

Eingereicht von

**Dusanic Maja**

**K01256561**

**Ljutic Anel**

**K01640349**

**Stojcevic Dejan**

**K01609062**

**Samardzic Ivan**

**K01623903**

**Tomic Milos**

**K01356229**

Angefertigt am 24.10.2018

**Institut für Wirtschaftsinformatik – Information Engineering**



Betreuer / Betreuerin

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

**Drin. Barbara Krumay**

**Dr. David Rückel**

Partnerunternehmen

**ReqPOOL GmbH**

Oktober 2018



Seminararbeit

im Bachelorstudium

Wirtschaftsinformatik

LVA-Nr.: 256.005; Information Engineering PS

Software-eigenentwicklungen in Österreich

**Integration von Software-Eigenentwicklungen in eine Applikationslandkarte**

Inhaltsverzeichnis

[1. Kapitel 3](#_Toc509819582)

[1.1. Unterkapitel 3](#_Toc509819583)

[1.2. Unterkapitel 3](#_Toc509819584)

[1.2.1. Unterkapitel 3](#_Toc509819585)

[1.2.2. Unterkapitel 3](#_Toc509819586)

**Hinweise zu Listen mit Aufzählungspunkten**

Wenn Sie eine Liste mit Aufzählungspunkten erstellen möchten, nutzen Sie bitte die Vorlage „Liste JKU“. Zu finden unter: Start – Absatz – Liste mit mehreren Ebenen – Listenformatvorlagen – Liste JKU.

* Ebene 1
  + Ebene 2
    - Ebene 3
      * Ebene 4
        + Ebene 5

**Hinweise zu nummerierten Überschriften**

Wenn Sie eine Liste mit nummerierten Überschriften erstellen möchten, nutzen Sie bitte die Vorlage „Liste JKU“. Zu finden unter: Start – Absatz – Liste mit mehreren Ebenen – Listenformatvorlagen – Überschriften JKU.

# Kapitel

## Unterkapitel

## Unterkapitel

### Unterkapitel

### Unterkapitel

#### Unterkapitel

#### Unterkapitel

**Hinweise zur Fußzeile**

Datum: Beim Öffnen des Dokuments wird automatisch das aktuelle Datum angezeigt.

**Hinweise zu Tabellen**

Tabelle JKU

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabelle 1: JKU

**Hinweise zur Struktur der Arbeit**

# Problem

## Einleitung

Bei den meisten Software Implementierungen stellt man sich am Anfang die Frage, ob die Software innerhalb des Unternehmens entwickeln werden sollte oder, ob man sie bei einem Drittunternehmen kaufen sollte. Diese „Make or Buy“ Entscheidung ist, besonders bei der ersten Auseinandersetzung, sehr fordernd, weil verschiedene Faktoren in Betracht gezogen werden müssen. Die Faktoren können von Unternehmen zu Unternehmen unterschiedlich sein. Dementsprechend sind Informationen über die einflussreichsten Faktoren eines jeweiligen Unternehmens sehr wichtig.

## Problembeschreibung und -nachweis

In Bezug auf Österreich sind diese Informationen nur teilweise existierend, weil das Wissen über eigenentwickelte Software nur gering ist. Auch wenn man davon ausgehen kann, dass in den letzten 40 bis 50 Jahren österreichische Unternehmen eine große Menge an Software entwickelt haben und seitdem betreiben, wurden diese eigenentwickelten Individuallösungen nur selten untersucht und dokumentiert. Für IT-Beratungsunternehmen wie beispielsweise ReqPOOL sind diese Informationen von wichtiger Relevanz, da sie Unternehmen beim Beschaffen von Softwarelösungen fachlich unterstützen können. Während Verkaufszahlen und Statistiken zu Standard-Softwarelösungen vorhanden sind, wurde konkret im Bereich der eigenentwickelten Softwarelösungen in Österreichs Unternehmen nur mangelhaft beziehungsweise gar nicht geforscht.

## Ziele

Ziel dieser Seminararbeit ist es, ein Erhebungsinstrument (Fragebogen) zu entwickeln, das ermöglicht, den Status Quo hinsichtlich eigenentwickelten Softwarelösungen in den 1000 wichtigsten Unternehmen in Österreich zu ermitteln.

## Forschungsfrage(n)

*„Wie sind existierende Software-Eigenentwicklungen in den größten Unternehmen Österreichs hinsichtlich unserer identifizierten Parameter beschaffen?“*

# Problemlösungsweg

## Vorgehensmodell

### Aktionsforschung

Es wird ein Praxisproblem durch einen gemischten Kreis aus Wissenschaft und Praxis gelöst. Hierbei werden mehrere Zyklen aus Analyse-, Aktions-, und Evaluationsschritten durchlaufen, die jeweils gering strukturierte Instrumente wie Gruppendiskussionen oder Planspiele vorsehen. [1]

Da in dieser Seminararbeit ein praxisnahes Thema bearbeitet wird und als Problemlösungsweg eine aktive Umfrage mittels des erstellten Erhebungstools vorhergesehen ist, bietet sich die Aktionsforschung als wissenschaftliche Vorgehensmethodik sehr gut an. Analyse- beziehungsweise Recherchezyklen werden im Laufe der Auseinandersetzung mit der Themenstellung eher am Beginn der Arbeit durchlaufen, jedoch werden auch fortlaufend Analysetätigkeiten notwendig sein. Als aktiver Teil der Vorgehensmethodik wird die Erstellung des Erhebungstools, die durchzuführende Umfrage mittels des Tools bei ca. zehn Unternehmen und in der finalen Phase das Erstellen einer allgemeingültigen Auswertungsmethode des erstellten Erhebungstools betrachtet. Hinsichtlich der Evaluationsphasen werden vor allem die Evaluierung der durchgeführten Umfrage sowie die daraus erhobenen Erkenntnisse im Mittelpunkt stehen. Bei wöchentlichen Gruppenmeetings werden zu bearbeitende Aufgaben an die jeweiligen Teammitglieder verteilt, erhobenes Wissen in der Gruppe geteilt als auch nächste Arbeitsschritte diskutiert. Alle zwei bis drei Wochen wird auch unsere Ansprechperson des Kooperationsunternehmens in diese Gruppenmeetings involviert um vor allem sicherzustellen, dass die Arbeits- und Denkweise in die entsprechende Richtung verläuft.

Dieses Vorgehensmodell bietet sich für unser Seminarthema sehr gut an, da das im Vorhinein recherchierte Wissen mit Erkenntnissen aus der Praxis, welche mittels der Umfrage als auch durch den Kontakt mit unserem Kooperationspartner erhoben werden, verglichen und diskutiert wird.

## Phase M (Inhaltliche Aufteilung)

## Recherche zu den Hauptaspekten

Um das Ergebungsinstrument zu entwickeln wurden verschiedene Parameter herangezogen (wie z.B. Art des Systems, Architektur, Programmiersprache, Bewertung, Umfang, Kosten, Einbettung in die Systemlandschaft, Alter, geplanter Wechsel und Weiterentwicklung). Aus diesen Parametern wurde versucht die möglichen Fragen, die das Erhebungsinstrument beinhalten soll, herauszulösen.

Während der Recherche über die unterschiedlichen Parameter von Softwarelösungen, haben sich die folgenden zwei Aspekte, bezüglich der Softwareeigenentwicklung, als die wichtigsten herausgestellt: Make Or Buy – Entscheidungen und die Unterscheidung zwischen Individual- und Standardsoftware.

### Make Or Buy von Softwarelösungen

Die Anwesenheit von Software in Computergeräten, Mobiltelefonen, tragbaren Geräten und Phablets hat Software zu einer extrem wichtigen Instanz für die Durchführung der Ereignisse des Lebens gemacht, und die Entwicklung von Software hat daher im Laufe der Zeit zugenommen. Es wurde festgestellt, dass die Hersteller und Konsumenten von Softwareanwendungen zugenommen haben und Softwareanwendungen rationaler an die Anforderungen angepasst worden sind. [2]

In einer typischen Entwicklungsumgebung, in der Verfahren, Standards und Teamstrukturen befolgt werden, ist die Entwicklung von Software eine kostenintensive Tätigkeit. Die Unternehmen, die keine geplante Zeit und notwendige finanzielle Ressourcen aufwenden, bringen Softwarerisiken in die Entwicklung ein und erhöhen die finanziellen Verbindlichkeiten und Verluste. Das Entwickeln von Software ist jedoch eine zeit- und kostenintensive Aktivität, für die viele andere Ressourcen entsprechend den Anforderungen der Entwicklungsaktivität platziert werden müssen. [2]

Die Eigenentwicklung sowie der Fremdbezug von Software Lösungen weisen einige Vor- und Nachteile auf, die Unternehmen dazu veranlassen, Make-or-Buy-Entscheidungen sorgfältig vorzubereiten. Dementsprechend ist die Entscheidung zwischen Eigenentwicklung oder Fremdbezug nicht leicht zu treffen, weil eine Reihe von Faktoren berücksichtigt werden müssen. [2], [3]

Die 10 wichtigsten Faktoren nach [2]:

* Strategie und Wettbewerbsvorteil
* Kosten
* Umfang und Komplexität
* Entsprechung der Anforderungen
* Entwicklungszeit
* Erfahrung in IS (Information Systems)
* Risiko
* Betriebliche Faktoren bzw. Produktions-Faktoren
* Geistiges Eigentum (Ideen, Patente)
* Altsystem

Unter Berücksichtigung der oben genannten Faktoren wurden verschiedene Vorteile bzw. Nachteile bezüglich der Eigenentwicklung einer Software identifiziert.

Die wichtigsten Vorteile der Eigenentwicklung einer Software nach [2][3] sind:

* Bewahrung vom eigenen IT-Know-How
* Sicherstellung des Datenschutzes und der Datensicherheit
* Fehlende Lizenskosten
* Keine Abhängigkeiten von Fremdfirmen
* Komplette Kontrolle über den Quellcode
* Komplette Kontrolle über den Entwicklungsplan

Die wichtigsten Nachteile der Eigenentwicklung einer Software nach [2][3] sind:

* Hohe Belastung der IT-Abteilung
* Lange Projektlaufzeiten
* Zusätzliche Schulungskosten der Mitarbeiter
* Abhängigkeiten vom Fremdunternehmen
* Geringen Kontrollmöglichkeiten
* Gefahr des Missbrauchs von internen Unternehmensdaten
* Ständiges Engagement der Mitarbeiter
* Geringere Funktionalität
* Updates sind schwieriger durchzuführen

### Individual- vs Standardsoftware

Unter Individualsoftware versteht man eine Software, die durch die eigene IT-Abteilung oder durch beauftragte Programmierer für eine spezielle betriebliche Aufgabenstellung im Unternehmen entwickelt wird. Als Standardsoftware werden Softwaresysteme verstanden, die einen klar definierten Anwendungsbereich abdecken und als vorgefertigte Produkte erworben werden können.[4]

s

Vorteile einer Individualsoftware im Unterschied zur Standardsoftware nach [4]:

* Wird spezifisch für verschiedene Anwendungsbereiche entwickelt
* Keine Abhängigkeit vom Softwarelieferanten
* Reduziertes Risiko von Datendiebstahl

Nachteile einer Individualsoftware im Unterschied zur Standardsoftware nach [4]:

* Höhere Kosten
* Längere Einführungsdauer
* Mehr auftretende Fehler
* Fehlende professionelle Arbeitskraft
* IT-Ressourcen werden stark ausgeschöpft

Im Weiterverlauf liegt der Fokus auf der Individualsoftware. Diese Entscheidung wurde in Absprache mit dem Kooperationsunternehmen (reqPool) gefällt.

## Erstellung des Fragebogens / Erhebungsinstruments

Der Inhalt des Fragebogens bzw. die Erstellung der Fragen basiert auf den gefundenen Ergebnissen der Recherche, die im Kapitel 2.2 näher beschrieben wurde. Weiters wurde dies durch die Kooperation des Projektpartners beeinflusst, durch welche einige Fragen dementsprechend angepasst worden sind. Im Folgenden wird die Struktur des Fragebogens und die dazu gehörenden Fragen genauer beschrieben.

### Struktur des Fragebogens

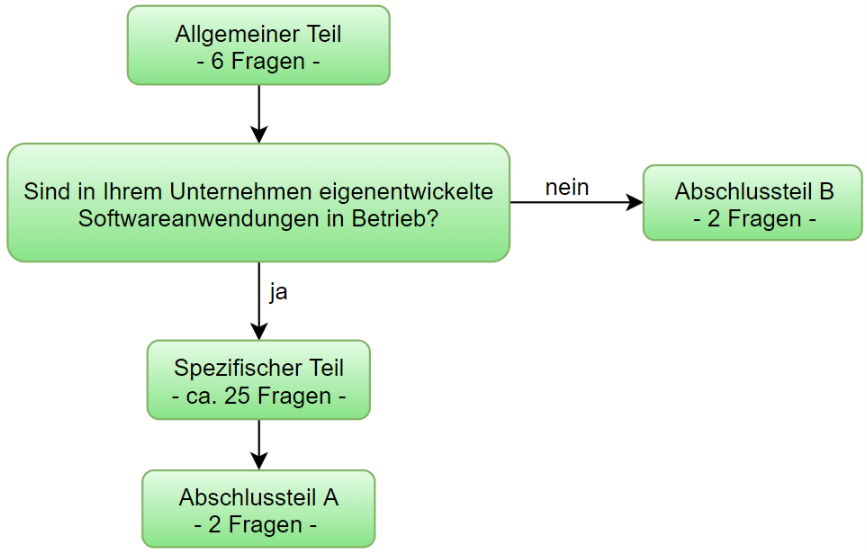


Abbildung 1: Struktur des Fragebogens

Wie in der Abbildung 1 gezeigt wird, besteht die Struktur des Fragebogens aus vier Bereichen:

* Allgemeiner Teil
* Spezifischer Teil
* Abschlussteil A
* Abschlussteil B
  + - 1. Allgemeiner Teil

Der allgemeine Teil besteht aus 6 Fragen. Diese Fragen dienen dazu mehr über das befragte Unternehmen bzw. die Ansprechperson herauszufinden. Der Zweck von diesem Teil ist es mehr Informationen u.a. über folgende Punkte zu bekommen:

* Die Position der Ansprechperson im Unternehmen
* Mitarbeiteranzahl in der IT-Abteilung
* Anzahl der zuständigen IT-Mitarbeiter für die Neuentwicklung/Instandhaltung
* Branche des Unternehmens

Die angeführten Branchen wurden unter der Verwendung von [5] ÖNACE-2008 Klassifikation erstellt und angepasst.

Im Anschluss vom allgemeinen Teil wird die folgende Frage gestellt: „Sind in Ihrem Unternehmen eigenentwickelte Softwareanwendungen in Betrieb?“ Diese Frage stellt eine Verzweigungsmöglichkeit dar. Abhängig von der Antwort kann der Befragte bei zwei verschiedenen Teilen des Fragebogens weitermachen.

Falls der Befragte positiv auf diese Frage geantwortet hat, wird er zum spezifischen Teil weitergeleitet. Falls er negativ auf diese Frage geantwortet hat, wird der Befragte den spezifischen Teil überspringen und einige Fragen über die eigene benutzte Software beantworten.

* + - 1. Spezifischer Teil

Der spezifische Teil besteht aus ca. 25 Fragen. Dieser Teil steht nur den Befragten zur Verfügung, die positiv auf die Frage beantwortet haben, ob sie im Besitz einer eigenentwickelten Software sind. Die Fragen im spezifischen Teil dienen dazu mehr Informationen über die eigenentwickelte Software herauszufinden.

Bei der Erstellung vom spezifischen Teil wurde zwischen zwei Optionen ausgewählt. Bei der ersten Option wurde der spezifische Teil in zwei Bereiche getrennt. Der erste Bereich bestand aus 5 Fragen, deren Zweck es war Informationen über die Implementierungszeit, Programmiersprache etc. zu sammeln. Der zweite Bereich behandelte ca. 15 Fragen. Der Zweck vom zweiten Bereich war es genauere Informationen über die eigenentwickelte Software herauszufinden (wie z.B. Architektur, Konnektoren etc.).

Der Sinn der ersten Option bestand darin, Informationen über mehrere eigenentwickelte Softwaren zu sammeln. Falls das Unternehmen mehr als nur eine eigenentwickelte Software im Besitz hat, würden alle durch den ersten Bereich behandelt werden, wobei nur eine von ihnen bei dem zweiten Bereich ankommen würde. Die Auswahl der Software für den zweiten Bereich würde nach der Priorisierung vorgehen, die im Kapitel 2.4.1.2.a. näher beschrieben wurde. Dieser Bereich würde im Anschluss genauere Informationen über diese eine eigenentwickelte Software sammeln. Es hätten dabei maximal fünf Softwaren behandelt werden können.

Die zweite Option bestand darin, die Trennung vom spezifischen Teil zu vermeiden. Dabei hätte nur eine eigenentwickelte Software behandelt werden können. Die Auswahl würde nach der Priorisierung im Kapitel 2.4.1.2.a. erfolgen.

Nach einer längeren internen Gruppendiskussion und der Kooperation mit dem Projektpartner wurde für die zweite Option entschieden. Grund für dies war die Gefahr der Verfälschung von Informationen, die die erste Option hätte verursachen können. Mit der Verfälschung von Informationen wird die Angabe falscher Informationen gemeint, weil der Befragte einer längeren Beantwortungszeit ausgesetzt wäre. Dies würde ein Desinteresse/Langeweile beim Befragten verursachen, weil er durch den ersten Bereich mehrere male durchgehen müsste bzw. dieselben Fragen für verschiedene eigenentwickelte Softwaren mehrmals beantworten müsste.

* + - * 1. Priorisierung

Nach der durchgeführten Recherche und der Kooperation des Projektpartners sind folgende Prioritäten entstanden:

* Größe (Anschaffungs- und Laufkosten)
* Alter (Implementierungszeit)
* Ablösegedanken

Diese Prioritäten wurden im Anschluss auf eine abstraktere Sicht gekapselt und durch die folgende Priorität ersetzt: „fachlich wirtschaftlich höchster Stellenwert.“

Die Fragen im spezifischen Teil beziehen sich alle auf die eigenentwickelte Software Anwendung, die diese Priorität bzw. Priorisierung erfüllt.

* + - * 1. Zweck

Der Zweck vom spezifischen Teil ist es mehr Informationen u.a. über folgende Punkte zu bekommen:

* Implementierungszeitraum

Die Informationen über den Implementierungszeitraum geben Auskunft über die technologische Umgebung der eigenentwickelten Software, während die Implementierung stattgefunden hat. Dies beeinflusst teilweise die Sicht auf die Faktoren wie z.B. Programmiersprache, Konnektoren, Architektur usw.

Diese Informationen werden durch die Frage in der folgenden Abbildung gesammelt:

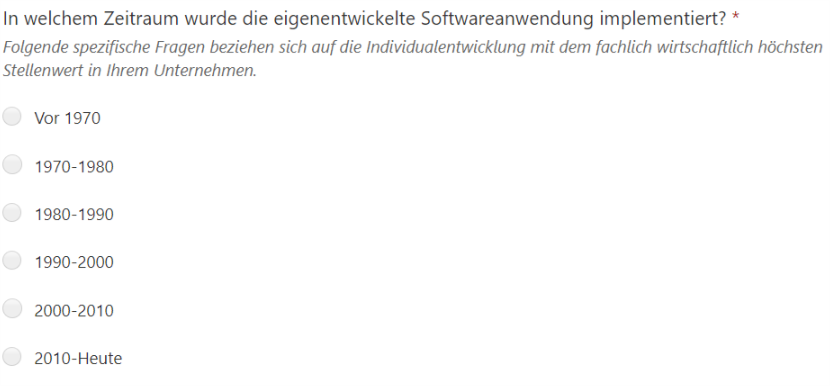


Abbildung 2: Implementierungszeitraum-Frage

Die angeführten Zeiträume in Abbildung 2 wurden durch die Kooperation mit dem Projektpartner und mit dem Software Engineering Institut an der Johannes Kepler Universität ausgewählt.

* Einführungskosten

Die Informationen über die Einführungskosten geben Auskunft über die Größe der eigenentwickelten Software. Dies beeinflusst teilweise die Sicht auf die Faktoren wie z.B. Kosten für die Aufrechterhaltung, Konnektoren usw.

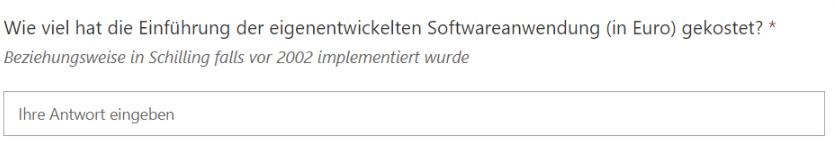
Diese Informationen werden durch die Frage in der folgenden Abbildung gesammelt:

Abbildung 3: Einführungskosten-Frage

* Unterstützte Kernprozesse durch die eigenentwickelte Software

Die Informationen über die unterstützen Kernprozesse durch die eigenentwickelte Software geben Auskunft über wirtschaftliche bzw. fachliche Bedeutung dieser Software für das Unternehmen. Dies beeinflusst teilweise die Sicht auf die Faktoren wie z.B. Kosten für die Aufrechterhaltung, Ablösegedanken usw.

Diese Informationen werden durch die Frage in der folgenden Abbildung gesammelt:

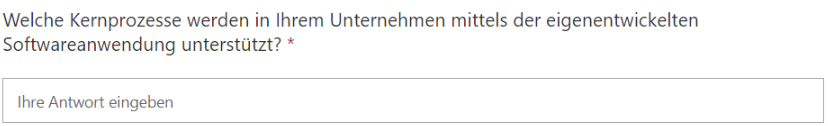


Abbildung 4: Unterstütze Kernprozess-Frage

* Programmiersprache

Die Informationen über die Programmiersprache(n) geben Auskunft über die verwendete(n) Programmiersprache(n) der eigenentwickelten Software. Es werden nur Programmiersprachen in Betracht gezogen, dessen Gebrauch für mehr als 10% vom Quellcode stattgefunden hat. Dies beeinflusst teilweise die Sicht auf die Faktoren wie z.B. Architektur, Konnektoren usw.

Diese Informationen werden durch die Frage in der folgenden Abbildung gesammelt:

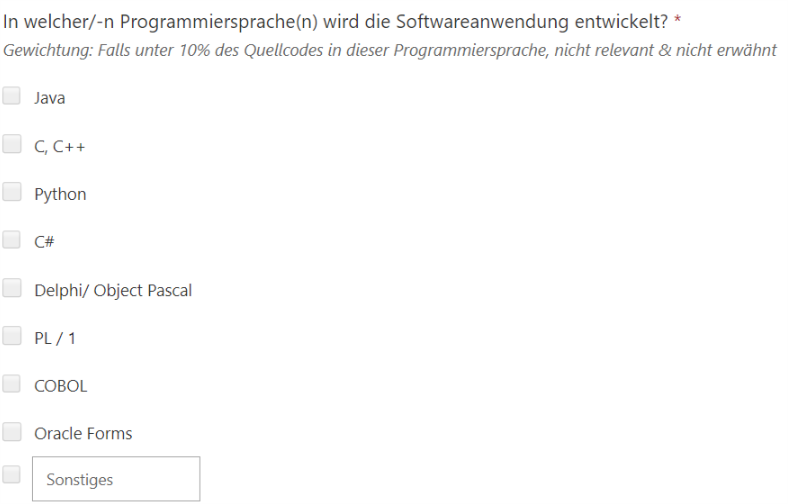


Abbildung 5: Programmiersprache - Frage

Die angeführten Programmiersprachen in Abbildung 5 wurden durch die Kooperation mit dem Projektpartner, mit dem Software Engineering Institut an der Johannes Kepler Universität und nach [6] TIOBE.com ausgewählt.

* Architektur (Art, Komponenten, Middleware, Datenhaltung)

Die Informationen über die Architektur der eigenentwickelten Software geben Auskunft über die Komplexität der Software bzw. Abhängigkeiten der Schichten, Typ von Clients, Anzahl von Back-End Komponenten.

Die Informationen über die verschiedenen Architekturteile wurden durch die Fragen gesammelt, die den folgenden Abbildungen zu entnehmen sind:

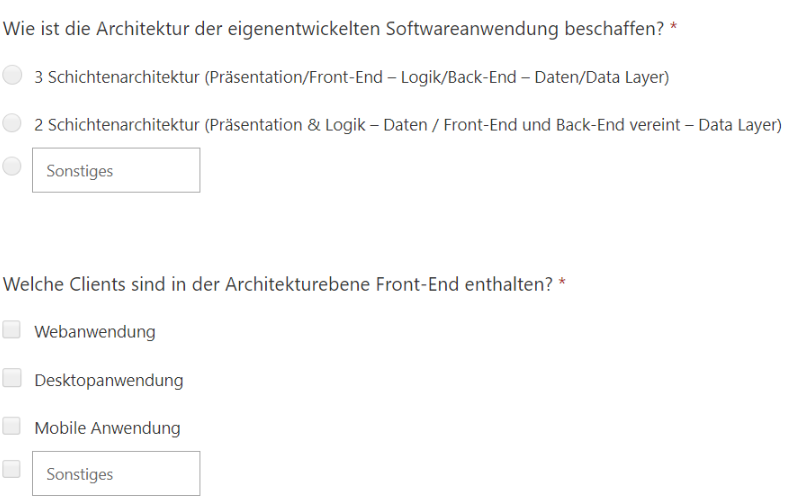


Abbildung 6: Architektur

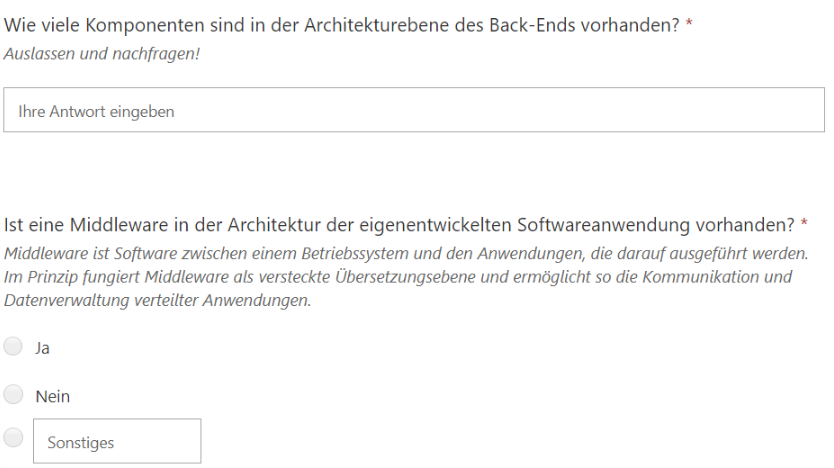


Abbildung 7: Architektur

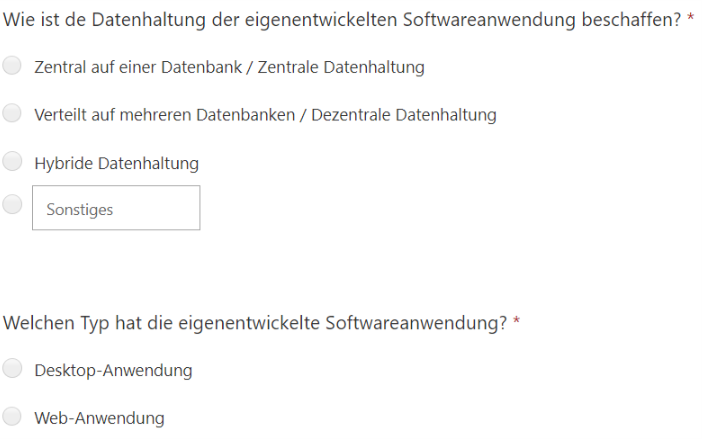


Abbildung 8: Architektur

Die angeführten Optionen in den Abbildungen 6 bis 8 wurden durch die Kooperation mit dem Projektpartner und mit dem Software Engineering Institut an der Johannes Kepler Universität ausgewählt.

* Typ

Die Informationen über den Typ geben Auskunft über die Art der eigenentwickelten Software. Dies beeinflusst teilweise die Sicht auf z.B. die Architektur.

Diese Informationen werden durch die Frage in der folgenden Abbildung gesammelt:

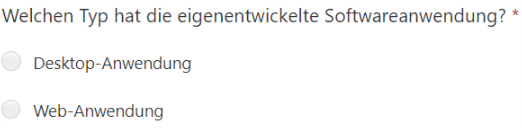


Abbildung 9: Typ

Die angeführten Arten wurden durch die Kooperation mit dem Projektpartner ausgewählt.

* Konnektoren

Die Informationen über die Konnektoren geben Auskunft über die verwendeten Schnittstellen der eigenentwickelten Software. Dies beeinflusst die Sicht auf die Komplexität der Software. Es werden Informationen über die Art der Konnektoren, ihre Anzahl und dem unterstützen Format der Rückgabewerte gesammelt.

Diese Informationen werden durch die Fragen in den folgenden Abbildungen gesammelt:

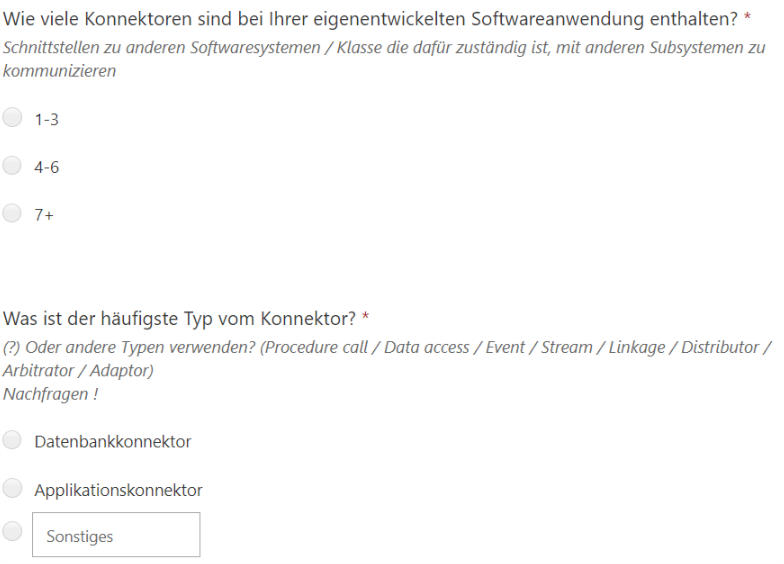


Abbildung 10: Konnektoren

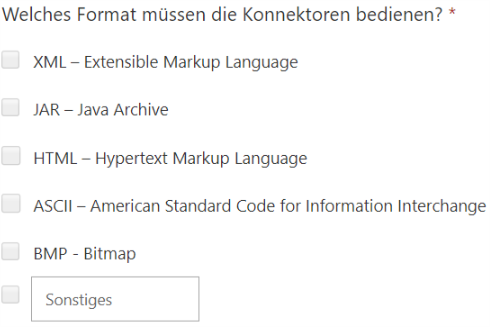


Abbildung 11: Konnektoren

Die angeführten Optionen in den Abbildungen 10 und 11 wurden durch die Kooperation mit dem Projektpartner und mit dem Software Engineering Institut an der Johannes Kepler Universität ausgewählt.

* Dokumentation

Die Informationen über die Dokumentation geben Auskunft über das existieren der Dokumentation vom Quellencode und die Art wie diese zu Stande gekommen ist. Dies beeinflusst die Sicht auf die Upgrades, die Probleme und die Ablösegedanken.

Diese Informationen werden durch die Fragen in der folgenden Abbildung gesammelt:

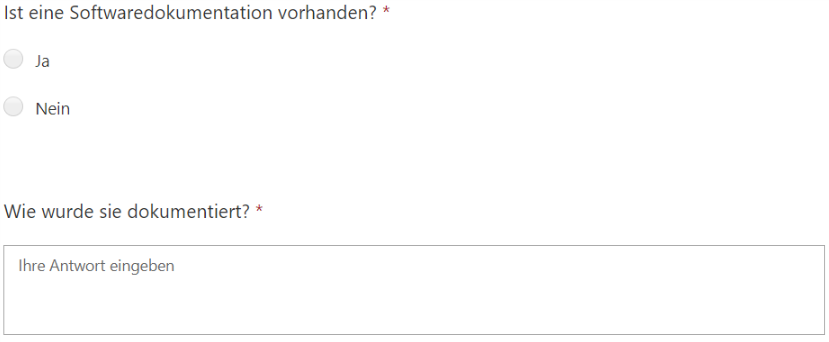


Abbildung 12: Dokumentation

* Aufrechterhaltungskosten

Die Informationen über die Aufrechterhaltungskosten geben Auskunft über ihre Höhe im letzten Jahr und die Faktoren, die den größten Beitrag dazu geleistet haben. Dies beeinflusst die Sicht auf die Upgrades, Probleme usw.

Diese Informationen werden durch die Fragen in den folgenden Abbildungen gesammelt:

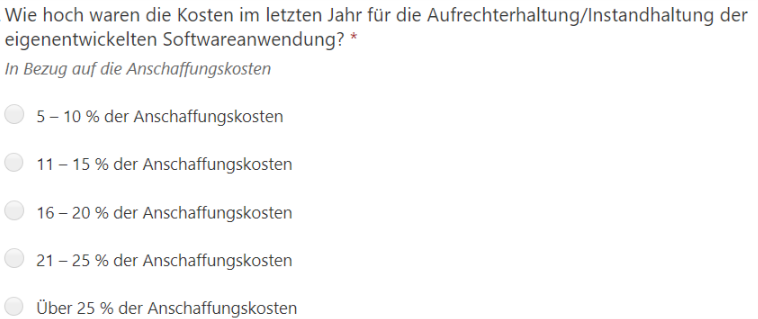


Abbildung 13: Aufrechterhaltungskosten

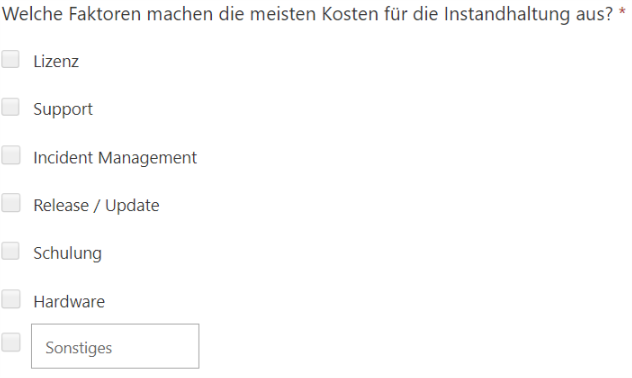


Abbildung 14: Aufrechterhaltungskosten

Die angeführten Optionen wurden durch die Kooperation mit dem Projektpartner ausgewählt.

* Upgrades

Die Informationen über die Upgrades geben Auskunft über ihre Anzahl seit der Implementierung, wann die letzte war und über die wichtigsten Änderungen. Dies beeinflusst die Sicht auf die Dokumentation, Ablösegedanken usw.

Diese Informationen werden durch die Fragen in den folgenden Abbildungen gesammelt:

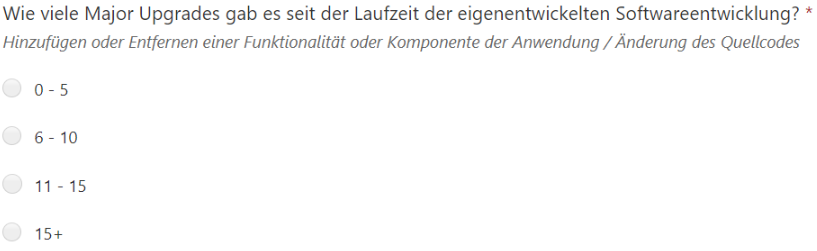


Abbildung 15: Upgrades

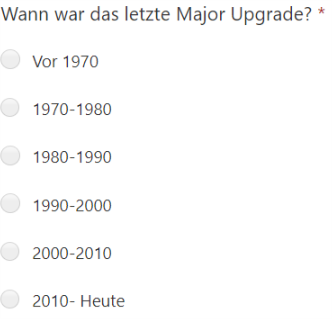


Abbildung 16: Upgrades

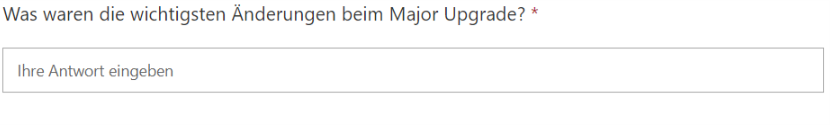


Abbildung 17: Upgrades

* Probleme

Die Informationen über die Probleme geben Auskunft über die Existenz von Problemen bezütlich der eigenentwickelten Software und dem Bereich wo sie entstanden worden sind. Dies beeinflusst die Sicht auf die Upgrades, Ablösegedanken usw.

Diese Informationen werden durch die Fragen in den folgenden Abbildungen gesammelt:

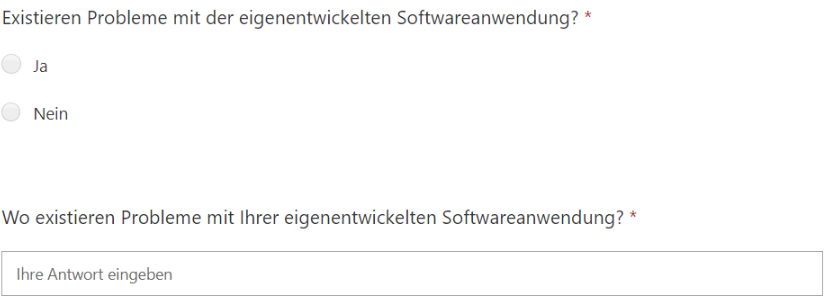


Abbildung 18: Probleme

Bei der zweiten Frage, die in der Abbildung 18 zu sehen ist, wurde zwischen einer Frage mit einer offenen Natur und einer Frage mit mehreren Auswahlmöglichkeiten ausgewählt. Nach einer längeren internen Gruppendiskussion und der Kooperation mit dem Software Engineering Institut an der Johannes Kepler Universität wurde für eine offene Frage entschieden. Grund für dies war die nicht realisierbare Darstellung aller Probleme, die in verschiedenen Bereichen auftretet können.

* Ablösegedanken

Die Informationen über die Ablösegendaken geben Auskunft über die Existenz von Plänen bezüglich der Ablösung der eigenentwickelten Software durch eine Andere. Dies beeinflusst die Sicht auf die Upgrades, Probleme usw.

Diese Informationen werden durch die Fragen in den folgenden Abbildungen gesammelt:

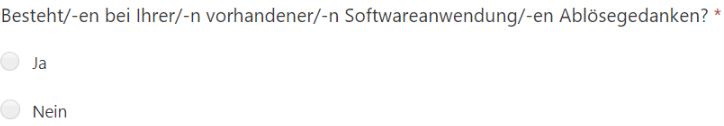


Abbildung 19: Ablösegedanken

* + - 1. Abschlussteil A

Die Fragen im Abschlussteil A dienen dazu Informationen über die Impressionen des Befragten bezüglich des Fragebogens zu sammeln. Dieser Teil steht nur den Befragten zur Verfügung, die positiv auf die Frage beantwortet haben, ob sie im Besitz einer eigenentwickelten Software sind.

Diese Informationen werden durch die Frage in der folgenden Abbildung gesammelt:

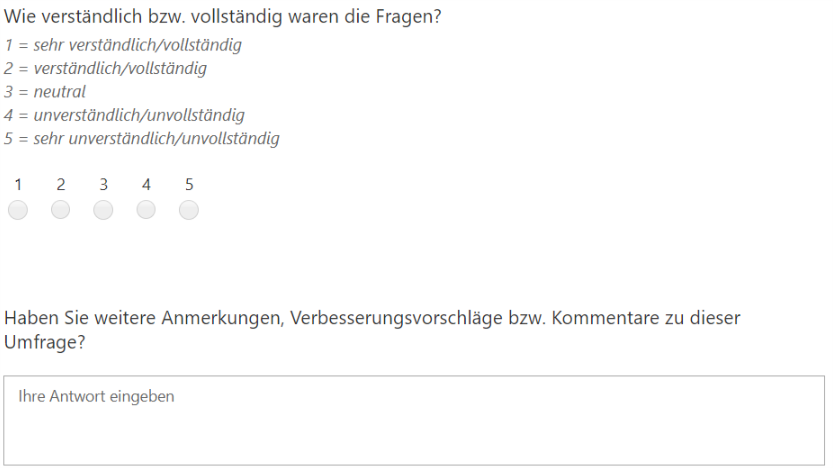


Abbildung 20: Abschlussteil A

* + - 1. Abschlussteil B

Die Fragen im Abschlussteil B dienen dazu erstens Informationen über den Grund für das nicht besitzen einer eigenentwickelten Software zu sammeln. Zweitens dienen sie dazu Informationen über die Standardsoftwareanwendungen im Besitz des Unternehmens zu sammeln. Dieser Teil steht nur den Befragten zur Verfügung, die negativ auf die Frage beantwortet haben, ob sie im Besitz einer eigenentwickelten Software sind.

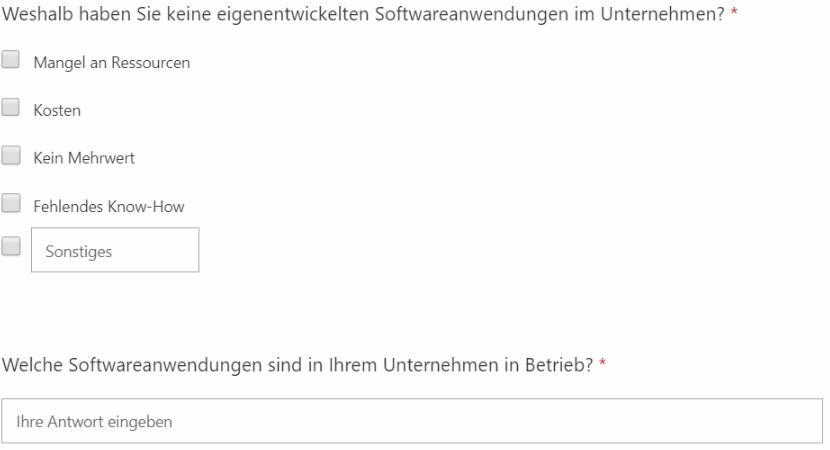
Diese Informationen werden durch die Frage in der folgenden Abbildung gesammelt:

Abbildung 21: Abschlussteil B

## Durchführung der Umfrage bei 5-10 Unternehmen

## Analyse der Antworten & des Erhebungsinstruments

## Erstellen einer allgemeingültigen Analyse- und Auswertungsstrategie

## Phase N

# Ergebnisse

# Diskussion

# Tabellenverzeichnis (Notwendig, falls vorhanden)

# Abbildungsverzeichnis

[Abbildung 1: Struktur des Fragebogens 8](file:///C:\Users\tomic\Documents\GitHub\PS_InfoEng\Arbeitsdokumente\Seminararbeit_Zwischenstand_10.12.docx#_Toc448567)

[Abbildung 2: Implementierungszeitraum-Frage 11](#_Toc448568)

[Abbildung 3: Einführungskosten-Frage 12](#_Toc448569)

[Abbildung 4: Unterstütze Kernprozess-Frage 12](#_Toc448570)

[Abbildung 5: Programmiersprache - Frage 13](#_Toc448571)

[Abbildung 6: Architektur 14](#_Toc448572)

[Abbildung 7: Architektur 14](#_Toc448573)

[Abbildung 8: Architektur 15](#_Toc448574)

[Abbildung 9: Typ 15](#_Toc448575)

[Abbildung 10: Konnektoren 16](#_Toc448576)

[Abbildung 11: Konnektoren 16](#_Toc448577)

[Abbildung 12: Dokumentation 17](#_Toc448578)

[Abbildung 13: Aufrechterhaltungskosten 17](#_Toc448579)

[Abbildung 14: Aufrechterhaltungskosten 18](#_Toc448580)

[Abbildung 15: Upgrades 18](#_Toc448581)

[Abbildung 16: Upgrades 19](#_Toc448582)

[Abbildung 17: Upgrades 19](#_Toc448583)

[Abbildung 18: Probleme 19](#_Toc448584)

[Abbildung 19: Ablösegedanken 20](#_Toc448585)

[Abbildung 20: Abschlussteil A 21](#_Toc448586)

[Abbildung 21: Abschlussteil B 21](#_Toc448587)

# Literaturverzeichnis

[1] M. Golensky and G. L. DeRuiter, “The urge to merge: A multiple-case study,” *Nonprofit Manag. Leadersh.*, vol. 13, no. 2, pp. 169–186, 2007.

[2] B. Shahzad, A. M. Abdullatif, N. Ikram, and A. Mashkoor, “Build Software or Buy: A Study on Developing Large Scale Software,” *IEEE Access*, vol. 5, pp. 24262–24274, 2017.

[3] C. Aichele and M. Schönberger, *Der professionelle Einstieg in die erfolgreiche App-Entwicklung*. 2016.

[4] P. Mertens, F. Bodendorf, W. König, M. Schumann, T. Hess, and P. Buxmann, *Grundzüge der Wirtschaftsinformatik*. 2017.

[5] ÖNACE-2008 Klassifikation

URL: https://www.gaw.institute/projekte/tiremo/sektoren.pdf (Stand: 05.02.2019)

[6] TIOBE Index for January 2019

URL: https://www.tiobe.com/tiobe-index/ (Stand: 06.02.2019)

# Anhang (Notwendig, falls vorhanden)