



**HOCHSCHULE OSNABRÜCK**  
UNIVERSITY OF APPLIED SCIENCES

**Fakultät Management, Kultur und Technik**

**Campus Lingen**

Institut für Management und Technik

Studiengang Wirtschaftsinformatik

Sommersemester 2018

LV: Softwareentwicklungsprojekt

Prof. Dr. Ralf Buschermöhle

Prof. Dr. Reinhard Rauscher

**Projektdokumentation**

**Travelling Salesman Problem**

Datum: 27.09.2018

<u>Vorname, Name</u>	<u>Matrikelnummer</u>
----------------------	-----------------------

Andrej Drobin:	654726
----------------	--------

Julian Geerdes:	609659
-----------------	--------

Deniz Kücüktaş:	653858
-----------------	--------

## Inhaltsverzeichnis

1. Einleitung .....	1
1.1 Projektanlass und Umfeld .....	1
1.2 Zielsetzung .....	1
2. Projektumfang .....	2
3. Realisierungsphase .....	3
3.1 Analysephase .....	3
3.2 Entwurfsphase .....	3
3.3 Implementierungs- und Testphase .....	3
3.4 Projektabschluss .....	4
4. Organisationsplan .....	4
4.1 Teambesetzung .....	5
4.2 Projektschnittstellen .....	5
5.0 Projektstandards .....	6
5.1 Dokumentation .....	6
5.2 Entwurf .....	6
5.3 Implementierung/Test .....	7
6.0 Hard- und Software .....	7
6.1 Hardware .....	7
6.2 Software .....	7
7. Projektplanung .....	8
7.1 Vorgehensmodell .....	8
7.2 Sollzeitplan .....	9
7.3 Soll-Ist Vergleich .....	10
8 Arbeitszeit .....	11
9 Projektfazit .....	12



## 1. Einleitung

Im Rahmen des 4. Fachsemesters des Studiengangs Wirtschaftsinformatik bekommen Studenten die Möglichkeit, angeeignetes Wissen und Fähigkeiten in Form eines Softwareentwicklungsprojektes in die Praxis umzusetzen. Unsere Aufgabe ist es dabei, das „Travelling Salesman Problem“ oder auf Deutsch, das Problem des Handelsreisenden zu lösen. Die Aufgabe der Problemstellung umfasst folgendes: Ein Handlungsreisender betrachtet eine gewisse Anzahl Städte ( $n$ -Stück), die er alle von einer Anfangsstadt so schnell wie möglich, aber nur einmal besuchen möchte und am Ende zur Ausgangsstadt zurückkehrt. Das Projekt wird in der Programmiersprache Java umgesetzt und umfasst eine Grafische Benutzeroberfläche, welche das Visualisieren ermöglicht.

### 1.1 Projektanlass und Umfeld

Da zwei Personen in der Gruppe sich schon einmal mit dieser Problematik befassen durften, wurde die Entscheidung einer Themenfindung nach Absprache innerhalb der Gruppenmitglieder erleichtert. Ein typisches Anwendungsgebiet der Informatik sind die Optimierungsprobleme, die es in der Wirtschaft und Wissenschaft zu lösen gilt. Das berühmteste Beispiel, was in Informatikschulen besprochen wird, ist das „Travelling Salesman Problem“ oder auf Deutsch, das Problem des Handlungsreisenden. Das Projekt entsteht in Kooperation mit der Hochschule Osnabrück. Die Notwendigen Informationen und Anforderungen des Projekts, werden im Lastenheft zusammengefasst.

### 1.2 Zielsetzung

Mit dem Abschluss des Softwareentwicklungsprojektes soll es dem Anwender möglich sein, mit der Software das Problem des Handlungsreisenden mithilfe von verschiedenen Algorithmen zu lösen und diese grafisch darzustellen.

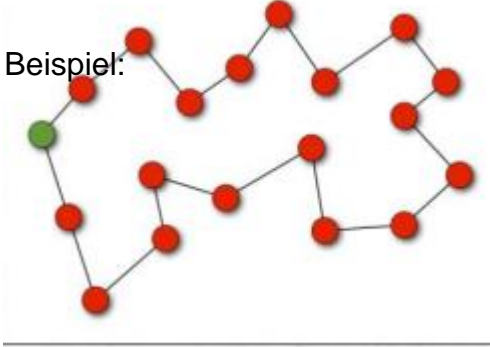
Folgendes Szenario: Ein Handlungsreisender muss auf seiner Tour eine bestimmte Zahl von Städten besuchen. Die Reihenfolge ist nicht festgelegt,



## Projektdokumentation zur Software TSP

es gilt die Bedingung, dass der Handelsreisende jede Stadt genau einmal besucht. Am Ende muss der Reisende wieder in der Ausgangsstadt ankommen. Das Programm soll die Informationen der Punkte auslesen und dann grafisch darstellen. Nachdem auslesen generiert er die kürzeste Tour zwischen allen Punkten.

Beispiel:



Hier sieht man, dass die gewählte Tour nicht rein zufällig ist. Die Tour wurde so gewählt, dass sie möglichst kurz ist. Ob es sich um die kürzeste Tour handelt, können wir durch das bloße Betrachten nicht feststellen. Somit kommen wir zu dem eigentlichen Optimierungsproblem. Das wesentliche Ziel ist es eine kurze Rundreise durch alle Knoten zu finden. Dafür dienen uns verschiedene Algorithmen - der Nearest-Neighbour-Algorithmus und der Brute-Force-Algorithmus.

## 2. Projektumfang

Mit dem Projekt „Travelling Salesman Problem“ wird ein Arbeitsaufwand von ca. 300 Stunden pro Person vorgesehen, entspricht in der Gruppe einer gesamten Stundenzahl von 900 Stunden. Das Ziel der Gruppe ist es einen für den Anwender möglichst funktionstüchtige Software herzustellen, die einfach zu bedienen und leicht zu überschauen ist. Ein wichtiger Punkt ist zudem die Zufriedenheit von allen beteiligten Personen beim Projektergebnis und der dementsprechenden Erfüllung möglichst vieler Vorstellungen. Die Vorgehensweise im Projekt wird in vier Projektphasen unterteilt, welche im weiteren Verlauf der Dokumentation noch genauer erläutert werden.



### 3. Realisierungsphase

Die Realisierungsphase dient dazu, das Softwareprojekt in einzelne Phasen einzuteilen. Diese Phasen werden durch einzelne Aufgaben definiert und finden in unterschiedlichen Zeitintervallen statt. Unser Projekt teilt sich in folgende Phasen auf: Analysephase, Entwurfsphase, Implementierungsphase, Testphase und Projektabschluss. Im Folgenden werden die weiteren Phasen erläutert.

#### 3.1 Analysephase

Die erste Phase wird als Analysephase bezeichnet, indem zunächst einmal die Projektdefinition festgelegt wird. Hierbei wird ausschließlich Abstrakt vorgegangen und der technische Inhalt wird ausgeschlossen. Dazu gehört eine klare Zieldefinition, eine ordentliche Organisation des Projektes, Beschluss verschiedener Projektstandards, die Spezifikation von Projektschnittstellen sowie die Vereinbarung von Regeln und Festlegung von Aufgaben. Durch die Aufstellung dieser Elemente entsteht eine Struktur, auf die im Laufe des Projekts immer wieder zugegriffen wird.

#### 3.2 Entwurfsphase

Hauptziel in dieser Phase ist es, ein vollständiges Lasten- und Pflichtenheft zu erarbeiten, um die Implementierungsphase zu erleichtern. In Absprache mit den Professoren werden Anforderungen und Wünsche nur umgesetzt, wenn sie auch realisierbar sind. Die Anforderungen werden nach Soll- und Kann-Kriterien definiert. Des Weiteren wird in der Entwurfsphase ein Zeitplan erstellt, um den Ablauf des Projektes darzustellen und Meilensteine beinhaltet. Um alle Anforderungen der Entwurfsphase zu veranschaulichen, wird mit einer Vielzahl von UML-Diagrammen wie zum Beispiel Zustandsdiagramm, Klassendiagramm und Use-Case Diagramm gearbeitet.

#### 3.3 Implementierungs- und Testphase

In dieser Phase wird schrittweise eine Software geformt, wobei als Grundlage die Anforderungen aus der Entwurfsphase dienen. Dauerhafte Doku-



mentation darf auch hier nicht außer Acht gelassen werden. Zur Entwicklung begleitend werden regelmäßig Tests von allen Projektmitgliedern durchgeführt, um die eigentliche Funktion des Programms zu garantieren. In der Systemdokumentation wurden alle Ergebnisse der Testphase festgehalten. Neben der Systemdokumentation muss ebenfalls eine Projektdokumentation, welche den Aufbau und Verlauf des ganzen Projekts widerspiegelt und die Benutzerdokumentation, welche alle Funktionen der Software für den Anwender darstellt, erstellt werden.

### 3.4 Projektabschluss

In der letzten Phase des Projektes werden die Dokumentationen und die Software fertig gestellt. Das Team stellt sicher, dass alle Anforderungen erfüllt worden sind und es findet ein Soll-Ist Abgleich statt, des Weiteren wird ein letztes Mal auf vollständige Funktionalität geprüft. Nach Vortrag der Abschlusspräsentation an die Professoren und Kommilitonen werden alle erstellten Dokumente den Projektbetreuern übergeben.

### 4. Organisationsplan

Eine ordentliche Organisation innerhalb der Gruppenmitglieder ist unerlässlich und sollte für eine frühzeitige Zuteilung an verschiedene Aufgaben beinhalten. Dies sorgt für Klarstellung innerhalb der Gruppe und verhindert das Unterbrechen der Projektstruktur. Eine Gruppe ist nur so stark wie sein schwächstes Glied. Um dieses Ungleichgewicht im Vorfeld schon zu verhindern, ist es notwendig, Regeln innerhalb der Gruppe aufzustellen an die sich jeder halten muss. Nach Einteilung der Aufgaben, fanden wöchentliche Meetings statt, um Ergebnisse zusammenzutragen, zu besprechen, Meilensteine abzuschließen und das weitere Vorgehen zu planen. Hatte man Schwierigkeiten die besprochenen Aufgaben umzusetzen, gab es genug Kommunikationsmittel um Fragen und Anregungen jederzeit per z.B. „WhatsApp“ zu diskutieren. Die Cloud-Website „Dropbox“ ermöglichte uns unter anderem Datenaustausch für notwendige Dokumente. Bei der Implementierung brachte uns „GitHub“ einen enormen Vorteil, da erstellte Quelltexte von Projektmitgliedern bereitgestellt und für Weiterbearbeitung zur Verfügung gestellt wurden. Weiterhin zu erwähnen ist, dass



Online-Meetings mit „Skype“ abgehalten wurden. Dieser Dienst ist allen Mitgliedern bekannt und hat sich auch in der Vergangenheit als sehr nützlich erwiesen, wenn es darum geht virtuell miteinander verbunden zu sein.

### 4.1 Teambesetzung

Das Softwareentwicklungsprojektteam „TSP“ besteht aus drei Studierenden: Andrej Drobin, Julian Geerdes und Deniz Kücüktaş. Jedes Teammitglied hat eine bestimmte Rolle, die mit Verantwortung zu tragen ist. Folgende Tabelle veranschaulicht die Zuteilung dieser Rollen.

Name	Verantwortung	Erklärung
Alle Teammitglieder	Projektleiter	Aufgabenerteilung, Kontrolle, Organisation
Andrej Drobin	Spezialist/ Kreativer Kopf	Fokus auf Programmierung
Julian Geerdes	Sprecher/Designer	Sorgt für Diskussion/ Zuständig für GUI
Deniz Kücüktaş	Zuhörer/Dokumentierer	Sorgt für Kommunikation/ Team-Chemie

Die Verantwortung des Projektleiters wird von allen Mitgliedern ausgeführt, da wir uns bei einer vergleichsweise niedrigen Teammitgliederzahl von drei Leuten, gegenseitig am besten kontrollieren, gemeinsam Aufgaben verteilen und für eine vernünftige Organisation sorgen können. Je nach entsprechenden Stärken und Schwächen wurden die anderen Verantwortungen auf die Teammitglieder verteilt. Um die Programmierung wird sich hauptsächlich Andrej Drobin kümmern, er weist eine Menge Erfahrung mit Java und dem Projektthema auf und ist mit dieser Rolle am wertvollsten für die Gruppe. Für große Teile der bei der Gestaltung und Einrichtung der GUI hat sich Julian Geerdes bereit erklärt. Der Hauptverantwortliche für die Dokumentation sowie das Pflichten- und Lastenheft ist Deniz Kücüktaş. Alle drei Teammitglieder haben zudem die Aufgabe, sich gegenseitig auf den neusten Stand der Dinge zu bringen und ihr Umgesetztes im Laufe des Projektes weiterzugeben, sodass alle auf dem gleichen Wissensstand bleiben. Dies hat das Ziel, dass bei der Abschlusspräsentation jeder in der Lage ist, auf Fragen von Professoren und Kommilitonen eingehen zu können.

### 4.2 Projektschnittstellen

Als Projektschnittstellen werden Personen bezeichnet, die von außen mit dem Projektteam zusammenarbeiten. Da sie maßgeblich an dem Erfolg



des Projektes beteiligt sind und bei Problemen oder Fragen immer zu Rat standen, werden sie in der folgenden Tabelle erwähnt.

Name	Funktion
Prof. Dr. habil. Reinhard Rauscher	Projektbetreuer
Prof. Dr. Ralf Buschermöhle	Projektbetreuer

### 5.0 Projektstandards

Zu jedem Projekt gehören sogenannte Standards, auf die man sich vor der Realisierung des Projektes einigt, um eine Gleichförmigkeit bei der Entwicklung zu schaffen. Probleme, die mehrfach auftauchen, werden bei vorher definierten Standards auf die selber Art und Weise gelöst. Dies führt zu einer leichteren Verständlichkeit und besserer Wartbarkeit. Außerdem erlaubt es einen flexibleren Einsatz von Projektmitarbeitern, da die Gleichförmigkeit zu einer schnellen Einarbeitung in neue Bereiche des Systems führt. Dadurch profitieren letztendlich auch die Anwender, da eine bessere Wartbarkeit und schnellere Reaktionszeiten für Zufriedenheit sorgen. Zu den Organisatorischen Standards zählen die wie bereits vorher erwähnten Dienste wie „WhatsApp“ für Kommunikation, „Dropbox“ für den Datenaustausch, „Github“ um Quelltext weiterzugeben und Nachzubearbeiten und „Skype“, um auch virtuelle Meetings abhalten zu können.

### 5.1 Dokumentation

Die Erstellung sämtlicher Dokumente (Lasten- und Pflichtenheft, Projektdokumentation, Benutzerdokumentation, Systemdokumentation) wurden ausschließlich mit „Microsoft Word“ erarbeitet. Jedem ist die Bedienung dieses Programms bewusst was für Flexibilität sorgt. Aus dem gleichen Grund wurde für das Erstellen von Tabellen und Protokollen „Microsoft Excel“ benutzt.

### 5.2 Entwurf

Einen Enormen Vorteil bot uns die Hochschullizenz für die Software „Visual Paradigm“, wodurch die gesamte Erstellung der Entwurfsphase und den damit zusammenhängenden Diagrammen erheblich erleichtert wurde. Die Anwendung ermöglicht und schnell und einfach eine komplexe Modellierung übersichtlich darzustellen.





### 5.3 Implementierung/Test

Für die Implementierung wurde in der Entwicklungsebene „IntelliJ“ programmiert. Während des Studienverlaufs haben wir diese Software bereits kennengelernt und uns damit angefreundet. In Folge dessen fiel uns auch die Entscheidung bei der Programmiersprache „Java“ leicht. Um Tests durchzuführen, wurde das Softwareinterne Test-Tool „JUnit“ benutzt, welches auf Java zugeschnitten wurde.

### 6.0 Hard- und Software

Um das Projekt möglichst Effektiv und Effizient zu bearbeiten, wird das Projekt mithilfe verschiedener Software optimiert. Vorher definierte Software wurde zu einer Liste zusammengestellt um ein Softwarewechsel im Laufe des Projektes auszuschließen.

#### 6.1 Hardware

- Private Laptops
- Private Tablets
- Öffentlich bereitgestellte Rechner der Hochschule Osnabrück, Standort Lingen
- Handy
- Beamer

#### 6.2 Software

- Microsoft Office
- Betriebssysteme: Windows 10
- Adobe Acrobat Reader
- WhatsApp
- Skype
- Dropbox
- Git
- Visual Paradigm
- IntelliJ IDEA
- Java Development Kit(JDK)

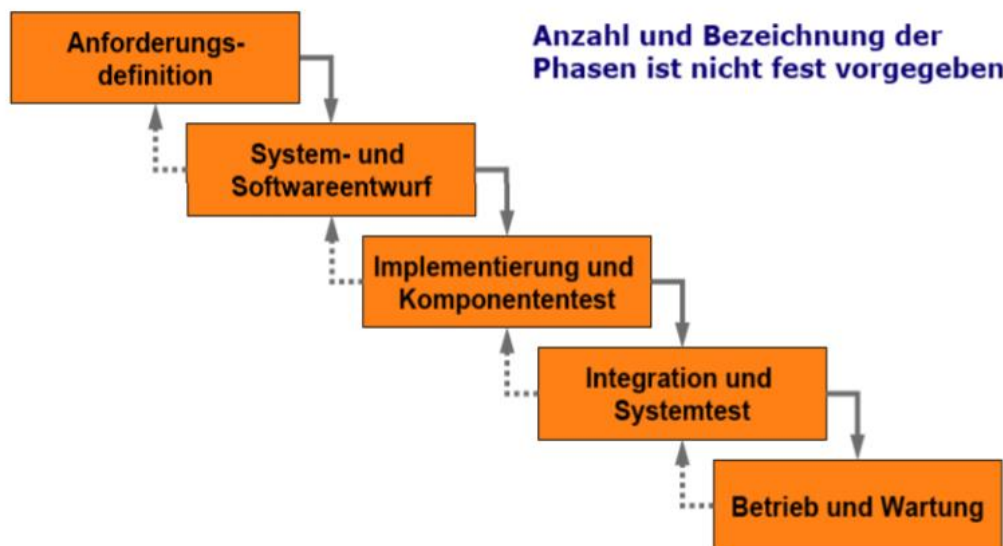


## 7. Projektplanung

Grundlage für die Projektplanung ist eine Festlegung des Vorgehensmodells und den damit folgenden Zeitplan sowie den anschließenden Netzplan. Ein weiterer wichtiger Bestandteil ist der Soll-Ist-Vergleich. Dadurch wird aufgezeigt in welchen Phasen der Soll-Zustand nicht eingehalten werden konnte.

### 7.1 Vorgehensmodell

Unser Vorgehensmodell ist das Wasserfallmodell, da unser Team bereits Erfahrung hierüber gesammelt hat und durch seine geordnete Struktur eine hohe Planungssicherheit gewährleistet. Das Projekt wird in mehrere Phasen unterteilt und baut bei einer vorher festgelegten Reihenfolge aufeinander auf, sodass nach Abschluss einer Phase die Entscheidungen nichtmehr rückgängig gemacht werden können. Zentraler Risikofaktor ist die fehlende Flexibilität am Ende des Projektes, da der Ablauf der Phasen strikt eingehalten wird und Fehler, die am Ende des Projektes auftauchen schwerer zu korrigieren sind.





## 7.2 Sollzeitplan

Sollzeitplan				
Nr	Deadline	Vorgang	Dauer	Vorgänger
1	06.03.2018	Themenauswahl	1	
2	13.03.2018	Anforderungsanalyse	7	1
3	18.03.2018	Erstellung des Lastenhefts	6	1
4	23.03.2018	Erstellung des Pflichtenhefts	5	2
5	03.04.2018	Erstellung einer Präsentation und Vorstellung des Lastenheftes	3	3,4
6	20.04.2018	Entwurfsphase	2	3,4,5
7	24.04.2018	Erstellung einer Präsentation und Vorstellung des Entwurfs	4	6
8	06.06.2018	Implementierung I, Verbesserung Entwurf	43	7
9	07.07.2018	Vorbereitung Prüfungen und Prüfungsphase	32	8
10	31.08.2018	Implementierung II	54	8,9
11	21.09.2018	Testphase	8	10
12	22.09.2018	Erstellen Systemdokumentation	15	11
13	22.09.2018	Erstellen Benutzerdokumentation	15	11
14	24.09.2018	Erstellen Projektdokumentation	175	5
15	25.09.2018	Abschlusspräsentation erstellen	4	14



### 7.3 Soll-Ist Vergleich

Mit dem Sollzeitplan wurden Ziele in einzelne Etappen definiert und zeitlich festgehalten. Nach Abschluss der Etappen wurde geprüft, ob diese rechtzeitig fertiggestellt wurden oder eine Verzögerung besteht.

Soll - Ist - Vergleich					
Nr	Vorgang	Soll	Dauer	Soll Dauer	Ist Dauer
1	Themenauswahl	06.03.2018	06.03.2018	1	1
2	Anforderungsanalyse	13.03.2018	13.03.2018	7	7
3	Erstellung des Lastenhefts	18.03.2018	18.03.2018	6	6
4	Erstellung des Pflichtenhefts	23.03.2018	23.03.2018	5	5
5	Erstellung einer Präsentation und Vorstellung des Lastenhefts	03.04.2018	03.04.2018	3	3
6	Entwurfsphase	20.04.2018	20.04.2018	22	22
7	Erstellung einer Präsentation und Vorstellung des Entwurfs	24.04.2018	24.04.2018	4	4
8	Implementierung I und Test, GUI und Verbesserung Entwurf	06.06.2018	06.06.2018	43	43
9	Vorbereitung Prüfungen und Prüfungsphase	07.07.2018	07.07.2018	32	32
10	Implementierung II und Tests	31.08.2018	06.10.2018	54	89
11	Testphase	21.09.2018	21.09.2018	8	8
12	Erstellen Systemdokumentation	22.09.2018	28.09.2018	15	21
13	Erstellen Benutzerdokumentation	22.09.2018	28.09.2018	15	21
14	Erstellen Projektdokumentation	24.09.2018	28.09.2018	178	182
15	Abschlusspräsentation erstellen	25.09.2018	25.09.2018	4	4
16	Nachbearbeitung	07.10.2018	07.10.2018	7	7

Bei unserem Soll-Ist Vergleich des Softwareentwicklungsprojektes wurden die Soll-Zeiten größtenteils eingehalten. In der zweiten Implementierungsphase benötigten wir 35 Tage mehr, da wir noch einige Überarbeitungen vornehmen mussten. Infolgedessen mussten bei der Dokumentation ebenfalls Nachbearbeitungen vorgenommen werden. Nach der Abschlusspräsentation wurde eine weitere Woche Nachbearbeitungszeit genehmigt,

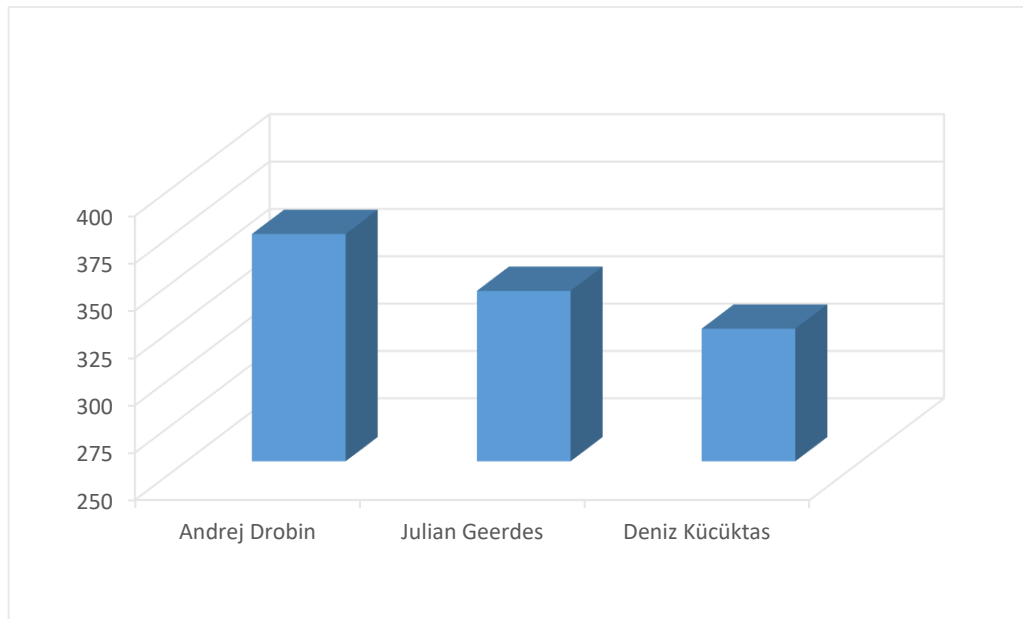


## Projektdokumentation zur Software TSP

um das Projekt den letzten Anforderungen der Projektbetreuer gerecht zu werden.

### 8 Arbeitszeit

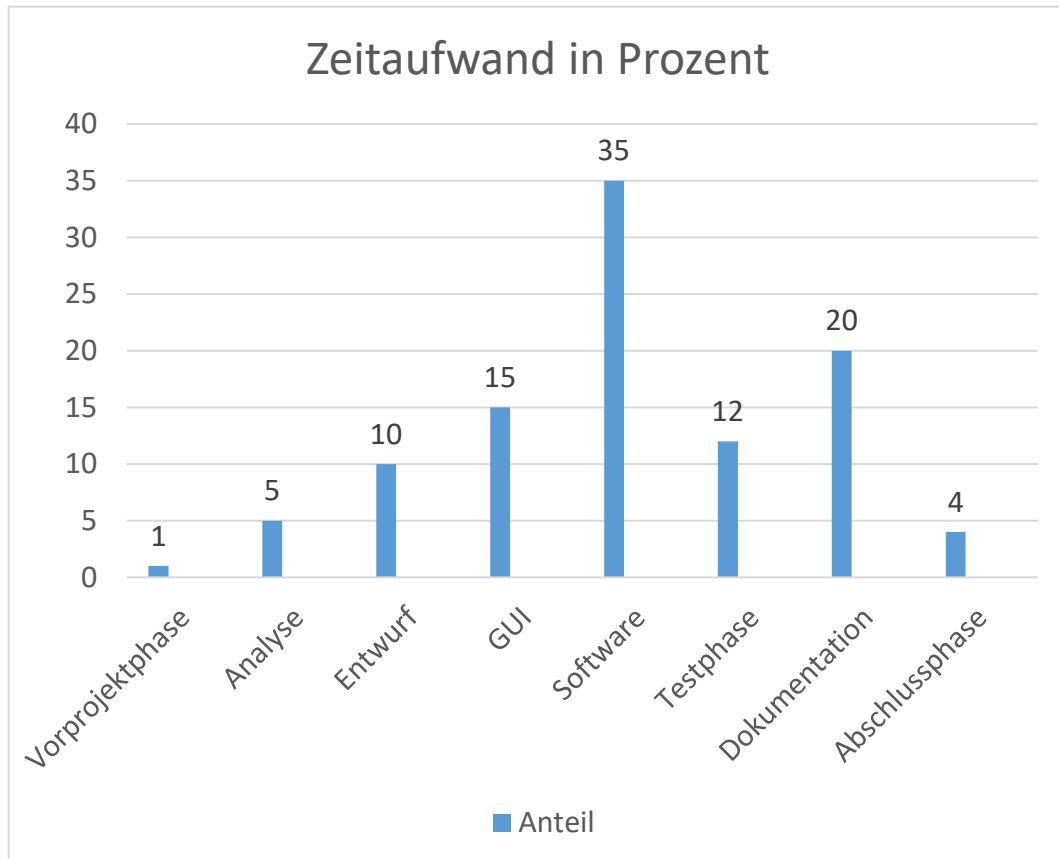
Mit der Arbeitszeit wird der Aufwand dokumentiert, welcher während des Projektes entstanden ist. Darunter wird der Aufwand pro Person und allgemein unterschieden. Beide Fälle werden anhand einer Grafik dargestellt.





## Projektdokumentation zur Software TSP

Der Arbeitsaufwand pro Person wurde zu Anfang des Projektes auf ca. 300 Std festgelegt. Da dies nur eine grobe Einschätzung war, kam es zu Abweichungen und der Aufwand war höher als die festgelegte Stundenzahl. Des Weiteren haben wir Differenzen zwischen den einzelnen Projektteilnehmern feststellen müssen. Jedoch war keiner der Teilnehmer unter der vorher festgelegten Stundenzahl, sodass jeder im Großen und Ganzen seinen Teil zum Projekt beigetragen hat.



Im oben dargestellten Balkendiagramm wird der Arbeitsaufwand in Prozent definiert und in alle Arbeitsbereiche aufgeteilt. So sieht man, dass gerade bei der Programmierung (Software) und GUI der Großteil an Stunden investiert wurde. Da die Testphase und die Dokumentation ein ständiger Begleiter in unserer Software war, wurde in diesen Bereichen ebenfalls viele Stunden investiert.

### 9 Projektfazit

Das Modul Softwareentwicklungsprojekt gab uns die Möglichkeit, die im vorherigen Semester gelernten Kenntnisse in die Praxis umzusetzen.



## Projektdokumentation zur Software TSP

Dazu gehörte ein halbes Jahr lang nicht nur die Arbeit vor dem Bildschirm, sondern auch jede Menge Teamarbeit mit Höhen und Tiefen. Die wöchentlichen Meetings und der damit verbundene Aufbau gegenseitiger Motivation sorgte nicht nur für regelmäßige Erfolge, sondern auch für Lust auf mehr. Das Ziel eine Software aus eigenen Händen zu kreieren und in der realen Welt einsetzen zu können sorgte für jede Menge Spaß und Erfahrung. Erfahrungen die in Zukunft benutzt und sicherlich weiterentwickelt werden können. Uns erstaunte wie viel so zu einem wirklichen Projekt dazugehört und lernten Praxisbeispiele für die vorherigen Theoretischen Module kennen. Bevor es wirklich zum eigentlichen Programmieren kommt, mussten wir uns aneignen, was es heißt, z.B. eine Anforderungsanalyse im Sinne von Pflichten- und Lastenheft zu schreiben oder die Entwurfsanalyse mit diversen Diagrammen für das eigene Projekt zu erstellen. Was wir hätten besser machen können sind zum einen eine bessere Einteilung des Arbeitsaufwandes. Fehler die zum Schluss erst auftraten sorgten für Probleme in vorherigen Arbeitsphasen und mussten erneut überarbeitet werden. So kann man sagen, dass eine gewisse Pufferzeit zwischen den einzelnen Phasen für einen optimaleren Arbeitsverlauf gesorgt hätte. Ein großer Pluspunkt zu Anfang, war die gute Teamchemie zwischen den Teammitgliedern. Es wurde stets für regelmäßige Kommunikation gesorgt und Differenzen untereinander schnell beseitigt.



## Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig und nur unter Zuhilfenahme der ausgewiesenen Hilfsmittel angefertigt habe. Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach anderen gedruckten oder im Internet verfügbaren Werken entnommen sind, habe ich durch genaue Quellenangaben kenntlich gemacht.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht physisch oder elektronisch veröffentlicht.

Lingen (Ems), den 27.09.2018

---

Andrej Drobin

Lingen (Ems), den 27.09.2018

---

Julian Geerdes

Lingen (Ems), den 27.09.2018

---

Deniz Kücüktaş



