

Projektantrag

1 Projektbezeichnung

AO-Beitragsportal

1.1 Kurzform der Aufgabenstellung

Im Rahmen des Projektes zur Automatisierung des Prozesses der Beitragsanpassung und zur Entlastung der Mathematik-Abteilung soll eine Webanwendung mit Datenbankzugriff erstellt werden, die automatisch verschiedene Daten in Bezug auf die Beiträge und Kosten eines Versicherungstarifes ermittelt, verarbeitet und darstellt.

1.2 Ist-Analyse

Die ALTE OLDENBURGER Krankenversicherung AG, im Folgenden AO genannt, ist eine private Krankenversicherung. Die Fachbereiche benötigen zur Erstellung von Versicherungsangeboten eine Übersicht über die Höhe des Beitrages in einem Tarif, der bei einem Abschluss abhängig von Geschlecht und Alter gezahlt werden muss. Aktuell werden dafür von der Mathematik-Abteilung manuell Excel-Tabellen mit Werten aus einer Datenbank gefüllt, was zeitaufwendig ist und oftmals zu Fehlern führt, die korrigiert werden müssen. Diese Dateien werden dann an die EDV-Abteilung weitergegeben, damit sie mit TortoiseSVN versioniert auf einem Netzlaufwerk abgelegt und vom Fachbereich eingesehen werden können.

Durch das Versicherungsvertragsgesetz (VVG) ist die AO dazu verpflichtet, eine Übersicht darüber zu führen, wie hoch die einmaligen und laufenden Kosten bei einem Vertragsabschluss eines Versicherten in einem Tarif sind. Auch diese Daten sind in einer Datenbank vorhanden und werden durch die Mathematik-Abteilung händisch in Excel-Tabellen eingepflegt und an die EDV-Abteilung gereicht. Da diese Daten an externe Softwarehäuser geliefert werden müssen, die verschiedene Dateiformate vorschreiben, muss eine manuelle Umformatierung und Konvertierung vorgenommen werden, bevor diese Dateien ebenfalls versioniert auf einem Netzlaufwerk abgelegt werden.

Ebenfalls wird durch das VVG vorgeschrieben, dass eine Übersicht über die Beitragsverläufe der letzten zehn Jahre einer 35-jährigen Person für alle Tarife erstellt werden muss. Dazu wird aus einer Simulation im Bestandsführungssystem der AO eine Datei im CSV-Format generiert, welche dann wieder importiert werden muss.

Diese Schritte sind jeweils zu einer Beitragsanpassung (BAP) notwendig, die jährlich zum 01. Januar und gegebenenfalls auch zum 01. Juli stattfindet. Dabei kann der Beitrag, der in einem Tarif bezahlt wird, z. B. aufgrund von höheren Kosten im Gesundheitsbereich angehoben oder gesenkt werden. Laut VVG ist es notwendig, eine Übersicht zu pflegen, in der ersichtlich wird, wann der Beitrag eines bestimmten Tarifes zum letzten Mal durch eine BAP verändert wurde. Dazu werden momentan alte Excel-Tabellen der oben genannten Beiträge durchgeschaut, bis eine Änderung ersichtlich ist.

Eine fachliche Prüfung der erstellten Daten ist in allen Fällen momentan nicht gewährleistet.

2 Zielsetzung entwickeln/Sollkonzept

2.1 Was soll am Ende des Projektes erreicht sein?

Ziel des Projektes ist die Erstellung einer Web-Anwendung mit grafischer Benutzeroberfläche und einer Anbindung an eine externe Datenbank, die die im nächsten Punkt genannten Anforderungen umsetzt. Die Mathematik-Abteilung soll von der manuellen Erstellung von Excel-Dateien durch eine Automatisierung entlastet werden. Zudem soll das Projekt komplett automatisiert werden, was alle Schritte von Continuous Integration bis hin zum Deployment mit einschließt.

2.2 Welche Anforderungen müssen erfüllt sein?

Folgende Anforderungen sollen durch das Projekt erfüllt werden:

- Anzeigen der Erstbeiträge, Kosten und Beitragsverläufe für jeden Tarif
- Exportieren der Kostenübersicht in verschiedene Formate wie PDF, Excel und CSV zur Weitergabe an externe Partner
- Kontrollieren der ermittelten Daten durch Plausibilitäten
- Anzeigen der Historie der Änderungen eines Tarifes durch eine BAP
- Ermitteln des letzten BAP-Datums eines Tarifes
- Speichern der Beitragsverläufe in einer Datenbank statt als CSV-Datei
- Einhalten von Ergonomie-Richtlinien in Bezug auf die Gestaltung
- Beachtung und Einhaltung des Corporate Design der AO
- Bereitstellen eines Single Sign Ons gegen das Active Directory
- Bereitstellen von Continuous Integration
- Einhalten der innerbetrieblichen Coding-Richtlinien

2.3 Welche Einschränkungen müssen berücksichtigt werden?

Aufgrund architektonischer Entscheidungen im Unternehmen muss diese Anwendung mit Java EE programmiert werden. Des Weiteren muss diese Webanwendung ausschließlich auf dem Browser Mozilla Firefox (Version 38.2) auf einem Desktop-PC lauffähig sein.

Aufgrund der Vorgabe, dass die Anwendung mit dem Corporate Design der AO harmonisieren soll, wird ein bereits bestehendes Stylesheet für die Darstellung verwendet werden. Dieses muss dann entsprechend angepasst werden.

Zudem ist anzumerken, dass die Ermittlung der Beitragsverläufe weiterhin im Bestandsführungssystem simuliert werden soll. Lediglich die Speicherung als Text-Datei soll durch eine neu anzulegende Datenbankstruktur abgelöst werden.

3 Projektstrukturplan erstellen

3.1 Was ist zur Erfüllung der Zielsetzung erforderlich?

Grundsätzlich soll sich die Umsetzung des Projektes an den Phasen des Wasserfall-Modells orientieren. Überwiegend sind die Anforderungen durch das VVG eindeutig definiert. Jedoch soll es im Bereich der Oberflächengestaltung durch regelmäßige Rücksprache mit dem Fachbereich einen agilen Prozess geben. Ebenso wird in Bezug auf die fachliche Logik und Prüfung der verarbeiteten Daten durch Plausibilitäten in iterativen Zyklen ein Anwendertest durch den Fachbereich schon in die Implementierungsphase integriert, sodass eine eigenständige Testphase entfallen kann. Durch diese Einbindung des Fachbereichs in den Entwicklungsprozess kann die Abnahmephase kürzer ausfallen.

Des Weiteren wird mithilfe von Unit-Tests testgetrieben entwickelt. Für die Versionsverwaltung soll Git genutzt werden und eine Dokumentation des Projektfortschrittes soll im Ticket-System Redmine vorgenommen werden. Als Build-Tool soll Gradle verwendet werden. Mit Jenkins soll kontinuierliche Integration sichergestellt werden.

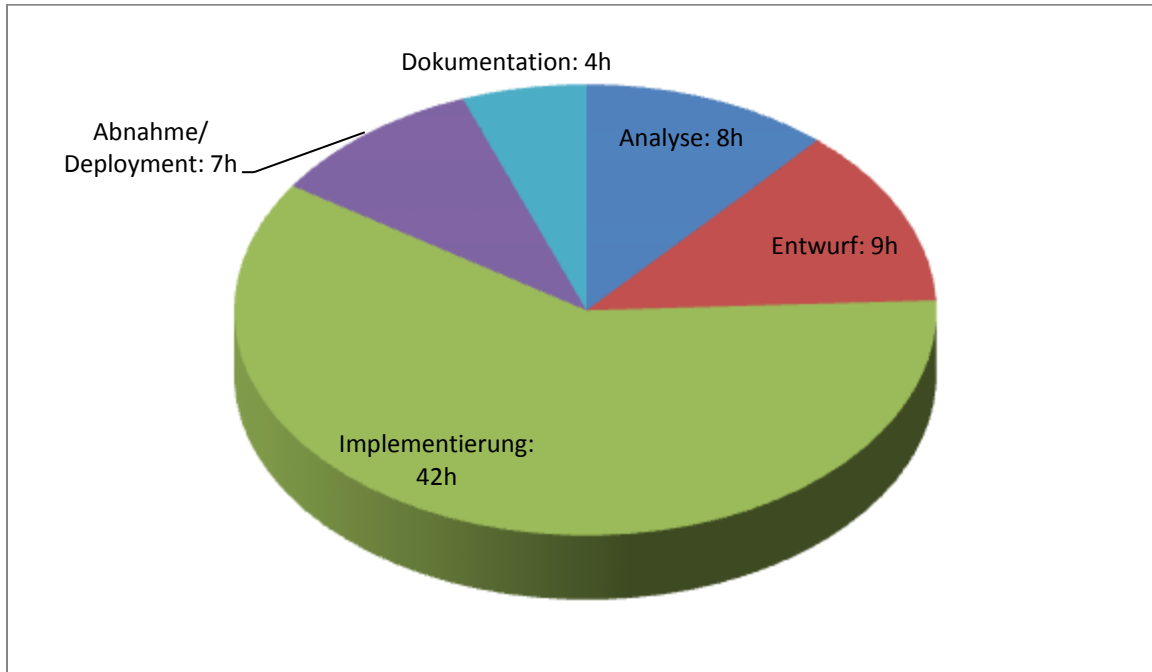
3.2 Aufgaben auflisten

- Analyse
 - Durchführen der Ist-Analyse
 - Durchführen der Wirtschaftlichkeitsanalyse inkl. Erstellen einer Break-Even-Analyse und Amortisationsrechnung
 - Ermitteln von Use-Cases und Erstellen eines Anwendungsfall-Diagramms
 - Unterstützen des Fachbereiches beim Erstellen des Lastenheftes
- Entwurf
 - Entwerfen der Benutzeroberfläche inkl. Erstellen von Mock-Ups
 - Entwerfen der Datenbankstruktur inkl. Erstellen eines ER-Modells
 - Ableitung des Tabellen- und Domänenmodells aus dem ER-Modell inkl. Erstellen eines Klassendiagramms
 - Planung der Architektur inkl. Erstellen eines Komponentendiagramms
 - Erstellen des Pflichtenheftes
- Implementierung inkl. Tests und Feedback vom Fachbereich
 - Anlegen des Java-Projekts
 - Einrichten des Gradle-Builds und der statischen Code-Analyse
 - Implementieren der Speicherung der Beitragsverläufe in einer Datenbank in Natural
 - Implementieren der Domänenklassen inkl. Tests
 - Herstellen der Datenbank-Verbindung in Java inkl. Tests
 - Implementieren der Datenverarbeitung in Java
 - Implementieren der Verarbeitung der Erstbeiträge inkl. Tests
 - Implementieren der Verarbeitung der Kosten inkl. Tests
 - Implementieren der Verarbeitung der Beitragsverläufe inkl. Tests
 - Implementieren der Ermittlung der BAP-Daten inkl. Tests
 - Implementieren der Oberfläche inkl. Tests
- Abnahme und Deployment
 - Code-Review mit dem Ausbilder
 - Abnahme durch die Fachabteilung
 - Continuous Integration (Einrichten eines automatischen Jenkins-Builds)
 - Deployment der Anwendung
- Dokumentation
 - Erstellen der Entwicklerdokumentation mit JavaDoc
 - Erstellen des Benutzerhandbuchs

3.3 Grafische und tabellarische Darstellung

z. B. ein Netzwerkplan. Bitte als PDF-Anhang am Ende des Schemas hochladen und an dieser Stelle bitte unbedingt einen Hinweis auf den Anhang vermerken.

Grafische Darstellung



Tabellarische Darstellung

Phase	Zeit in Stunden
Analyse	8
Entwurf	9
Implementierung	42
Abnahme und Deployment	7
Dokumentation	4

4 Projektphasen mit Zeitplanung in Stunden

Analysephase	8h
• Durchführen der Ist-Analyse	2h
• Durchführen der Wirtschaftlichkeitsanalyse inkl. Erstellen einer Break-Even-Analyse und Amortisationsrechnung	2h
• Ermitteln von Use-Cases und Erstellung eines Anwendungsfall-Diagramms	2h
• Unterstützung des Fachbereichs bei der Erstellung des Lastenheftes	2h
Entwurfsphase	9h
• Entwerfen der Benutzeroberfläche inkl. Erstellen von Mock-Ups	2h
• Entwerfen der Datenbankstruktur inkl. Erstellen eines ER-Modells	2h
• Ableitung des Tabellen- und Domänenmodells aus dem ER-Modell inkl. Erstellen eines Klassendiagramms	1h
• Planung der Architektur inkl. Erstellen eines Komponentendiagramms	1h
• Erstellen des Pflichtenheftes	3h
Implementierungsphase inkl. Tests und Feedback vom Fachbereich	42h
• Anlegen des Java-Projekts	1h
• Einrichten des Gradle-Builds und der statischen Code-Analyse	2h
• Implementieren der Speicherung der Beitragsverläufe in einer Datenbank in Natural	4h
• Implementieren der Domänenklassen inkl. Tests	5h
• Herstellen der Datenbank-Verbindung in Java	2h
• Implementieren der Datenverarbeitung in Java	21h
○ Implementieren der Verarbeitung der Erstbeiträge inkl. Tests	6h
○ Implementieren der Verarbeitung der Kosten inkl. Tests	5h
○ Implementieren der Verarbeitung der Beitragsverläufe inkl. Tests	6h
○ Implementieren der Ermittlung der BAP-Daten inkl. Tests	4h
• Implementieren der Oberfläche inkl. Tests	7h
Abnahme und Deployment	7h
• Code-Review mit dem Ausbilder	2h
• Abnahme durch die Fachabteilung	1h
• Continuous Integration (Einrichten eines automatischen Jenkins-Builds)	2h
• Deployment der Anwendung	2h
Dokumentation	4h
• Erstellen der Entwicklerdokumentation mit JavaDoc	1h
• Erstellen des Benutzerhandbuchs	3h

5 Name der Ausbildungsstätte in dem das Projekt durchgeführt wird

ALTE OLDENBURGER Krankenversicherung AG

EDV-Abteilung

5.1 Name des Ausbilders, bzw. Projektverantwortlichen mit Angabe der Tel. Nr.

Stefan Macke (Ausbilder)

Tel.: ----/-----

E-Mail: -----