

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](#_Toc5123333)

[1.1. Einleitung 3](#_Toc5123334)

[1.2. Problembeschreibung und -nachweis 3](#_Toc5123335)

[1.3. Ziele 3](#_Toc5123336)

[1.4. Forschungsfragen 3](#_Toc5123337)

[2. Problemlösungsweg 3](#_Toc5123338)

[2.1. Customer-Relationship-Management 3](#_Toc5123339)

[2.2. Vorgehensmodell 4](#_Toc5123340)

[2.3. Phase 4](#_Toc5123341)

[2.4. Phase 4](#_Toc5123342)

[3. Ergebnisse 4](#_Toc5123343)

[4. Diskussion 4](#_Toc5123344)

[5. Tabellenverzeichnis 4](#_Toc5123345)

[6. Abbildungsverzeichnis 4](#_Toc5123346)

[7. Literaturverzeichnis 4](#_Toc5123347)

[8. Anhang 4](#_Toc5123348)

# Problem

## Einleitung

## Problembeschreibung und -nachweis

## Ziele

## Forschungsfragen

# Problemlösungsweg

## 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 (Bull, 2003). Eine andere Ansicht von CRM ist, dass es eine Technologielösung, 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 und Popovich, 2003).

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

Kunden profitieren von Einfachheit und Komfort für die Durchführung von Transaktionen (Chen und Popovich, 2003), sparen Zeit und Geld und bekommen bessere Informationen (Bull, 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 und 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. (Krogh und 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. (Krogh und Hippel, 2006)

Vorteile und Nachteile

Die Verwendung von Open Source Software ist mit verschiedenen Vorteilen bzw. 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, Krogh und Spaeth, 2007):

* 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, 2018)

## 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 und 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 und 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 bzw. zu binden und die Notwendigkeit wirksamer Programme zur Kontaktaufnahme mit den Sponsoren. Diese Herausforderungen grenzen die allgemeine Wirksamkeit und Effizienz der NPO ein. (Rathi und Given,2016)

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 und Given,2016)

* 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

## Anforderungskatalog

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 bzw. 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.

(Navaro et al. 2006, McLlroy et al. 2012)

### 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.

Zweck des Programms ist es grundlegende Funktionen einer CRM-Lösung für Kontakt- bzw. Eventmanagement zu bieten. Dies beinhaltet u.a. Informationen über Kunden zu sammeln, Einladungen bezüglich verschiedener Events herauszuschicken usw.

* + - 1. Allgemeine Beschreibung

#### 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.

#### Designeinschränkungen

* Das Programm muss auf folgenden operativen Systemen lauffähig sein: Windows, Linux und OSX
* Das Programm muss auf mobilen Endgeräten lauffähig sein
* Bezüglich der Benutzung dürfen keine Lizensprobleme auftreten
* Das Programm muss Open Source sein
* Das Programm muss ausbaubar sein
* Es darf kein cloudbasiertes System sein

#### Benutzereigenschaften

Insgesamt sollen zwei Arten von Benutzern: „Softwareadministratoren“ und „Editoren“ existieren.

Der Softwareadministrator soll für den reibungslosen Betrieb des Systems zuständig sein, sowie für die Verteilung der Rollen. Er soll in der Lage Benutzer zu löschen und alle weiteren Funktionalitäten des Programms benutzen zu können. Der Editor soll „nur“ Schreib- und Leserechte bekommen.

* + - 1. Spezifische Anforderungen

#### Im Folgenden werden die Funktionalitäten, welche das Programm haben soll, behandelt.

* Funktionelle Anforderung 1

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

-need to have

* Funktionelle Anforderung 2

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

-nice to have

* Funktionelle Anforderung 3

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

-nice to have

* Funktionelle Anforderung 4

Das Programm soll in der Lage sein, Zusatzinformationen zu den Kontakten bzw. Einladungen zu erstellen.

-need to have

* Funktionelle Anforderung 5

Das Programm soll in der Lage sein, die Informationen über die Kunden schnell ändern zu können. Bzw. man sollte nicht mehrere Fenster öffnen müssen um dies zu tun.

-need to have

* Funktionelle Anforderung 6

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

-need to have

* Funktionelle Anforderung 7

Das Programm soll in der Lage sein, bei dem hinzufügen von Events einen Kalender zu vorlegen, um das Datum leichter zu selektieren.

* Funktionelle Anforderung 8

Das Programm soll in der Lage sein, Mails bzw. Einladungen zu verschicken.

-need to have

* Funktionelle Anforderung 9

Das Programm soll in der Lage sein, Events bei dem Verschicken von den Einladungen automatisch in die Facebook-Gruppe zu posten.

-nice to have

* Funktionelle Anforderung 10

Das Programm soll in der Lage sein, vollautomatisiert Einladungen zu verschicken.

-need to have

* Funktionelle Anforderung 11

Das Programm soll in der Lage sein, Serienbriefe zu verschicken.

-need to have

* Funktionelle Anforderung 12

Das Programm soll in der Lage sein bei jedem Kunden einen Status erstellen zu können, wo man sieht, ob der Kunde die Einladung zu einem Event akzeptiert bzw. abgelehnt hat oder nicht geantwortet hat. Diesen Status soll man einfach umändern können.

-need to have

* Funktionelle Anforderung 13

Das Programm soll in der Lage sein, die Kunden nach dem Status der Einladung (bzw. Akzeptiert/Abgelehnt) zu sortieren.

-need to have

* Funktionelle Anforderung 14

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

-need to have

* Funktionelle Anforderung 15

Das Programm soll in der Lage sein, die Kunden anhand vom Status der Einladung (bzw. Akzeptiert/Abgelehnt) zu zählen.

-need to have

* Funktionelle Anforderung 16

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

-nice to have

* Funktionelle Anforderung 17

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

-need to have

* Funktionelle Anforderung 18

Das Programm soll in der Lage sein, Daten in verschiedenen Formaten (z.B. CSV, Excel, PDF, Word, JPG) zu exportieren.

-need to have

* Funktionelle Anforderung 19

Das Programm soll in der Lage sein, die Daten in einer übersichtlichen Form (z.B. Grafisch) darstellen zu können

-need to have

* Funktionelle Anforderung 20

Das Programm soll eine Suchmöglichkeit enthalten.

-need to have

* Funktionelle Anforderung 21

Das Programm soll in der Lage sein, die Suche zu spezifizieren (z.B. Suche nur in den Kontakten und nicht in den Mails).

-nice to have

* Funktionelle Anforderung 22

Das Programm soll in der Lage sein, eine Benachrichtigung anzuzeigen wenn eine Mail versendet wird bzw. bei eine neue Mail kommt.

-nice to have

* Funktionelle Anforderung 23

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

-nice to have

* Funktionelle Anforderung 24

Das Programm soll in der Lage sein, sich an den letzten Login zu erinnern bzw. die Daten sollte man nicht wieder beim Login eingeben müssen, weil sie automatisch erstellt werden.

-nice to have

* Funktionelle Anforderung 25

Das Programm soll in der Lage sein, eine Zeiterfassung zu führen.

-nice to have

## Szenarien & Faktoren zur Entscheidung

Im Zuge der durchgeführten Anforderungsanalyse sollen nun Szenarien für die jeweiligen Entscheidungen (Make or Buy) entwickelt werden. Intention ist es, mithilfe dieser Gegebenheiten einen Vorschlag mit Begründungen für die schwerfällige Entscheidung zu bringen. Zudem wird versucht einen Plan für die Umsetzung der jeweiligen Variante aufzustellen.

|  |  |  |
| --- | --- | --- |
| **Aspekte / Umsetzungsart** | **Use** | **Make** |
| *Aufwandstreiber* | - Programmieraufwand bis man zur Erkenntnis kommt ob möglich oder nicht | + Schnelle Recherche durch Nachforschen ob möglich oder nicht |
| *Komplexität* | -        (?) |  |
| *Flexibilität* | + Erweiterbar  ?Schnittstellen, Zusatzmodule | + Da Open Source, bleibt erweiterbar |
| *Dauer Erweiterung / Implementierung* | -         auf jeden Fall länger als MAKE | + |
| *Risiken*  -        *Während*  -        *Nach* | -        Einarbeitungszeit  -        Auswirkung der Erweiterung auf bestehende Fkt.  -        Nicht im Zeitrahmen umsetzbar  -        Ansprechperson nicht erreichbar (Studenten) | -        Komplikationen & Verschlechterung im Vergleich zur bestehenden Lösung  - |
| *Auswirkungen* | -        Nicht den Erwartungen entsprechend bzw. mangelhafte Umsetzung | -        Nicht angenommen von Stakeholdern |
| *Verbundene (nachträgliche) Aufgaben* | -        Testen  -        Einschulung / Aufklärung  -        Dokumentation verfassen | -        Testen  -        Einschulung  -        Datentransfer  - |

### MAKE

* Sollte die derzeitige CRM Software weiterentwickelt werden, müssten nicht allzu gravierende Veränderungen durchgeführt werden.
* Zudem erspart man sich viel Zeit für Installation und für die Anbindung an den Server sowie für die Einschulung ins neue System (Benutzer & Studenten).
* Das Positive am derzeitigen CRM Lösung ist, dass viele Anforderungen leicht im Internet durch paar Klicks abgeändert werden können.
* Es besteht aber die Gefahr, dass es gewisse Umstellungen nicht möglich sind bzw. viel Zeit beanspruchen.

### USE

Beim Anlegen einer neuen CRM bzw. Eventmanagement Lösung wird mehr Zeit aufgewendet, da vieles erst den Anforderungen angepasst werden muss und auch alle Daten verknüpft werden müssen. Der Zeitaufwand kann sich von einer Lösung zur anderen stark unterscheiden, da die Einarbeitungszeit unterschiedlich lang sein kann. Außerdem hat nicht jede Open Source Variante genug Dokumentation zur Installation und besonders nicht zu spezifischen Anpassungen (CiviCRM). Somit kann die ganze Implementierung viel länger dauern als sich mit dem derzeitigen zu beschäftigen und dies so Umstellen, dass es den Anforderungen gewachsen ist.

|  |  |  |
| --- | --- | --- |
| **need to have** | **CiviCRM** | **Odoo** |
| *1.Das Programm soll in der Lage sein,* ***Event*** *zu* ***hinzufügen****. Diesbezüglich werden folgende Informationen eingefügt: „Name“, „Datum“, „Ort“, „Veranstalter“, „Kommentar“.* | JA, geht es gibt sogar mehr Infos |  |
| *2.Das Programm soll in der Lage sein,* ***Kontakte******hinzu zufügen****. Diesbezüglich werden folgende Informationen eingefügt: „Titel“, „Vorname“, „Nachname“, „Email“, „Nummer“, „Adresse“, „DSGVO* | JA, geht es gibt sogar mehr Infos | JA geht, es gibt sogar mehr Infos |
| *3.Das Programm soll in der Lage sein, die* ***Informationen über die Kunden schnell*** *ändern zu können. Bzw. man sollte nicht mehrere Fenster öffnen müssen um dies zu tun.* | schnell nicht, ist aber machbar |  |
| 4.*Das Programm soll in der Lage sein,* ***Zusatzinformationen*** *zu den Kontakten bzw. Einladungen zu erstellen* | JA funktioniert |  |
| 5.*Das Programm soll in der Lage sein,* ***Mails*** *bzw.* ***Einladungen*** *zu* ***verschicken****.* | JA funktioniert |  |
| 6.*Das Programm soll in der Lage sein,* ***vollautomatisiert Einladungen*** *zu verschicken.* | JA funktioniert grundestzlich |  |
| 7.*Das Programm soll in der Lage sein,* ***Serienbriefe*** *zu verschicken.* | ? NEIN |  |
| 8.*Das Programm soll in der Lage sein bei jedem Kunden einen* ***Status*** *erstellen zu können, wo man sieht, ob der Kunde die Einladung zu einem Event akzeptiert bzw. abgelehnt hat oder nicht geantwortet hat. Diesen Status soll man einfach umändern können.* | sehr schlecht gemacht, keine Übersicht, funktioniert nur dauert lange |  |
| 9.*Das Programm soll in der Lage sein, die* ***Kunden*** *nach dem Status der Einladung (bzw. Akzeptiert/Abgelehnt) zu* ***sortieren****.* | JA funktioniert |  |
| 10.*Das Programm soll in der Lage sein, die* ***Kunden*** *anhand vom Status der Einladung (bzw. Akzeptiert/Abgelehnt) zu* ***zählen****.* | JA funktioniert |  |
| *11.Das Programm soll in der Lage sein, bis zu* ***500*** *Kunden zu verwalten.* | JA vorhanden |  |
| 12.*Das Programm soll in der Lage sein, die Kunden nach dem* ***Status der Einladung*** *(bzw. Akzeptiert/Abgelehnt) zu* ***exportieren***. | JA vorhanden |  |
| 13.*Das Programm soll in der Lage sein, Daten in verschiedenen Formaten (z.B. CSV, Excel, PDF, Word, JPG) zu* ***exportieren*** | JA vorhanden (PDF, CSV) |  |
| 14.*Das Programm soll in der Lage sein, die Daten in einer* ***übersichtlichen Form*** *(z.B. Grafisch) darstellen zu können.* | Übersicht nicht vorhanden |  |
| 15.*Das Programm soll eine* ***Suchmöglichkeit*** *enthalten.* | JA vorhanden, aber sehr komplex |  |

### Szenarien für USE

|  |  |  |
| --- | --- | --- |
|  | **CiviCRM** | **Odoo** |
| *Aufwandstreiber* | Laut CiviCRM kann man den Funktionsumfang manuell erweitern.   * ABER wenig Information zur selbständigen Anpassung am Code   pro Anforderung max 2 Personenstunden (hängt davon ab wie komplex die Anforderung ist) |  |
| *Komplexität* | Da CiviCRM sehr viele Funktionen anbietet kann die Erweiterung bzw. Anpassung sehr komplex sein. |  |
| *Flexibilität* | Man kann am SourceCode unternehmensspezifische Änderungen vornehmen. | Man kann weiter Funktionen installieren. |
| *Dauer* | Installation & Datenimport: 1 Tag + Anpassungen am Programm: 1-2 Wochen (1 Woche intensiv) | Installation & Datenimport: 1 Tag + Anpassungen am Programm: 1-2 Wochen (1 Woche intensiv) |
| *Auswirkungen* | Funktioniert nur in Verbindung mit Drupal, WordPress oder Joomla. |  |
| *Risiken* | Nicht alle Anforderungen zu 100% funktionsfähig,  langsam, kann zu kompliziert wirken, schlechte usability |  |
| ANMERKUNGEN | NICHT auf mobilen Endgeräten nutzbar, kein gutes Design | auf iOS & Android lauffähig |

### Planung für die Umsetzung:

1. Installation am Server: Herunterladen der benötigten Dateien und am Web Server anpassen; Installieren der CRM Lösung
2. Datenimport: Kontakte und weiter wichtige Daten importieren
3. Sich mit dem System auseinandersetzen/vertraut machen: Benutzerhandbuch (wenn vorhanden) durchlesen, Tutorials anschauen
4. Konfigurieren nach Anforderungen: System nach need-to-have Anforderungen anpassen und wenn möglich nice-to-have Features hinzufügen
5. Optimieren zwecks Usability: Design benutzerfreundlich gestalten, wichtige Funktionen leicht auffindbar machen
6. Testen: Überprüfen aller benötigten Funktionen
7. Einschulung mittels eines Workshops: Benutzerhandbuch schreiben, alle Funktionalitäten den Benutzern/Administratoren zeigen und vermitteln

## FAZIT:

**CiviCRM** bietet einen großen Umfang an Funktionen, die aber sehr versteckt sind. Bei manchen Funktionen gibt es zu viele Informationen, was sehr unübersichtlich wirkt. Man muss oft mehrere Seiten öffnen um beispielsweise zur Übersicht aller Events zu kommen. Da eben die Effizienz ein wichtiger Punkt für eine erfolgreiches CRM System ist, ist dies der größte Nachteil für die Implementierung von CiviCRM. Zudem ist es in Sachen Design zurückgeblieben. Im Vergleich zu dem derzeitigen CRM System bietet CiviCRM viel mehr Möglichkeiten Informationen zu speichern, die aber nicht notwendig sind bzw. eher störend als nützlich sind.

Im Vergleich zu Odoo CRM hat CiviCRM keine mobile Anwendung und kann derzeit nur auf drei Content-Management-System installiert werden. Auch im Bereich Design und Usability ist Odoo im Vorsprung.

Aus diesen Gründen wäre es sinnvoller den ganzen Aufwand in die derzeitige (SuiteCRM) zu stecken anstatt eine neue CRM Lösung zu installieren, die nicht allen Anforderungen entspricht und weitere manuelle Anpassungen benötigt. Natürlich ist dies nur dann der Fall, wenn am derzeitigen CRM alle fehlenden Anforderungen gedeckt werden.

Szenarien für jeweilige CRM-Lösung

## Vorgehensmodell

## Phase

## Phase

# Ergebnisse

# Diskussion

# Tabellenverzeichnis

# Abbildungsverzeichnis

# Literaturverzeichnis

G. Von Krogh and E. Von Hippel, “The Promise of Research on Open Source Software The Promise of Research on Open Source Software,” no. March 2019, 2006.

S. Haefliger, G. von Krogh, and S. Spaeth, “Code Reuse in Open Source Software,” *Manage. Sci.*, vol. 54, no. 1, pp. 180–193, 2007.

F. Nagle, “Open Source Software and Firm Productivity,” no. March 2019, 2018.

P. Hung and G. C. Low, “Factors affecting the buy vs build decision in large Australian organisations,” *J. Inf. Technol.*, vol. 23, no. 2, pp. 118–131, 2008.

D. Rathi and L. M. Given, “Non Profit Organizations ’ Use of Tools and Technologies for Knowledge Management : A Comparative Study,” 2016.

Bull, C. (2003). Strategic issues in customer relationship management (CRM) implementation. Business Process Management Journal, 9, 592-602.

Chen, I., & Popovich, K., (2013). Understanding customer relationship management (CRM). Business Process Management Journal, 9, 672-688.

# Anhang