

Eingereicht von

**Maja Dusanic**

**Merisa Pargan**

**Anel Ljutic**

**Ivan Samardzic**

**Milos Tomic**

Angefertigt am

**Institut für Wirtschaftsinformatik – Information Engineering**



Betreuer / Betreuerin

**Dr. David Rückel**

**Dr. Barbara Krumay**

**Univ.-Prof. Dr. Stefan Koch**

Partner

**Institut für Wirtschaftsinformatik – Information Engineering**

06 2019



Seminararbeit

zur Erlangung des akademischen Grades

Bachelor of Science (MSc/BSc)

im Bachelorstudium

Wirtschaftsinformatik

Evaluierung einer CRm-lösung

**Institut für Wirtschaftsinformatik – Information Engineering**

Inhaltsverzeichnis

[1. Problem 3](#_Toc12942828)

[1.1. Einleitung 3](#_Toc12942829)

[1.2. Problembeschreibung und -nachweis 3](#_Toc12942830)

[1.3. Ziele 4](#_Toc12942831)

[2. Problemlösungsweg 4](#_Toc12942832)

[2.1. Literaturrecherche 4](#_Toc12942833)

[2.2. Anforderungskatalog / Anforderungsanalyse 15](#_Toc12942834)

[2.3. Vorgehensmodell & Umsetzungsplanung 23](#_Toc12942835)

[3. Ergebnisse 27](#_Toc12942836)

[4. Tabellenverzeichnis 27](#_Toc12942837)

[5. Abbildungsverzeichnis 27](#_Toc12942838)

[6. Literaturverzeichnis 28](#_Toc12942839)

[7. Anhang 29](#_Toc12942840)

# Problem

## Einleitung

Für Unternehmen und Organisationen hat heutzutage das Grundwissen über Kunden eine große Bedeutung. Daten sind das wichtigste Kapital in jeder Organisation, aus dem Informationen extrahiert und in Wissen umgewandelt werden können.

Insbesondere der Einsatz von IT ermöglicht Informationen von Kunden effektiv zu sammeln und zu verwalten.

Wie in jeder Organisation müssen starke Beziehungen zu den Kunden aufgebaut werden. Das sogenannte Kundenbeziehungsmanagement verleiht diesbezüglich neue Dimensionen und ermöglicht mit "Customer Relationship Management", oft als CRM abgekürzt, alle diese Informationen in einer einzigen CRM-Datenbank zu speichern. Mit der CRM-Software können Unternehmen hauptsächlich Geschäftsbeziehungen, die damit verbundenen Daten und Informationen verwalten. Daneben kann mit CRM andere Aspekte wie zum Beispiel das Managen von Events durchgeführt werden. Da solche Programme viele Funktionalitäten anbieten muss jede Organisation für sich selbst bestimmen, welche Lösung den Anforderungen anspricht und mit welchem CRM einen Mehrwert für das Unternehmen generiert werden kann.

## Problembeschreibung und -nachweis

Das Institut für Wirtschaftsinformatik – Information Engineering hat vor einiger Zeit eine Open Source CRM-Lösung in Betrieb genommen, die vor allem dazu genutzt wird, Einladungen zu den Events (Business-Science-Brunch, Abschlussevent) zu managen. Allerdings entspricht das Modul zur Verwaltung von Events nicht den aktuellen Anforderungen, sondern muss mit temporären Workarounds händisch angepasst werden (z.B. Export / Import der Gäste).

Aus diesem Grund soll eine Entscheidung getroffen werden, ob entweder eine andere Open Source Software eingesetzt wird, die den aktuellen Anforderungen besser entspricht oder das derzeitige CRM System, um die fehlenden Anforderungen zu erweitern beziehungsweise das Event-Modul neu zu gestalten.

Dabei werden die wichtigsten Erkenntnisse aus der Literatur zu Customer-Relationship-Management, Eventmanagement, Open Source, und Einsatz von Softwarelösungen im non-profit Organisationen zusammengefasst. Zudem werden die Anforderungen mithilfe von Interviews sowie einem Workshop mit den Haupt-Nutzern erhoben. Durch die Analyse der Anforderungen soll entweder der Ansatz zu Neuimplementierung oder der Erweiterung verfolgt werden und die Software diesbezüglich angepasst werden.

## Ziele

Durch die fehlenden Funktionalitäten und geringer Usability der derzeitigen CRM Lösung vom Institut Information Engineering ist daher Ziel dieses Projekts, eine CRM Lösung zu implementieren, die den Anforderungen entspricht. Das Projekt gilt als erfolgreich umgesetzt, wenn die Lösung vom Auftraggeber abgenommen wurde und eine kurze Einschulung mit den Beteiligten vorgenommen wurde.

# Problemlösungsweg

## Literaturrecherche

**Customer Relationship Management**

In den letzten Jahren wurde der Begriff CRM -Customer Relationship Management immer wichtiger für große sowie kleine Unternehmen. Mit der Entwicklung von CRM werden immer umfassendere Definitionen entwickelt, weshalb verschiedene Personen unter diesem Begriff eine unterschiedliche Bedeutung verstehen.   
Ein weit verbreitetes Verständnis ist die Verwendung von Wissen und Informationen, welches sich auf die Kunden bezieht, um relevante Dienstleistungen oder Produkte an Kunden zu liefern (Christopher, 2003). Eine andere Ansicht von CRM ist, dass es eine Technologielösung ist. Zur Verbesserung von soll CRM unter diesem Verständnis die separaten Datenbanken und Tools zur Automatisierung von Außendienstmitarbeitern erweitern, um die Vertriebs- und Marketingfunktionen zu überbrücken. Andere Organisationen betrachten CRM als ein Tool, das speziell für die Kundenkommunikation entwickelt wurde. Diese Aufgabe liegt ausschließlich in den Bereichen Vertrieb / Service, Call Center oder Marketing (Chen & Popovich, 2003).

Laut (Christopher, 2003) entwickelte sich CRM aus Geschäftsprozessen wie dem Beziehungsmarketing und der zunehmenden Betonung einer verbesserten Kundenbindung durch das effektive Management von Kundenbeziehungen. *Schlussendlich* soll man unter CRM eine funktionsübergreifende, kundenorientierte, technologieintegrierte Strategie für das Geschäftsprozessmanagement, die Beziehungen maximiert und die gesamte Organisation umfasst, verstehen (Chen & Popovich, 2003).

Eine CRM-Geschäftsstrategie verknüpft die Front-Office-Funktionen und Back-Office-Funktionen. Sie nutzt also Marketing, Betrieb, Vertrieb, Kundenservice, Personal, Forschung und Entwicklung, Finanzen sowie Informationstechnologie und Internet, um die Rentabilität der Kundeninteraktionen zu maximieren (Chen und Popovich, 2003).

Unternehmen, die erfolgreich CRM implementieren, haben vielerlei Vorteile. Zu einem werden die Aussichten für die Rentabilität einer Organisation langfristig verbessert (Christopher, 2003). Zum anderem steigt die Wettbewerbsfähigkeit, was sich in höheren Einnahmen und niedrigeren Betriebskosten auswirkt. Durch die effektive und effiziente Verwaltung von Kundenbeziehungen in CRM Lösungen wird die Kundenzufriedenheit und die Kundenbindung erhöht (Chen & Popovich, 2003).

Kunden profitieren von Einfachheit und Komfort für die Durchführung von Transaktionen (Chen & Popovich, 2003), sparen Zeit und Geld und bekommen bessere Informationen (Christopher, 2003).

Eine erfolgreiche Implementierung ist jedoch für viele Unternehmen schwer umsetzbar, vor allem, weil sie nicht verstehen, dass CRM eine unternehmensweite, funktionsübergreifende, kundenorientierte Neugestaltung von Geschäftsprozessen erfordert. Mögliche Risiken wie Projektversagen, unzureichende Kapitalrendite, unglückliche Kunden, Verlust des Vertrauens der Mitarbeiter und fehlende Ressourcen müssen gut durchdacht sein (Chen & Popovich, 2003).

**Open Source Software**

Unter einer Open Sorce Software versteht man eine Software, dessen Quellcode öffentlich und frei verfügbar ist. Sie ist nicht konkurrenzfähig und setzt eine urheberrechtliche Lizenz ein, um private Patentansprüche sowohl von Software-Innovatoren als auch von Software-Adoptern aus dem Weg zu räumen. Die Erstehung von dieser Art von Software ist in der Regel mit Open Source Softwareprojekten verbunden, die häufig von einer Einzelperson oder eine Gruppe initiiert werden, deren Ziel es ist eine Software zu entwickeln, die ihren eigenen Bedürfnissen entspricht. Heutzutage gibt es eine Vielzahl solcher Projekte, und sie produzieren Code für viele verschiedene Zwecke. (von Krogh & von Hippel, 2006)

Die Motivationen für die Beiträge zu diesen Projekten sind ein zentrales Thema bei der Erforschung der Open Source Software. Frühe empirische Arbeiten zu diesem Thema dokumentierten eine Reihe von Motiven für die Beteiligung von Projektbeteiligten, wie Spaß, Reputationsaufbau, Lernen und den privaten Gebrauchswert der Software, die entwickelt wird. (von Krogh & von Hippel, 2006)

**Vorteile und Nachteile**

Die Verwendung von Open Source Software ist mit verschiedenen Vorteilen beziehungsweise Nachteilen verbunden.

Der wichtigste Vorteil sind die geringeren Anschaffungskosten. Diese haben in den meisten Fällen einen starken Einfluss auf den Entscheidungsprozess einer Non profit Organisation, wenn es um die Kaufentscheidung zwischen einer Open Source Software und einer kommerziellen Software geht.

Der nächste Vorteil ist die Möglichkeit den Code wiederzuverwenden. Die Wiederverwendung vom Code unterstützt die folgenden Szenarien (Haefliger, von Krogh, & Spaeth, 2008):

* Die Funktionalität soll schnell integriert werden
* Es wird zeitlich und technisch unter begrenzten Ressourcen gearbeitet
* Es sollen die Entwicklungskosten gesenkt werden

Zu den Vorteilen zählt auch die größere Flexibilität bei der Änderung und Verbesserung der Codebasis. Diese kann zu einer höheren Produktivität führen.

Zu den Nachteilen gehört auf der ersten Stelle die Tatsache, dass die technische Unterstützung nicht garantiert wird. Einige Benutzer bieten zwar Hilfe bei der Erstellung von Handbüchern oder der Beantwortung von Benutzerfragen an, es gibt jedoch keine Garantie dafür, dass die Frage eines Benutzers jemals beantwortet wird, da Benutzer keinen Servicevertrag mit einem Anbieter haben können. Zu den weiteren Nachteilen gehören Sicherheitsbedenken (z.B. Sicherheit der Daten) und die fehlenden vertraglichen Beziehungen. Die Kombination dieser Nachteile kann dazu führen, dass Benutzer einer Open Source Software Zeit für die Problembehebung ihrer Software verschwenden, anstatt an der Verbesserung der Software zu arbeiten. Dies kann sich in einigen Fällen negative auf die Produktivität des Unternehmens auswirken. (Nagle, 2015)

**Make vs Buy**

Bei der Entwicklung eines Informationssystems stehen Unternehmen vor einer grundlegenden Entscheidung: Ein vorentwickeltes Paket zu kaufen (möglicherweise anzupassen oder die Geschäftsprozesse des betreffenden Unternehmens an das Paket anzupassen) oder das System selbst zu erstellen. Es wurde festgestellt, dass eine Reihe von Kernfaktoren die Kauf- und Baukosten kritisch beeinflussen(Hung & Low, 2008):

* Strategie
* Rohstoff- und Wettbewerbsvorteil
* Reifegrad
* Kosten
* Anforderungen

Zu erwähnen sind auch Faktoren, die die Kauf- und Bauskoten weniger beeinflussen, dennoch aber wichtig für den Entscheidungsprozess wichtig sind: (Hung & Low, 2008):

* Risiko
* Zeit
* Politische Faktoren
* Laufende Unterstützung

**Non Profit Organisationen (NPO)**

NPO sind Organisationen, die durch folgende Merkmale gekennzeichnet sind:

* Organisation als eine Institution
* Sind privat und kein Teil der Regierung
* Selbstverwaltung ihrer Aktivitäten und Verfahren
* Keine Verteilung vom Gewinn

Diese Organisationen müssen sich verschiedenen Herausforderungen stellen, die in ihrem organisatorischen Kontext einzigartig sind. Hiermit sind unter anderem monetäre Herausforderungen gemeint (z.B. reduzierte staatliche Mittel), der zunehmende Wettbewerb, die Herausforderung Freiwillige zu rekrutieren beziehungsweise zu binden und die Notwendigkeit wirksamer Programme zur Kontaktaufnahme mit den Sponsoren. Diese Herausforderungen grenzen die allgemeine Wirksamkeit und Effizienz der NPO ein. (Rathi & Given, 2017)

NPO Technologielandschaft

Um sich den oben genannten Herausforderungen zu stellen benutzen die NPO verschiedene Technologien. Nach dem Zusammenfügen dieser Technologien ergeben sich folgende allgemeine Kategorien: (Rathi & Given, 2017)

* Low-cost/no-cost Cloud basierte Dienste
* Öffentliche Webseiten
* Kommerzielle Produktivitätssoftware
* Private Webseiten
* Open-source content management Software
* Low-cost/no-cost Produktivitätssoftware
* Enterprise content management Software

**Marktrecherche**

Hier werden praktikable Lösungen vorgestellt, in denen die Funktionen der wichtigsten CRM und Eventmanagement-Tools nach neuesten Veröffentlichungen beschrieben werden. Jedes dieser Tools wird beschrieben, um ihre Fähigkeiten, ihre Stärken und die unterschiedlichen Aspekte zu bewerten und die Anwendbarkeit jedes dieser Tools bei der Anpassung an die Anforderungen zu untersuchen.

**Suite CRM**

Suite CRM ist ein Open-Source-Tool, dass viele Funktionen, wie

Sicherheit, systemweiter und umfassender Audit-Trail, Plattformunabhängigkeit, praktisch unbegrenzt Skalierbarkeit, bietet.

SuiteCRM ist eine stark erweiterte Version von SugarCRM, die auch viele zusätzliche Sicherheitsupdates enthält, die in SugarCRM nicht verfügbar sind.

LINK zur Installation:https://www.veonconsulting.com/suitecrm-installation-steps/

|  |  |
| --- | --- |
| **Max. Kontakte** | Unbegrenzt |
| **Nutzer** | Unbegrenzt |
| **Speicher** | Unbegrenzt |
| **Features** | Kontaktverwaltung, Angebote, Marketingkampagnen, E-Mail-Kampagnen, Auswertungen, Projektverwaltung, Rechnungen, Verträge, Events usw. |
| **Plattform** | Web |
| **Mobile** | keine |
| **Relevanten Sprachen** | Englisch, (Deutsch – von diligent kostenlos bereitgestellt) |
| **Programmiersprache** | PHP |
| **Betriebssystem** | Cross-platform |
| **Data backend** | MariaDB, MySQL, Microsoft SQL Server |

Nachteil:

* Die Kontakt- oder Accountübersicht kann bei einer sehr großen Zahl ungünstig sein.

Vorteil:

* Man kann Prozesse automatisieren.
* Ein Tool mit sehr vielen Möglichkeiten. [2]
* Es gibt einen Bug Tracker auf GitHub

**1CRM**

Diese CRM Lösung ist für kleinere bis mittlere Unternehmen geeignet und bietet umfangreiche Lösungen an, die auf einen eigenen Server betrieben wird. 1CRM bietet auch die Möglichkeit, Veranstaltungen zu planen, durchzuführen und auszuwerten. Das ist nicht nur aus Marketing-Sicht für Produktpräsentationen interessant, sondern auch für Ausbildungen, Seminare und Lehrgänge. Kontakte und Interessenten können Veranstaltungen zugewiesen werden, Erinnerungsmails an alle Teilnehmer sind ebenso möglich wie die Überprüfung der Anwesenheit.

LINK:https://1crm-system.de

|  |  |
| --- | --- |
| **Max. Kontakte** | 750 |
| **Nutzer** | 3 |
| **Speicher** | Unbegrenzt |
| **Features** | Terminverwaltung, Kalender, E-Mail Management, Beziehungsmanagement, Zeiterfassung, Kontakt- & Kundenverwaltung, Eventmangement, Service und Supportmanagement, Import & Export |
| **Plattform** | Web |
| **Mobile** | iOS |
| **Relevanten Sprachen** | Deutsch |
| **Programmiersprache** | PHP |
| **Kompatible Betriebssysteme** | Linux |
| **Data backend** | MariaDB, MySQL |

Nachteil:

· Design und Usability sind auf mobilen Geräten nicht auf dem neusten Stand.

Vorteil:

· Großer Funktionsumfang

· Einfache Personalisierung per Drag & Drop

· Hohe Flexibilität bei der Lizenzierung

**CiviCRM**

CiviCRM ist ein Open-Source-Tool, das auf Relationship-Management-Lösungen basiert. Dieses Tool ist webbasiert, mehrsprachig und richtet sich an Interessenvertreter, gemeinnützige Organisationen und Nichtregierungsorganisationen. Mit CiviCRM können Daten für die verschiedenen Bestandteile einer Organisation aufgezeichnet werden.

Das webbasierte Tool lässt sich direkt in die beliebten Open-Source-Content-Management-Systeme Drupal und Joomla integrieren und bietet Mehrsprachenunterstützung wie Französisch, Spanisch, Deutsch, Niederländisch,

Portugiesisch usw. Die Software ist lizenziert unter der GNU AGPL, die Änderungen im Code erlauben.

Die Hauptstärken des Tools sind:

* Speichern von Informationen über Personen, Organisationen und Haushalte
* einfache Integration mit anderen Websites
* für den weltweiten Einsatz konzipiert
* Möglichkeit, mehrere Personen gleichzeitig an verschiedenen Standorten zu verbinden
* Zugänglichkeit innerhalb und außerhalb der Unternehmensinfrastruktur
* Kommunikation zwischen Personen praktisch und zugänglich zu machen.

|  |  |
| --- | --- |
| **Max. Kontakte** | unbegrenzt |
| **Nutzer** | unbekannt |
| **Speicher** | unbegrenzt |
| **Features** | Mitgliederverwaltung, Kontaktverwaltung, Spendenverwaltung, Eventmanagement, Kampagnenmanagement, Auswertungen |
| **Plattform** | Web |
| **Mobile** | - |
| **Relevanten Sprachen** | Deutsch, Englisch |
| **Programmiersprache** | JavaScript, PHP |
| **Betriebssystem** | Server: Linux  Client: plattformunabhängig |
| **Database backend** | MySQL & MariaDB |

Vorteil:

* Sehr große Funktionsvielfalt

**SugarCRM**

SugarCRM ist eine der führenden CRM-Lösungen

und einer der bekanntesten Namen in der Branche. SugarCRM enthält einige Funktionen, z.B.: Automatisierung der Vertriebsabteilung, Marketingkampagnen, Kundensupport, Zusammenarbeit und Berichterstellung. Es ist in hohem Maße anpassbar und es sind zahlreiche Plugins und Erweiterungen online verfügbar. Die Software in der Version 5.0 ist unter den Bedingungen der GNU GPLv3 veröffentlicht.

|  |  |
| --- | --- |
| Max. Kontakte | unbegrenzt |
| Nutzer | unbekannt |
| Speicher | unbekannt |
| Features | Mitgliederverwaltung, Kontaktverwaltung, Spendenverwaltung, Eventmangement, Kampagnenmangement, Auswertungen, keine Veranstaltungen, Service und Support Management, kein Projekt Management |
| Plattform | Web, (cloudbasiert) |
| Mobile | Android |
| Relevanten Sprachen | Englisch |
| Betriebssystem | Linux, Windows, Unix |
| Programmiersprache | PHP |
| Database backend | MySQL, Microsoft SQL Server |

Vorteile:

* Aktivitäten, Kontakte, Kampagnen, Leads und Chancen importieren sowie exportieren

Nachteile:

* Keine Weiterentwicklung des Quellcodes verfügbar
* Die Sugar Community Edition ist daher nur bedingt, und somit etwa für Test- und Trial-Zwecke zu empfehlen.

**Odoo**

Odoo ist eine integrierte ERP-Software-Lösung. Zielgruppe sind Unternehmen jeder Größenordnung. Odoo beinhaltet folgende Module: CRM, Website/E-Commerce, Abrechnung, Finanzbuchhaltung, Produktion, Lagerverwaltung, Projektmanagement, Dokumentenmanagement. Odoo bietet eine Vielzahl von Funktionen für jede Phase des Ereignisplanungsprozesses.

|  |  |
| --- | --- |
| Max. Kontakte | abhängig von Server |
| Nutzer | unbegrenzt |
| Speicher | Nicht angegeben |
| Features | Ticketing, Event schedule, Social media integration, Email marketing, Google Analytics integration, Sponsor management |
| Plattform | Web |
| Mobile | nicht vorhanden |
| Relevanten Sprachen | Deutsch, Englisch |
| Betriebssystem | Linux, Unix-like, OS X, Window |
| Programmiersprache | Python, JavaScript, XML |
| Database backend | PostgreSQL |

Vorteile:

* erweiterbar, flexibel, äußerst benutzerfreundlich
* Erleichtert das Tätigen von Anrufen, Dokumente elektronisch signieren, E-Mails versenden und vieles mehr aus einem einzigen Modul heraus.
* Die Integration von Apps wie Astrix eröffnet Branchen, die täglich auf eingehende und ausgehende Anrufe angewiesen sind, neue Möglichkeiten.

## Anforderungskatalog / Anforderungsanalyse

Unter Systemanforderungen versteht man Anforderungen an das System und die Umgebung, in der sich das System befindet. Durch die Beschreibung der genauen Anforderungen für das Systemverhalten und die Architektur wird die Grundlage für die Entwicklung dieses Systems gelegt. Die sich daraus ergebenden Anforderungsspezifikationen stellen dabei die Grundlage für die Analyse der Anforderungen dar. Sie geben an, dass sie tatsächlich das sind, was die Interessengruppen wollen, definieren, was Konstrukteure schaffen müssen, und vergewissern sich, dass sie dies bei der Lieferung der Ergebnisse richtig gemacht haben.

Es gibt viele Ansätze beziehungsweise Techniken, die man verwenden kann, um Anforderungen einer Software zu spezifizieren, analysieren und validieren. Einer dieser Ansätze ist IEEE 830-1998-Standard. Dieser schlägt eine Vorlage vor, die mehrere Anforderungsartefakte identifiziert. Die wichtigsten werden in einem Abschnitt mit der Bezeichnung "Spezifische Anforderungen" beschrieben. Diese Artefakte umfassen: "Externe Schnittstellen" für Benutzer, Hardware, Software und / oder Kommunikation, "Funktionen" der Software, "Einschränkungen" für Software; "Design Constraints" usw. (McIlroy & Stanton, 2012; Navarro, Letelier, Mocholi, & Ramos, 2018)

**Anforderungsanalyse**

Im Folgenden wird eine Anforderungsanalyse der Interview-Ergebnisse durchgeführt, die durch die Befragung folgender Personen zu Stande gekommen ist: Frau Angela Dorfer vom Institut Information Engineering (JKU), Frau Dr. Barbara Krumay vom Institut Information Engineering (JKU), Herr Dr. David Christoph Rückel vom Institut Information Engineering (JKU) und Herr Michael Holoubek vom Institut Information Engineering (JKU).

Als Ansatz für die Anforderungsanalyse wird IEEE 830-1998-Standard verwendet. Es werden dabei nicht alle Punkte durchgenommen, weil die Ergebnisse der Interviews nicht alle Punkte behandeln. (Elliott, Sr. & Allen, 2013; Standard, 2010)

**Zweck**

Zweck des Programms ist es grundlegende Funktionen einer CRM-Lösung für Kontakt- beziehungsweise Eventmanagement zu bieten. Dies beinhaltet unter anderem Informationen über Kunden zu verwalten und Einladungen für spezifische Events auszusenden.

**Allgemeine Beschreibung**

Im weiteren Verlauf werden die allgemeinen Eigenschaften des Programms beschrieben.

**Allgemeine Funktionalität**

Die allgemeine Funktionalität des Programms besteht darin Informationen über Kunden zu sammeln, Kontakte zu organisieren, Einladungen herauszuschicken und Daten in gewünschter Form zu darstellen.

**System und Hardware Einschränkungen**

* Das Programm muss auf folgenden operativen Systemen lauffähig sein: Windows, Linux und OSX

-need to have

* Es sollen kein cloudbasiertes System sein.

-need to have

* Das Programm soll auf mobilen Endgeräten lauffähig sein

-nice to have

**Rechtliche Rahmenbedingungen**

* Bezüglich der Benutzung dürfen keine Lizenzprobleme vorhanden sein.

-need to have

**Sonstige Einschränkungen**

* Das Programm muss als Open Source Version verfügbar sein

-need to have

* Der Quellcode des Programms muss erweiterbar sein.

-need to have

**Benutzerzielgruppen**

Insgesamt sollen die folgenden Benutzerrollen existieren:

* Superadministrator
* Administrator
* Editor (Studentischer Mitarbeiter)
* Studentische Hilfskraft

**Spezifische Anforderungen**

Im Folgenden werden die Funktionalitäten, welche das Programm erfüllen muss, folgend „need to have“, und weiteres Anforderungen, die das Programm umsetzen sollte, folgend „nice to have“, behandelt.

***Funktionale Anforderungen***

* Funktionale Anforderung 1

Das Programm muss in der Lage sein, Kontakte zu hinzufügen. Diesbezüglich werden folgende Informationen eingefügt: „Titel“, „Vorname“, „Nachname“, „Email“, „Nummer“, „Adresse“, „DSGVO“

-need to have

* Funktionale Anforderung 2

Das Programm muss in der Lage sein, Zusatzinformationen zu den Kontakten beziehungsweise zu den Einladungen zu erstellen.

-need to have

* Funktionale Anforderung 3

Das Programm muss in der Lage sein, die Informationen über die Kunden schnell ändern zu können und folgend die Kundeninformationen in derselben Maske bearbeiten zu können. (Kein neues Fenster zur Bearbeitung der Kundeninformationen öffnen)

-need to have

* Funktionale Anforderung 4

Das Programm muss in der Lage sein, ein Event zu erstellen. Diesbezüglich werden folgende Informationen eingefügt: „Name“, „Datum“, „Ort“, „Veranstalter“, „Kommentar“.

-need to have

* Funktionale Anforderung 5

Das Programm muss in der Lage sein, beim Hinzufügen von Events einen Kalender anzuzeigen, um das Datum des geplanten Events einfacher zu selektieren und direkt eintragen zu können.

-need to have

* Funktionale Anforderung 6

Das Programm muss in der Lage sein, Mails beziehungsweise Einladungen an ausgewählte Kontakte zu versenden.

-need to have

* Funktionale Anforderung 7

Das Programm muss in der Lage sein, beim Erstellen einer Mail ein Template (z.B. JKU-Daten) auszuwählen und hinzuzufügen. Die Auswahl der Templates basiert auf den benutzten Templates der letzten sechs Monate.

-need to have

* Funktionale Anforderung 8

Das Programm muss in der Lage sein, vollautomatisiert Einladungen für Events an definierte Kontakte zu versenden.

-need to have

* Funktionale Anforderung 9

Das Programm muss in der Lage sein, Massenmails zu versenden.

-need to have

* Funktionale Anforderung 10

Das Programm muss in der Lage sein bei jedem Kontakt einen Status anzulegen. Dieser Status zeigt an ob der Kontakt die Einladung zu einem Event akzeptiert, abgelehnt oder auf die Einladung nicht geantwortet hat. Diesen Status muss der Benutzer einfach umändern können.

-need to have

* Funktionale Anforderung 11

Das Programm muss in der Lage sein, die Kontakte nach dem Status der Einladung (Akzeptiert/Abgelehnt/Nicht geantwortet) zu sortieren und anzuzeigen.

-need to have

* Funktionale Anforderung 12

Das Programm muss in der Lage sein, die Kunden nach dem Status der Einladung (Akzeptiert/Abgelehnt/Nicht geantwortet) zu exportieren.

-need to have

* Funktionale Anforderung 13

Das Programm muss in der Lage sein, die Anzahl der Kontakte anhand hinsichtlich Status der Einladung (Akzeptiert/Abgelehnt/Nicht geantwortet) anzuzeigen.

-need to have

* Funktionale Anforderung 14

Das Programm muss in der Lage sein, Daten in verschiedenen Formaten wie CSV, Excel, PDF, Word oder JPG zu exportieren.

-need to have

* Funktionale Anforderung 15

Das Programm muss in der Lage sein, die Daten in einer übersichtlichen Form, wie grafische Darstellung, anzuzeigen.

-need to have

* Funktionale Anforderung 16

Das Programm muss eine Suchmöglichkeit nach definierten Suchkriterien bieten.

-need to have

**Nicht-Funktionale Anforderungen**

* Nicht-Funktionale Anforderung 1

Das Programm soll in der Lage sein, doppelte Datensätze zu erkennen.

-nice to have

* Funktionale Anforderung 2

Das Programm soll in der Lage sein, alle Kontakte auf einer Seite anzeigen zu können.

-nice to have

* Funktionale Anforderung 3

Das Programm soll in der Lage sein, ein default Template (z.B. JKU-Daten) bei der Erstellung der Mails dem Benutzer anzuzeigen.

-nice to have

* Funktionale Anforderung 4

Das Programm soll in der Lage sein, beim Aussenden von Einladungen für ein Event automatisch dieses Event in einer bestehenden Facebook-Gruppe zu posten.

-nice to have

* Funktionale Anforderung 5

Das Programm soll in der Lage sein, Kunden, die nicht auf eine Einladung geantwortet haben, eine „Reminder“-Mail zu senden.

-nice to have

* Funktionale Anforderung 6

Das Programm soll in der Lage sein, die Suche anhand von Einschränken eines Suchbereiches zu spezifizieren. (z.B. Suche nur im Modul „Kontakte“ und nicht im Modul „Mails“)

-nice to have

* Funktionale Anforderung 7

Das Programm soll in der Lage sein, einen Benutzer bei durchgeführten Mailversand beziehungsweise bei einkommendem Mail zu benachrichtigen.

-nice to have

* Funktionale Anforderung 8

Das Programm soll in der Lage sein, einen Kalender bei der Terminverwaltung anzuzeigen.

-nice to have

* Funktionale Anforderung 9

Das Programm soll in der Lage sein, sich an den letzten Login zu erinnern beziehungsweise die Logindaten zu merken um eine erneute Eingabe der Logindaten zu vermeiden.

-nice to have

***Performance Anforderungen***

Das Programm muss in der Lage sein, bis zu 500 Kunden zu verwalten.

-need to have

Das Programm muss sowohl in der deutschen als auch englischen Sprache verfügbar sein.

-need to have

***Logische Datenbankanforderungen***

* Zugriffsrechte

Der Superadministrator soll für den reibungslosen Betrieb des Systems zuständig sein, sowie für die Verteilung der Rollen. Er soll in der Lage sein Benutzer zu löschen, Events zu erstellen, Kontakte zu erstellen und alle weiteren Funktionalitäten des Programms benutzen zu können.

Der Administrator erhält die gleichen Rechte wie der Superadministrator, außer der Möglichkeit die Einstellungen vom Programm zu ändern. Beziehungsweise er kann Rollen verteilen, Benutzer löschen, Events erstellen usw.

Der Editor (studentischer Mitarbeiter) erhält die gleichen Rechte wie der Administrator, außer der Möglichkeit Rollen zu verteilen, Benutzer zu löschen und Events zu erstellen. Beziehungsweise er kann Kontakte erstellen, Einladungen versenden usw.

Die studentische Hilfskraft erhält nur Leserechte im Programm beziehungsweise sie erhält die Möglichkeit zur Einsicht in die Kontaktelisten oder in die versendeten Einladungen.

* Zugriffshäufigkeit

Das Programm beziehungsweise die Datenbanken sollten jederzeit verfügbar sein.

* Datenintegrität

Es sollen zwei Datenbanken zur Verfügung stehen, in denen verschiedene Rollen zur Nutzung des Programms definiert sind.

-nice to have)

## Vorgehensmodell & Umsetzungsplanung

Unter einem Vorgehensmodell wird eine Folge von Aktivitäten und Ergebnissen verstanden, die zur erfolgreichen Durchführung eines Projekts erforderlich sind. Ergänzt wird ein Vorgehensmodell durch Prinzipien, Methoden, Techniken und Werkzeuge. Hauptziel eines Vorgehensmodells ist es, eine komplexe Problemstellung in leichter lösbare Teilaufgaben zu reduzieren und diese phasenweise zu strukturieren, um somit schneller und verlässlicher die angestrebten Projektziele zu erreichen.

Des Weiteren erlauben Vorgehensmodelle Erfahrungswissen zu explizieren und so eine weitgehende Personenunabhängigkeit zu erreichen. Die Modelle dürfen aber nicht zum Selbstzweck werden und sollten nicht im Widerspruch zu Intuition und Kreativität stehen. Vielmehr sind sie als Leitfaden für die Lösung praktischer Probleme zu verstehen.

**Planung der Umsetzung**

Im Folgendem werden die Schritte für die Planung der Umsetzung vorgestellt.

Auswahl der CRM-Software

In diesem Schritt wird die CRM-Software ausgewählt. Die Auswahl ist von den oben angeführten Anforderungen abhängig beziehungsweise es wird geschaut wie sehr die CRM-Software diese Anforderungen erfüllt.

Set-Up

In diesem Schritt mach man sich mit der CRM-Software, durch das Durchlesen des Benutzerhandbuches und den verschiedenen Tutorials, vertraut.

2.1. Konfiguration nach den Anforderungen

2.2 Testphase 1 mit Demodaten

3. Installation am Server

In diesem Schritt wird die CRM-Software am Zielort installiert.

3.1. On-premise Lösung

3.2. Implementieren notwendiger Module

4. Datenexport & -import

4.1. Export des Altsystems

4.2. Import der Daten ins Odoo

Hier werden auch die benötigten Daten (Kontaktdaten usw.) integriert.

5. Finale Konfiguration der erhobenen Anforderungen

In diesem Schritt wird die CRM-Software nach den need-to-have Anforderungen angepasst. Wenn möglich werden auch die nice-to-have Anforderungen angepasst.

5.1. Frist für das Erheben neuer Anforderungen

5.2. Testphase 2 mit Stakeholder

6. Optimieren zwecks Usability

6.1. übersichtliches Design

7. Test- & Reviewphase

7.1. Funktionale Anforderungen

In diesem Schritt werden alle Funktionalitäten der CRM-Software getestet.

7.2. Nicht funktionale Anforderungen

8. Abgleich umgesetzter Anforderungen

8.1. Zeitgleiche Dokumentation der Umsetzung

8.2. Fazit

9. Einschulungsphase mit Stakeholder

In diesem Schritt werden die Zielpersonen eingeschult.

9.1. Prozessmodellierung

9.1.1. Abstimmung der Stakeholder

9.2. Finale Konfiguration hinsichtlich Layouts & Design

10. Finale Datenmigration

10.1. Finaler Export

10.2. Import ins neue System

10.3. Ablösen des Altsystems

(Al-Mudimigh, Zairi, & Al-Mashari, 2001; Ehie & Madsen, 2005; Parr & Shanks, 2000; Peslak & Subramanian, 2008; Waring & Maddocks, 2005)

***Schulungskonzept***

In der weiteren Folge wird das Konzept der Schulung, die in Bezug auf die CRM Software „Odoo“ angewendet wird, näher erklärt.

**Aufbau der Schulung**

Der Aufbau der Schulung entwickelt sich aus einer theoretischen Inputphase zu immer größerer Aktivität und Beteiligung der Schulungsteilnehmer in den weiteren Phasen. Die Auswahl der Schulungsinhalte orientiert sich an der Vermittlung der Inhalte zur Anwendung und Umsetzung von „Odoo“. (Brem, 2019; Bucki & Pesqueux, 2002; Kleindienst, Koch, Ritz, & Brüngger, 2015)

**Teilnehmer**

Die Teilnehmer der Schulung sind: Frau Dr. Barbara Krumay vom Institut Information Engineering (JKU), Herr Dr. David Christoph Rückel vom Institut Information Engineering (JKU) und Herr Michael Holoubek vom Institut Information Engineering (JKU).

Die oben genannten Teilnehmer werden zusammen, am gleichen Ort und zur gleichen Zeit an der Schulung teilnehmen. Darüber hinaus wird Rücksicht auf die Rollen der Teilnehmer genommen. Beziehungsweise Teilnehmer, die sich mit Funktionen vertraut machen wollen, die nur für den Administrator gedacht worden sind, werden auch dementsprechend eingeschult.

**Werkzeuge**

Die Teilnehmer müssen ein geeignetes mobiles Endgerät (z.B. Laptop) bereithaben.

**Schulungstermine**

Für die Schulung ist nur ein Termin vorgesehen. Dieser findet am **<Termin>** statt.

**Schulungsschritte**

Im Rahmen der Schulung werden die folgenden Schritte durchgeführt:

* **Theoretischer Input**
* Präsentation der Ziele der Schulung
* Darstellung der Relevanz der Ziele
* Einen Überblick über die Lerninhalte vermitteln
* Die Vorgehensweise der weiteren Schulung erklären
* **Einführung in die CRM Software Odoo**
* Theoretischer Input zu Odoo (Allgemeine Informationen über die Software)
* Vorstellung vom Design
* Vorstellung der Vorteile beziehungsweise Nachteile von Odoo
* **Durchgehen der Funktionen zusammen mit den Teilnehmern**
* Schrittweise Vorstellung der Funktionen (Anlegen vom Event, Einsicht in die Kontaktlisten usw.) mithilfe vom Projektor, während die Teilnehmer die dargestellten Schritte nachmachen
* Den Teilnehmern Hilfe anbieten, falls sie Probleme bei Nachmachung der vorgestellten Schritte haben
* **Diskussion über die Leistung des Programms**
* Feedback entgegennehmen
* Über mögliche Fehler beziehungsweise Missverständnisse diskutieren

# Ergebnisse

# Tabellenverzeichnis

# Abbildungsverzeichnis

# Literaturverzeichnis

Al-Mudimigh, A., Zairi, M., & Al-Mashari, M. (2001). ERP software implementation: An integrative framework. *European Journal of Information Systems*, *10*(4), 216–226. https://doi.org/10.1057/palgrave.ejis.3000406

Brem, A. (2019). Creativity on Demand: How to Plan and Execute Successful Innovation Workshops. IEEE Engineering Management Review, 47(1), 94–98. https://doi.org/10.1109/EMR.2019.2896557

Bucki, J., & Pesqueux, Y. (2002). Flexible workshop: about the concept of flexibility. International Journal of Agile Management Systems, 2(1), 62–70. https://doi.org/10.1108/14654650010312615

Chen, I. J., & Popovich, K. (2003). Understanding customer relationship management (CRM): People, process and technology. *Business Process Management Journal*, *9*(5), 672–688. https://doi.org/10.1108/14637150310496758

Christopher, B. (2003). Strategic issues in customer relationship management (CRM) implementation. *Business Process Management Journal*, *9*(5), 592. Retrieved from http://proquest.umi.com/pqdweb?did=436780121&Fmt=7&clientId=43922&RQT=309&VName=PQD

Ehie, I. C., & Madsen, M. (2005). Identifying critical issues in enterprise resource planning (ERP) implementation. *Computers in Industry*, *56*(6), 545–557. https://doi.org/10.1016/j.compind.2005.02.006

Elliott, Sr., R. a., & Allen, E. B. (2013). A Methodology for Creating an IEEE Standard 830-1998 Software Requirements Specification Document. *Journal of Computing Sciences in Colleges*, *29*, 123–131. Retrieved from http://dl.acm.org/citation.cfm?id=2535418.2535437%5Cnhttp://dl.acm.org/ft\_gateway.cfm?id=2535437&type=pdf

Haefliger, S., von Krogh, G., & Spaeth, S. (2008). Code Reuse in Open Source Software. *Management Science*, *54*(1), 180–193. https://doi.org/10.1287/mnsc.1070.0748

Hung, P., & Low, G. C. (2008). Factors affecting the buy vs build decision in large Australian organisations. *Journal of Information Technology*, *23*(2), 118–131. https://doi.org/10.1057/palgrave.jit.2000098

Kleindienst, C., Koch, J., Ritz, F., & Brüngger, J. (2015). Förderung der Resilienz durch organisationales Lernen. Ein Schulungskonzept für Leitwartenteams in einem Kernkraftwerk. Wirtschaftspsychologie, (4), 54–61.

McIlroy, R. C., & Stanton, N. A. (2012). Specifying the requirements for requirements specification: The case for Work Domain and Worker Competencies Analyses. *Theoretical Issues in Ergonomics Science*, *13*(4), 450–471. https://doi.org/10.1080/1463922X.2010.539287

Nagle, F. (2015). Open Source Software and Firm Productivity. *Ssrn*, (March 2019). https://doi.org/10.2139/ssrn.2559957

Navarro, E., Letelier, P., Mocholi, J. A., & Ramos, I. (2018). A Metamodeling Approach for Requirements Specification. *Journal of Computer Information Systems*, *46*(5), 67–77. https://doi.org/10.1080/08874417.2006.11645925

Parr, A., & Shanks, G. (2000). A model of ERP project implementation. *Journal of Information Technology*, *15*(4), 289–303. https://doi.org/10.1080/02683960010009051

Peslak, A., & Subramanian, G. (2008). The phases of ERP software implementation and maintenance: A model for predicting preferred ERP use. *Journal of Computer*, *48*(2), 25–33. https://doi.org/10.1080/08874417.2008.11646006

Rathi, D., & Given, L. M. (2017). Non-profit organizations’ use of tools and technologies for knowledge management: a comparative study. *Journal of Knowledge Management*, *21*(4), 718–740. https://doi.org/10.1108/JKM-06-2016-0229

Standard, I. (2010). Systems and software engineering — Vocabulary INTERNATIONAL STANDARD ISO / IEC / IEEE 24765:2010, *2010*.

von Krogh, G., & von Hippel, E. (2006). The Promise of Research on Open Source Software. *Management Science*, *52*(7), 975–983. https://doi.org/10.1287/mnsc.1060.0560

Waring, T., & Maddocks, P. (2005). Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future. *International Journal of Information Management*, *25*(5), 411–428. https://doi.org/10.1016/j.ijinfomgt.2005.06.002

# Anhang