	25 € + 15 € = 40 €	1000 €
			<b>2739,20 €</b>

Tabelle 2: Kostenaufstellung

### 3.2.3 Amortisationsdauer

- Im Folgenden wird festgestellt, an welchem Punkt sich die Entwicklung der Software amortisiert hat. Durch die Implementierung dieser Software in die internen Geschäftsabläufe wird der Faktor Datenqualität erhöht. Es gibt eine große Anzahl verschiedener Dimensionen und Kriterien, mit denen die Datenqualität beschrieben werden kann. Durch die Art und Weise, wie die Software aufgebaut ist, wird das Potenzial für Unannehmlichkeiten für den Kunden reduziert – was den potenziellen Handel schneller und konsistenter macht. Die CRM-Administratoren haben außerdem einen besseren Überblick und können inkonsistente Daten besser identifizieren. Da es möglich ist, die Software automatisch im Rahmen von geplanten Arbeitsabläufen auszuführen, ist der manuelle Arbeitsaufwand sehr gering, wodurch die Personalkosten auf ein Minimum reduziert werden. Im Folgenden soll nun die Arbeitszeiterparnis in tabellarischer Form ermittelt werden. Die Anzahl der Vorgänge pro Quartal und die Zeit pro Vorgang wurden durch den Salesforce-Administrator ermittelt

**Beispielrechnung (verkürzt)** Bei einer Zeiteinsparung von 10 Minuten am Tag für jeden der 5 Salesforce-Administratoren und 220 Arbeitstagen im Jahr ergibt sich eine gesamte Zeiteinsparung von

TODO: Rechnen

Die Amortisationszeit beträgt also...

## Beispiel

### 3.3 Anwendungsfälle

- Wichtige Anwendungsfälle für dieses Projekt sind Verfahren, die in den täglichen Arbeitsablauf eines Salesforce-Administrators einfließen. Ein Salesforce-Admin kann planen, wann das Programm ausgeführt wird, ohne dass danach viel Input nötig ist.

Eine grafische Darstellung des Anwendungsfalls ist in Abbildung 2 zu finden.

Voraussetzung dafür ist, dass der Salesforce-Administrator Zugang hat und die richtigen Daten zur Analyse auswählt. Hierfür wird eine Benutzerdokumentation erstellt, die in Anhang 3 zu finden ist.

Diese Anwendung ist nicht nur für die Korrektur alter Kundendaten vorgesehen, sondern kann auch für jeden neuen Datensatz, der wöchentlich in Salesforce eingegeben wird. Hierfür wird wöchentlich ein Bericht mit allen neuen Kontodaten erstellt und in das Programm übertragen. Das wichtigste Erfolgsszenario besteht darin, dass das Programm aufschlussreiche Daten liefert, die keine großen Eingaben seitens des Benutzers erfordern. Für den Fall, dass der Webscraper die falschen Daten abrufen oder nicht in der Lage ist, die vorhandenen Daten zu verarbeiten, werden Fehlerprotokolle exportiert, um dem Benutzer eine Vorstellung davon zu geben, warum die Daten nicht ausgewertet werden können.

### 3.4 Lastenheft

- Bevor das Projekt in die Umsetzungsphase eintritt, wurde ein Lastheft erstellt, in dem alle von den Salesforce-Administratoren gestellten Anfragen aufgelistet sind. Die Anwendung muss alle aufgeführten Aufgaben erfüllen. Alle Einzelkriterien wurden in User Stories aufgeteilt, die in einem Kanban-Board mit Trello organisiert wurden. Ein Abschnitt ist auf Anlage 4 zu sehen.

### 3.5 Lastenheft (Auszug)

Es folgt ein Auszug aus dem Lastenheft mit Fokus auf die Anforderungen:

Die Anwendung muss folgende Anforderungen erfüllen:

#### 1. Herauslesen der Stammdaten

- 1.1. Die Anwendung muss Kundendaten entweder als CSV oder direkt aus Salesforce verarbeiten können.
- 1.2. Die Anwendung muss die Daten auf einheitliche Weise formatieren und Formatierungsregeln anwenden.

#### 2. Darstellung der Daten

- 2.1. Die Anwendung muss eine einfache Möglichkeit bieten, nach Datensegmenten zu suchen und Berichte auf der Grundlage dieser Segmente zu erstellen.
- 2.2. Fehler und Anomalien müssen dem Nutzer deutlich angezeigt werden.
- 2.3. Der Benutzer sollte die Möglichkeit haben, geplante Jobs zu erstellen, zu sehen und zu löschen
- 2.4. Die Anwendung sollte die Daten in Salesforce nicht überschreiben, sondern den Benutzer auf Diskrepanzen aufmerksam machen
- 2.5. CSV-Daten, die direkt vom Dataloader kommen, sollten ohne Probleme in der Anwendung eingelesen werden können
- 2.6. Alle eingelesenen Daten müssen in der Benutzeroberfläche sichtbar sein

#### 3. Webscraper

- 3.1. Der Webscraper muss in der Lage sein, eine gegebene URL zu validieren und zu prüfen, ob die entsprechende Webseite verfügbar ist.
- 3.2. Der Webscraper sollte nicht versuchen, ungültig formatierte URLs anzupassen.
- 3.3. Der Benutzer sollte die Möglichkeit haben, Einstellungen für den Webscraper vorzunehmen, einschliesslich der Anzahl der maximalen Seiten und ob http-Seiten (unsichere Seiten) ausgelesen werden dürfen
- 3.4. Der Webscraper muss auf der Kommandozeile laufen können, unabhängig von der Schnittstelle.
- 3.5. Wenn möglich, soll der Webscraper versuchen nach einer Kontaktseite zu suchen, wenn die Informationen nicht auf der Homepage aufgeführt sind.

#### 4. Sonstige Anforderungen

- 4.1. Die Anwendung sollte den Salesforce-Administratoren auf einem gemeinsamen Dateiserver zur Verfügung gestellt werden
- 4.2. Die Daten der Anwendung müssen jede Nacht bzw. nach jedem **SVN!**-Commit automatisch aktualisiert werden.

- 4.3. Die Anwendung soll jederzeit erreichbar sein.
- 4.4. Da sich die Entwickler auf die Anwendung verlassen, muss diese korrekte Daten liefern und darf keinen Interpretationsspielraum lassen.
- 4.5. Mit dem Fortschreiten des CRM-Projekts müssen in späteren Phasen neue Datentypen implementiert werden.