Praxissemesterbericht

im Studiengang  
Computervisualistik und Design

vorgelegt von

**Marc Kammermann**Matr.-Nr.: 2140119

am 28. März 2017   
an der Hochschule Hamm-Lippstadt

Betreuer: Prof. Dr. Simon Nestler

# Zusammenfassung

Diese Dokumentvorlage ist eine Hilfestellung für die Erstellung von studentischen Arbeiten im Studiengang Computervisualistik und Design (CVD) an der Hochschule Hamm-Lippstadt (HSHL). Die Verwendung dieses Templates ist nicht verpflichtend, die existierenden Regelungen für die Erstellung von Abschlussarbeiten bleiben davon unberührt.

Allgemein gilt: Der Praxisbericht ist eine WISSENSCHAFTLICHE ARBEIT! *(Verweis auf das Buch, dass zu Studienanfang verteilt worden ist)* Deshalb sollten Sie so früh wie möglich damit beginnen. Nehmen Sie sich als Richtwert die chronologische Aufteilung in der beigefügten Roadmap!

**Schlagwörter**: Dokumentvorlage, wissenschaftliche Arbeit, Praxissemester, CVD, HSHL

*Die Arbeit beginnt mit dieser Zusammenfassung. In der Zusammenfassung geben Sie einen Gesamtüberblick über die Inhalte und Ziele Ihrer Arbeit. Zusätzlich sollten Sie noch einige Schlagwörter angeben, um die Inhalte Ihrer Arbeit zu beschreiben.*

**Umfang:** 1 Seite

***Die Angaben zum Umfang sind nur grobe Richtwerte, die je nach Bedarf in Absprache Ihrer Betreuerin bzw. Ihrem Betreuer anzupassen sind. Der Gesamtumfang Ihres Praxissemesterberichts sollte auf Grundlage dieser Formatierung ca. 20 – 25 Textseiten (ohne Titelblatt, Zusammenfassung, Inhaltsverzeichnis, Anhänge und Literaturverzeichnis) umfassen.***

# Abkürzungsverzeichnis

FhG = Fraunhofer Gesellschaft

IPA = Institut für Produktionstechnik und Automatisierung

D&B = Drag & Bot

SUS = System Usability Scale

FuB = Function Block

# Inhaltsverzeichnis

Zusammenfassung 2

Abkürzungsverzeichnis 3

Inhaltsverzeichnis 4

1 Vorstellung des Unternehmens 5

2 Projekte 7

3 Usability Studies zur Software "Drag & Bot" 10

3.1 Problemstellung 10

3.2 Stand der Technik 10

3.3 Analyse und Lösungsansatz 10

3.4 Abnahmekriterien 10

3.5 Durchgeführte Arbeiten 11

3.6 Ergebnis 11

3.7 Gestaltungsqualität 11

3.8 Zusammenfassung 11

4 Fazit 12

Anhang: Materialien 13

Literaturverzeichnis 14

Eidesstattliche Versicherung 15

# Vorstellung des Unternehmens

*Stellen Sie das Unternehmen vor, in welchem Sie Ihr Praxissemester verbracht haben. Falls Sie ein Auslandssemester gemacht haben, so stellen Sie hier die von Ihnen besuchte Partneruniversität vor.*

*Illustrieren Sie das Unternehmen bzw. die Universität mit mindestens einem Foto, um der Leserin bzw. dem Leser Ihrer Arbeit eine möglichst anschauliche Vorstellung über die konkrete Situation vor Ort zu geben.*

**Umfang:** ca. 2 - 3 Seiten

Der Autor leistete sein Praxissemester beim Fraunhofer Institut für Produktionstechnik und Automatisierung (IPA) am Standort Stuttgart Vaihingen.

Die Fraunhofer Gesellschaft (FhG) ist die größte Organisation für anwendungsorientierte Forschung in Europa mit 67 Instituten und Forschungseinrichtungen in Deutschland. Insgesamt sind in diesen Einrichtungen circa 24.000 Mitarbeiterinnen und Mitarbeiter beschäftigt, welche überwiegend eine natur- oder ingenieurwissenschaftliche Ausbildung aufweisen. Finanziert wird die Organisation zu über 70 Prozent durch industrielle Aufträge sowie öffentlich finanzierte Forschungsprojekte, circa 30 Prozent steuern Bund und Länder bei.1 Auftraggeber beziehungsweise Kooperationsunternehmen der FhG sind beispielsweise die Unternehmen „Festo“ und „Volkswagen“ sowie die Bundesdruckerei. 2

Die Fraunhofer Gesellschaft sieht ihre Mission vor allem in der innovativen Zukunftsgestaltung, wodurch sowohl das Gemeinschaftswohl als auch die Wirtschaft in Deutschland und Europa gefördert werden sollen. Die FhG definiert sich selbst als Innovationstreiber und entwickelt technologische Durchbrüche zur „Lösung künftiger Herausforderungen“ 3. Unter den Leitsätzen des Unternehmens finden sich unter anderem das Zusammenspiel aus Forschung und anwendungsorientierter Entwicklung sowie die „nachhaltige Entwicklung im Sinne einer ökologisch intakten, ökonomisch erfolgreichen und sozial ausgewogenen Welt“ 4. Auf dieser Philosophie aufbauend decken die unterschiedlichen Forschungsfelder der FhG fast jeden Bereich der menschlichen Bedürfnisse wie Gesundheit, Kommunikation oder Energie ab. Daraus lässt sich ableiten, dass die Forschungsergebnisse der unterschiedlichen Institute das Leben der Menschen sowie deren Gewohnheiten nachhaltig beeinflussen. Aufgrund der häufig wechselnden Bedürfnisse der Menschen werden immer wieder innovative Konzepte und Ideen gefordert. Fraunhofer entwickelt solche Konzepte sowie darauf aufbauende Technologien und Lösungen.

Das Fraunhofer IPA entwickelt innovative Lösungen vorrangig im Bereich der Robotik. Hierbei lassen sich im Allgemeinen zwei unterschiedliche Typen von Robotern unterscheiden, welche am IPA (weiter-)entwickelt werden: Einerseits Industrieroboter, welche entwickelt werden, um Prozesse in der Produktion zu erleichtern oder zu beschleunigen. Sie werden in der Regel stationär eingesetzt und sollen mittels einer grafischen Benutzerschnittstelle von Mitarbeitern der Einsatzunternehmen Anweisungen erhalten. Andererseits werden sogenannte Service-Roboter entwickelt, welche in Zukunft unterschiedlichste Dienstleistungen für den Menschen übernehmen sollen. Sie sind mobil und können für gewöhnlich mittels bildverarbeitungstechnischen Informationen autonom ihre Umgebung erkunden, um sich darin zu bewegen. Die vom Fraunhofer IPA erstellten Entwicklungsmodelle Care-O-bot sollen den Menschen im Haushalt (beispielsweise beim Kochen) unterstützen sowie Aufgaben im öffentlichen Bereich, wie Unterstützung von Patienten in medizinischen Einrichtungen oder Essenslieferung in Restaurants, übernehmen.

**(TODO: Foto)**

# Projekte

*Geben Sie einen Überblick über alle Projekte, in denen Sie im Rahmen Ihres Praxissemesters involviert waren. Beschreiben Sie für jedes der Projekte die Ziele, den Gesamtumfang (in Mannstunden) und ihren Eigenteil an dem Projekt (in Mannstunden).*

*Falls Sie ein Auslandssemester verbracht haben, geben Sie an dieser Stelle einen Überblick über die von Ihnen besuchten Lehrveranstaltungen.*

**Umfang:** ca. 4 – 5 Seiten

Der Autor leistet sein Praktikum am Fraunhofer IPA für das Projekt Drag & Bot (D&B).

Bei Drag & Bot handelt es sich um eine Software zur einfachen Programmierung von Robotern in der Produktionsindustrie. Momentan müssen Industrieroboter von Spezialisten individuell je nach Robotermodell und geplanter Funktion neu- oder umprogrammiert werden, sodass man bei einer sich ändernden Produktion stets einen Informatiker (o.Ä.) zu Rate ziehen muss. Dies ist jedoch für kleine und mittelständische Unternehmen ein großes Hindernis, da entsprechende ökonomische und personelle Ressourcen nicht zur Verfügung stehen. Dies birgt für die Unternehmen die Gefahr, den Anschluss an den dynamischen Markt zu verlieren. D&B soll sich in die Arbeitsabläufe oben genannter Unternehmen integrieren und die Roboterprogrammierung durch ein intuitives und einfach zu bedienendes grafisches Interface auch von Laien vollziehen lassen. Die Software ist als Single Page Web Application konzipiert und ist für Computer und Tablets optimiert. Momentan werden die Browser Mozilla Firefox und Google Chrome unterstützt. Die Produktion der Software begann im Sommer 2014 auf der Basis einiger Grundfunktionalitäten, welche zwischen 2011 und 2012 entwickelt wurden.

Die Software ist in vier architektonische Bereiche aufgeteilt. Zum einen das mittels AngularJS sowie den gängigen Web-Programmiersprachen HTML und CSS realisierte Frontend, welches mit dem auf Java basierenden Backend kommuniziert. Dieses wiederum kümmert sich um die Verwaltung der Datenbank sowie die Weiterleitung von Befehlen an den sogenannten Executor, welcher mittels dem eigens für Robotersteuerung entwickelten Frameworks ROS („Robot Operating System“) den Roboter anspricht.

**(TODO: Validieren)**

Innerhalb dieses Projekts übernimmt der Autor mehrere Aufgaben, welche im Folgenden kurz vorgestellt werden. Die Erstellung von Arbeitsabläufen für Roboter wird in D&B durch den „Builder“ realisiert, welcher für die Erstellung sogenannter Programme verwendet wird. Bei einem Programm handelt es sich um eine Aneinanderreihung von logisch aufeinander folgenden Elementen, die später vom Roboter ausgeführt werden sollen. Dabei werden vier wesentliche Kategorien von Elementen mit verschiedenen Funktionen verwendet, welche fortan als „Function Blocks“ (FuB) bezeichnet werden. Die erste Kategorie stellen die „Basic Functions“ dar, welche grundlegende Funktionen wie Bewegung, Inputs o.ä. steuern. Außerdem werden Bedingungen, wie beispielsweise Schleifen, bedingte Verzweigungen oder parallel ablaufende Anweisungen, verwendet. Eine weitere Gruppierung bilden die toolspezifischen Anweisungen wie beispielsweise die Öffnung/Schließung eines Greifers. Als vierte Kategorie sind die Subprogramme zu nennen, das heißt die Nutzung bereits vorhandener Programme als Baustein für komplexere Aufgaben. Diese werden per Drag & Drop in ein dafür vorgesehenes Fenster („Dropzone“) geschoben, in dem der Ablauf der Applikation dargestellt wird. **(TODO: Anhang Builder)**

Einige der Elemente benötigen weitere Informationen wie beispielsweise die Angabe von Koordinaten für eine Bewegungsanweisung. Die Parametrisierung dieser FuBs findet über Wizards statt. Diese sind eigenständige Programmteile, welche sowohl bei der Programmerstellung als auch zur Laufzeit Inputs und/oder Outputs erzeugen und bearbeiten können. Auf Dauer soll eine Schnittstelle implementiert werden, mit welcher beliebig viele weitere Wizards in die Software integriert werden können. Dies ermöglicht eine beschleunigte Bedienung sowie die Abdeckung von Sonderfällen. Ein Beispiel hierfür ist ein Wizard, welcher die Kamera am Roboterkopf nutzt, um per Bilderkennung beispielsweise Schraublöcher zu identifizieren und die daraus gewonnenen Daten als Parameter in einen FuB zu übernehmen.

Eines der vom Autor hauptverantwortlich implementierten Programmteile ist das „Robot Control Panel“, welches den Standard-Wizard für die Software darstellt. Der Wizard ermöglicht die manuelle Steuerung des Roboters über das Web-Interface, sodass zur Parametrisierung eines FuBs lediglich die gewünschte Position mittels des Control Panels angefahren werden muss, um sie als Werte zu speichern. Der Wizard unterstützt die Steuerung des Roboters einerseits im Weltkoordinatensystem, welches sich auf die Position des Roboterkopfes im Raum bezieht. Andererseits wird die Steuerung im Jointkoordinatensystem ermöglicht, welches jedes Glied des Roboterarms („Joints“) einzeln betrachtet bzw. anspricht. Außerdem ist eine 3D-Robotersimulation in das Panel integriert, welche den aktuellen Zustand des Roboters in Echtzeit darstellt.

**(TODO: Anhang ctrlPanel)**

Der Verfasser entwickelt weiterhin in Zusammenarbeit mit drei anderen Praktikanten ein Konzept zur Umstrukturierung der Frontend-Architektur. Dafür wird im ersten Schritt die vorhandene Architektur analysiert, wobei vor allem auf Fehler, Verbesserungspotential sowie Nonkonformität Rücksicht genommen wird. Anschließend werden drei Hauptverbesserungen herausgearbeitet und spezifiziert. Als zentrales Element dieser Umstrukturierung lässt sich die Anpassung der Web Application an das Model-View-Controller-Modell nennen. **(TODO: MVC)** In der vorhandenen Architektur gibt es mehrere Stellen, an welchen eine Kommunikation zwischen Frontend und Backend stattfindet, was aufgrund von fehlerhafter Logik beispielsweise zu doppelten Speicherungen von Programmen oder fehlenden Referenzen auf die Elemente der Datenbank führt. Um dieses Problem zu umgehen, wird ein „Core-Modul“ erarbeitet, welches als einzige Schnittstelle zwischen Darstellung und Logik fungiert. Auf dieses können andere Module der AngularJS-App zugreifen, sodass das „Core-Modul“ nunmehr die alleinige Handhabung der Daten aus der Datenbank innehält. Dadurch erhält die Software einerseits die funktionierende Logik und andererseits eine wesentlich übersichtlichere Architektur. **(TODO: Anhang Architektur Grafiken)**

Weiterhin wird die Einführung sogenannter „Helper-Klassen“ definiert, welche mehrfach verwendeten Programmcode beinhalten und auslagern. Dies sind unter anderem eine Drag & Drop Library, Wizard-Services oder darstellungsspezifische Elemente wie eine Slideshow. Als dritte Hauptänderung sind die Einführung des Component-Patterns sowie die Verwendung von fest definierten Code-Conventions zu nennen. Das Component-Pattern reguliert die Ordnerstruktur des Frontends so, dass die Ordner nach Komponenten und nicht nach Dateityp angelegt werden. **(TODO: Anhang C-Pattern googlen, ansonsten erstellen)**

Die Code-Conventions befassen sich mit der Benennung von Dateien und folgen dem Angular Styleguide von John Papa. **(TODO: Anhang John Papa, Quelle John Papa)** Die hierdurch entstehende Konsistenz hat zwar keinen Einfluss auf die Logik bzw. Funktionalität der Software, führt aber zu verbesserter Übersicht sowie zum schnellen Finden und Verstehen von Dateien und deren Abhängigkeiten.

Darüber hinaus übernimmt der Autor unterschiedlichste Aufgaben bei der Optimierung der vorhandenen Software, dazu gehören die Entwicklung und Implementierung von neuen Funktionalitäten, akut benötigte Darstellungsanpassungen des User Interfaces sowie Schnittstellenanpassungen sowie Änderungen der Datenbankstruktur im Backend.

Einen weiteren, zentralen Aufgabenbereich des Verfassers stellt die Evaluation und Verbesserung der Gebrauchstauglichkeit der vorhandenen Software dar. Hierfür übernimmt der Autor die Hauptverantwortlichkeit und befasst sich mit der Analyse, Optimierung, unter Zuhilfenahme von Konventionen und Normen, und anschließender Validierung der Änderungen. Nach Abschluss des iterativen Prozesses kümmert sich der Autor federführend, zusammen mit anderen Praktikanten, um die Integration der Ergebnisse in das interaktive System.

**(TODO: UX Vorstellen + Ergebnis/Umsetzung)**

# Usability Evaluation zur Software „Drag & Bot“

**Umfang:** ca. 13 - 15 Seiten

*Beschreiben Sie eines der in Kapitel 2 vorgestellten Projekte in der Tiefe. Erläutern Sie in dieser Einleitung kurz Ihre Vorgehensweise beim Projektmanagement und nutzen Sie für die fachliche Durchführung Ihres Projektes die nachfolgend skizzierte Untergliederung.*

*Nutzung von Scrum, Bindung an Sprintlänge, weitere Einspannung in Programmieraufgaben*

Die Organisation der Aufgaben zur Planung, Vorbereitung und Durchführung der Usability Evaluation von Drag & Bot gliedert der Autor in die vom Unternehmen genutzte Vorgehensweise via Scrum ein. Diese Technik zum Projektmanagement sieht eine Aufgabenplanung und -verteilung in (in diesem Fall) zweiwöchigen Zyklen, sogenannten „Sprints“, vor. Der Autor unterteilt den Workload in Absprache mit Vorgesetzten in mehrere kleinere Aufgaben, um den Überblick über den bereits erreichten Fortschritt zu behalten und die Langzeitmotivation sicherzustellen. Darauf aufbauend wird der Arbeitsaufwand dieser Aufgaben vom gesamten Team geschätzt, sodass eine rational zu realisierende Menge an Aufgaben im Kontext eines Sprints zugeteilt werden kann. Zur Fokussierung der Usability definiert der Autor abgrenzend zu mehreren anderen Aufgaben reine „Usability-Sprints“, während welchen er sich fast ausschließlich mit der Analyse des Ist-Zustandes, der Anpassung von Wireframes, der Definition von Use-Cases und weiteren Maßnahmen sowie anschließender Durchführung von Usability-Tests beschäftigt.

(TODO: Quelle Scrum)

## Problemstellung

*Beschreiben Sie das Problem, welches das Projekt löst. Gehen Sie dabei insbesondere auf die Projektziele, den vorgesehenen Benutzerkreis und die Bedürfnisse der Benutzer im Detail ein*.

*warum ist usability wichtig? warum sollte man usability machen? d&b macht es bisher nicht! das ist schlecht, weil…! warum sollte d&b usability machen?*

Neben der Funktionalität spielt auch die Nutzbarkeit von Software eine zentrale Rolle bei der Konzeption und Implementierung ebendieser. Die Gebrauchstauglichkeit (engl. Usability) bezeichnet das Maß an Nutzbarkeit eines Systems und beinhaltet die drei Komponenten Effektivität, Effizienz und Zufriedenheit, welche im Folgenden genauer definiert werden sollen.

Ein System gilt als effektiv, wenn mit dessen Hilfe eine Aufgabe erfüllt werden kann. Dies bezieht sich vorrangig auf das Verhältnis zwischen erwünschtem sowie tatsächlich erreichtem Ergebnis. Ein hierbei zentrales Element stellt die Genauigkeit dar, mit welcher ein Problem gelöst wird. Die möglichst aufgabengetreue Erfüllung des Auftrags muss von einem effizienten System begünstigt und gefördert werden. Effizienz beschreibt den für die Aufgabenerfüllung erforderlichen Aufwand. Eine effiziente Lösung minimiert beispielsweise den zeitlichen oder ökonomischen Aufwand, ermöglicht dem Nutzer also eine ressourcenarme Möglichkeit zum Ziel zu kommen. Die dritte Komponente Zufriedenheit beschreibt die subjektive Nutzereinschätzung bezüglich der Nutzung des Systems, beispielsweise begünstigen stress- und störungsfreies Arbeiten mit der Software die Zufriedenheit des Nutzers. Dies wird vorrangig durch Emotionen, Vorstellungen, Vorlieben und der Wahrnehmung beeinflusst. (TODO: Uelle 9241-11, lieber in Stand der Technik packen)

Im wirtschaftlichen Konkurrenzkampf benötigt ein Produkt signifikante Vorteile gegenüber ähnlichen Angeboten, um sich am Markt durchzusetzen. Eine optimierte Nutzbarkeit stellt dabei eine Möglichkeit dar, sich von der Masse abzuheben. Durch eine hohe Usability verringern sich Einarbeitungsaufwand, Zeitaufwand sowie Frust- und Stresssituationen während der Nutzung. Außerdem zeigen Studien, dass eine auf gute Usability optimierte Software die Sympathie des Nutzers der Anwendung gegenüber sowie die empfundene Seriösität des Anbieters deutlich steigert. **(TODO: Quelle Studie)** Weiterhin ist es wichtig, alle Funktionen des Programms sinnvoll in die Oberfläche zu integrieren um eine konsistente und für den Nutzer ersichtliche Programmstruktur herzustellen, in welcher er sich auf Anhieb orientieren kann. Daraus resultiert die unbedingte Notwendigkeit von Usability Studies zur Verbesserung der Gebrauchstauglichkeit von Software.

Drag & Bot wirbt als eines der drei im Fokus der Entwicklung stehenden Elemente mit der einfachen Bedienung der Anwendung. Der Aspekt Usability wird allerdings bislang bei der Entwicklung sowie der konzeptuellen Planung nicht berücksichtigt. Dadurch negiert sich aller Wahrscheinlichkeit nach die versprochene „einfache Bedienung“. Während der Einarbeitung des Autors in das Projekt fielen bereits erste Inkonsistenzen auf, welche gerade bei Beginn der Arbeit mit der Software die Orientierung beziehungsweise das Finden von Funktionen erschweren. Die Rückmeldung erster Pilotkunden zeigt außerdem, dass Abgesehen von einiger fehlerhafter oder fehlender Funktionalität vor allem Fehler und Wünsche bezüglich der Bedienoberfläche geäußert werden. D&B soll in der eingesetzten Branche auch von ungeschulten Mitarbeitern genutzt werden, welche trotzdem schnell die von ihnen benötigten Funktionen erreichen müssen. Aufgrund dessen und dem Zusammenspiel mit der Fülle an Funktionen, welche allerdings hauptsächlich von „Experten“ genutzt werden soll, ist es notwendig, eine möglichst intuitive und gut strukturierte Bedienoberfläche bereitzustellen, welche für alle Arten von potentiellen Nutzern gleichermaßen zufriedenstellend ist.

All dies impliziert, die vorhandene Version der Bedienoberfläche zu evaluieren, auf die Anforderungen der internationalen Standards zur Mensch-Computer-Interaktion zu prüfen und gegebenenfalls entsprechende Anpassungen vorzunehmen. Diese Änderungen wiederum müssen der ursprünglichen Version gegenübergestellt und getestet werden, um sie fundiert als Mehrwert bezeichnen zu können. Um dies zu erreichen führt der Autor mehrere Usability Tests durch. Die Durchführung sowie die Ergebnisse dieser Tests werden im weiteren Verlauf der Arbeit dargelegt und erläutert.

## Stand der Technik

*Beschreiben Sie den aktuellen Stand der Technik. Wie wird das Problem aktuell in dem Unternehmen gelöst? Wo sind die größten Defizite an dem gegenwärtigen Ansatz? Referenzieren Sie an dieser Stelle aktuelle Literatur zu dem Thema des Projektes.*

**WIE SETZEN ANDERE UNTERNEHMEN DAS UM?**

**ISO 9241**

**METHODEN ZUM TESTEN**

**ERSTELLUNG VON WIREFRAMES IM DETAIL**

Die internationale Norm ISO 9241 mit dem Titel „Ergonomie der Mensch-System-Interaktion“ gilt heutzutage als allgemeiner Standard im Bereich Usability, weshalb ihr Inhalt auch hier als Arbeitsgrundlage dient. Hierbei werden hauptsächlich vier Teilbereiche der Norm berücksichtigt:

* DIN EN ISO 9241-11:2016 Gebrauchstauglichkeit: Begriffe und Konzepte
* DIN EN ISO 9241-13:1998 Benutzerführung
* DIN EN ISO 9241-110:2006 Grundsätze der Dialoggestaltung
* DIN EN ISO 9241-210:2010 Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme

### Grundlagen der Dialoggestaltung

Der Teil 110 der oben genannten ISO-Norm definiert sieben Anforderungen an gute Interfaces, welche „als eine Zusammenstellung allgemeiner Ziele für die Gestaltung und Bewertung von Dialogen“ (TODO: Quelle 9241:110 4.1) dienen.

(TODO: Abbildung 9241-110)

Als erste Anforderung wird die Aufgabenangemessenheit genannt. Diese ist gegeben, wenn ein System den Nutzer bei der Erfüllung seiner Aufgabe leitet und unterstützt. Dabei ist essenziell, einerseits alle für die Aufgabenerfüllung relevanten Informationen einzublenden, andererseits auch unwichtige Informationen, welche sich oftmals kontraproduktiv auswirken, nicht anzuzeigen. Außerdem sollten möglichst intelligente Vorauswahlen getroffen werden, welche die Bedienung vereinfachen. Ein Beispiel dafür stellt das Vorschlagen der am häufigsten verwendeten Eingabe dar. Letztlich sind auch die Schnittstellen zwischen Nutzereingabe und Systemausgabe zu beachten, beispielsweise sprachbasierte Kommunikation mit dem System beim Autofahren, um dem Nutzer eine an die Situation angepasste Interaktionsmöglichkeit anzubieten.

Die Selbstbeschreibungsfähigkeit beschreibt das konsequente Wissen des Nutzers, wo er sich in einer Software zu jedem beliebigen Zeitpunkt befindet. Dies wird unter anderem durch schrittweise Führung durch bestimmte Dialoge, Anzeige von Systemzustandsänderungen und genaue Informationen über erwartete Interaktionen, beispielsweise Darstellung des gewünschten Formats einer Eingabe, erreicht.

Unter Erwartungskonformität versteht man die für den Nutzer vorhersehbare Handlungsweise eines Systems. Dabei sind Konsistenz und Erfahrung des Klienten die beiden zentralen Elemente. Software sollte einerseits nur zielgruppengerechtes Vokabular verwenden, auch unter Berücksichtigung kultureller sowie sprachlicher Disparitäten, andererseits auch in bekannten Konzepten geordnet sein. Ferner müssen für ein gutes Nutzererlebnis Rückmeldungen über vom Bediener getätigte Schritte angezeigt werden. Hierbei ist auf konstruktive sowie den Anforderungen entsprechende Dialoggestaltung zu achten.

Um den Einstieg in eine neu zu erlernende Software zu erleichtern ist auf Lernförderlichkeit zu achten. Lernförderlichkeit soll den Nutzer mittels Tutorials sowie erklärenden Anzeigen, zum Beispiel bei erstmaliger Nutzung einer Funktion, bei der Aufgabenerfüllung unterstützen. Dabei sollten Hilfestellungen sowohl für Laien als auch für Experten optimiert sein. Der Anfänger soll unterstützt werden, der Experte allerdings nicht an der Arbeit gehindert werden. Des Weiteren sollten Informationen immer auf Abruf verfügbar sein.

Steuerbarkeit ist definiert als vom Nutzer anpassbare Richtung und Geschwindigkeit eines Interfaces. Dies wird vorrangig durch das Anbieten von Entscheidungsmöglichkeiten, oft mit der häufigsten oder logisch folgenden Variante als Vorauswahl, umgesetzt. Darüber hinaus sind oft verwendete Konzepte zur Erreichung von Steuerbarkeit die Zwischenspeicherungen von Eingaben, die Archivierung veralteter Daten zur Wiederaufnahme in den Kontext sowie die Möglichkeit, ausgeführte Funktionen zu revidieren. Gerade bei großen Datenmengen ist es außerdem von Vorteil, mittels Filterung und/oder unterschiedlicher Darstellungsformen die Übersichtlichkeit der Informationen zu gewährleisten.

Ein weiteres zentrales Element bei komplexen Systemen ist die Fehlertoleranz. Vorrangig soll versucht werden, Fehler des Nutzers zu vermeiden. Bei trotzdem auftretenden Fehlern muss der Dialog einerseits den Fehler sowie dessen Behebung für den Nutzer möglichst spezifisch erläutern, andererseits auch visuelle Hilfestellung an fehlerhaften Stellen anbieten. Durch technische sowie konzeptuelle Fortschritte ist es heutzutage auch üblich, Fehler in Echtzeit zu erkennen und zum Beispiel schon vor Absendung eines Formulars kenntlich zu machen. Auch sollten interaktive Systeme in der Lage sein, Fehler selbst zu beheben, was allerdings angezeigt und gegebenenfalls vom Nutzer angepasst werden muss.

Die jüngere Entwicklung der Dialoggestaltung zeigt einen Trend zur Individualisierbarkeit von Software auf. Dies resultiert in der Anpassung des Nutzungserlebnisses an die Bedürfnisse des einzelnen Nutzers. Eine konkrete Anwendung dessen stellt die Barrierefreiheit von Software durch die Anpassung der Schriftgröße sowie Sprachausgabe von Texten dar. Das Wechseln sowie Anpassen der Erscheinungsform eines User Interfaces steht dabei neben individueller Benennung sowie Hinzufügung von Funktionen. Außerdem sind unterschiedliche Wege zur Zielerreichung und damit Anpassung an unterschiedliche Nutzerniveaus ein weiterer Aspekt von Individualisierbarkeit. Analog zur Steuerbarkeit und Fehlertoleranz soll außerdem die Möglichkeit gegeben sein, zum ursprünglichen Dialogzustand zurückzukehren.

Diese für gute Nutz-/Bedienbarkeit unabdingbaren Richtlinien stehen sich nicht zwangsläufig gleichrangig gegenüber. Jedes interaktive System sowie jedes Konzept für ein solches muss hinsichtlich unterschiedlicher Faktoren analysiert werden, sodass eine Gewichtung der einzelnen Prinzipien stattfinden kann. Beeinflussend wirken unter anderem die Ziele der Organisation, die mit dem System auszuführenden Aufgaben, die Anforderungen der Nutzer sowie der technische Hintergrund, auf welchem das System aufbaut. (TODO: ISO 9241 4.2)

Die in der Norm skizzierten Anforderungen können auf unterschiedliche Weise für die Entwicklung interaktiver System benutzt werden, zum Beispiel zur genauen Spezifikation eines Nutzungskonzeptes hinsichtlich oben genannter Einflussfaktoren. In dieser Arbeit wird eine bereits vorhandene grafische Oberfläche an die Richtlinien der Norm angepasst, dabei werden Dialogtechniken und -konzepte der ISO 9241-14 bis ISO 9241-17 benutzt. (TODO: mache ich das wirklich?)

Hauptziel durch die Einhaltung der hier erläuterten Prinzipien ist die signifikante Verbesserung der in Kapitel X.X.X (TODO: Kapitel 9241-11) genannten Grundsäulen von Gebrauchstauglichkeit. (TODO: Grafik 9241-110)

http://link.springer.com/chapter/10.1007%2F978-3-319-41956-5\_16

<http://ieeexplore.ieee.org/document/7433059/?reload=true>

<http://link.springer.com/chapter/10.1007%2F978-3-319-32467-8_92>

## Analyse und Lösungsansatz

*Analysieren Sie Ihre Problemstellung und erläutern Sie den daraus von Ihnen oder von dem Unternehmen gewählten Lösungsansatz. Gehen Sie dazu insbesondere auf das allgemeine Prinzip und die grundlegenden Werkzeuge ein, welcher zur Lösung des Problems verwendet werden. An dieser Stelle sind auch die von Ihnen verwendeten Tools und/oder Programmiersprachen zu erläutern. Verweisen Sie auch auf das resultierende Pflichtenheft*.

*User Stories, Alter Screenshot, Erklären Was Bzgl Iso Nicht Beachtet Wird, Ausblick Wie Es Zu Lösen Ist, Erklärung Wireframe Tool Usw*

Als ersten Analyseschritt definiert der Autor in mehreren Meetings in Kooperation mit Mitarbeitern und Vorgesetzten sogenannte User Stories. Dabei handelt es sich um typische Anwendungsaufgaben, die verschiedene Arten von Nutzern mit dem System zu erledigen haben. Diese User Stories werden im weiteren Verlauf genutzt, um konkrete Use-Cases zu definieren und diese als Testaufgaben bei der Usability-Evaluierung zu integrieren.

D&B richtet sich laut Zielgruppenanalyse an drei unterschiedliche Gruppen von Unternehmensmitarbeitern. Erstere besteht aus Produktionsmitarbeitern, welche nur grundlegende Funktionen der Software nutzen und diese dementsprechend schnell zu finden sowie übersichtlich angezeigt sein sollten. Außerdem benötigt diese Nutzergruppe große Bedien- und Anzeigeelemente, da sie hauptsächlich in Werkshallen arbeitet und entsprechend des Umfeldes trotzdem effizient agieren muss. *Diese Mitarbeiter müssen vor allem Roboter starten beziehungsweise stoppen können und sollen die Zustände der von ihnen verwalteten Roboter auf einen Blick sehen können. Zudem sollen sie die Möglichkeit erhalten, bei Fehlern, welche sie nicht selbst lösen können, einen Techniker zu rufen, welcher ihnen bei der Behebung der Probleme assistiert*. Als zweite Gruppe sind Techniker zu nennen, welche sich hauptsächlich um die eigentliche Programmierung und die Wartung der sich im Produktionsprozess befindlichen Roboter kümmern. *Techniker sollen neue Roboterprogramme erstellen und bestehende Programme bearbeiten sowie auf andere Produktionsteile umparametrisieren. Sie sollen Probleme und Fehler des Roboters identifizieren können um sie zu lösen.* Dementsprechend benötigt diese Zielgruppe einen umfangreichen Funktionspool, welcher trotzdem ohne großen Einarbeitungsaufwand zu bedienen sein muss. Die letzte Hauptzielgruppe wird als Experten definiert, welche Informationen über den Produktionsprozess benötigen. *Hierbei müssen sie einerseits den Fortschritt der Produktion erkennen, um neue Aufgaben und deren zeitliche Einordnung in den Prozess planen zu können, andererseits aber auch eine Übersicht über die Produktionsauslastung erhalten, womit sie die Wirtschaftlichkeit des/der Roboter(s) abschätzen können.*

Aufgrund der oben erläuterten Probleme bezüglich der Bedienbarkeit der Software bietet sich die Darstellung sowohl der aktuellen als auch einer veränderten Version mittels Wireframes an, welche anschließend mittels A/B-Test miteinander verglichen werden. Zum schnellen Erhalt erster grundlegender Informationen entscheidet sich der Autor für die Nutzung von Screen-Captures sowie für die Auswertung des von den Testteilnehmern beantworteten SUS-Fragebogens. Der Autor nutzt die System Usability Scale aufgrund ihrer hohen Vergleichbarkeit mit anderen Softwaresystemen sowie dem schnellen Erhalt von Ergebnissen, was aufgrund der zeitlich begrenzten Praktikumszeit sowie eingeschränkter Mittel zwingend notwendig ist.

Für die Erstellung von Wireframes wird das Prototyping-Tool „Balsamiq“ verwendet, welches als eines der Führenden im Bereich der Wireframe-/Mockup-Erstellung gilt. Zu den offensichtlichen Vorteilen dieser Software bezogen auf die angewandte Methodik zählen unter anderem die zeiteffiziente Erstellung von Skizzen, welche ebenso schnell bearbeitet werden können, sowie den sogenannten „Präsentationsmodus“, **(TODO: heißt der so?)** in welchem per Klick auf entsprechend programmierte Elemente der Wechsel zwischen unterschiedlichen Screens ermöglicht wird. Weiterhin wird mittels der Screen-Capture-Software „AutoScreenRecorder“ **(TODO: heißt der so?)** das Geschehen auf dem Display während der Usability Tests erfasst, wobei Mausklicks zur besseren Identifikation hervorgehoben werden. Dies ermöglicht in Zusammenspiel mit audiovisuellen Aufnahmen der Probanden die nachträgliche Analyse des Geschehenen.

## Abnahmekriterien

*Erläutern Sie Ihre Abnahmekriterien an das Projekt und wie Sie die Erreichung der Anforderungen messen und (nachhaltig) sicherstellen.*

*Sus Beschreiben, daten der tests, pilotkunden*

Bei der Analyse der durch die Studien erhobenen Daten sowie des daraus resultierenden potentiellen Erfolges der Arbeit lassen sich drei Hauptbestandteile unterscheiden.

Zuerst zu nennen sind die Ergebnisse der SUS-Tests. Die „System Usability Scale“ ist ein vom Briten John Brooke im Jahr 1986 entwickeltes Instrument zur Evaluation von Usability. Sie gilt als allgemein anerkannte und oft verwendete Methode, da sie schnelle Ergebnisse unabhängig der Plattform des Systems liefern kann. Sie stützt sich ausschließlich auf die subjektive Einschätzung von Testperson bezüglich der Gebrauchstauglichkeit einer Software. Daraus resultiert die zwingend notwendige Befragung mehrerer Nutzer, um dadurch einen Durchschnitt zu errechnen und … Zum Erhalt valider Daten wird von den Testpersonen im Anschluss an den eigentlichen Usability-Test, also der Erfüllung mehrerer Aufgaben, ein Fragebogen bezüglich ihren Erfahrungen mit dem System ausgefüllt. Dieser Fragebogen enthält 10 Items zur Nutzung des Systems, welche eine Skala mit mittlerer Kategorie von 1 („strongly disagree“) bis 5 („strongly agree) als Antwortgrundlage anbietet.

Die Ergebnisse der einzelnen beantworteten Fragebögen lassen sich zum sogenannten „SUS-Score“ addieren, welcher zwischen 0 und 100 liegt und auf dessen Grundlage eine Einordnung sowie ein Vergleich stattfinden kann. Als Ziel setzt sich der Autor im ersten Schritt die Erreichung eines SUS-Scores von über 68, was als durchschnittlicher Wert von über 200 per SUS getesteten Softwaresystemen ermittelt wurde. **(TODO: Quelle SUS Score, grafik)** Jeder Wert über 68 impliziert eine gute Usability, 100 gilt als perfekte Gebrauchstauglichkeit und Werte kleiner als 50 weisen auf große Mängel bezüglich der Nutzbarkeit hin. Trotz der Skalierung von 0 bis 100 stellt der SUS-Score keine Perzentile dar, sollte also zur besseren Einordnung in den Gesamtkontext der bisher per SUS getesteten Systeme in absolute Prozentwerte umgewandelt werden. Dieser Wert liefert eine Grundlage, anhand derer die Verbesserung vom ursprünglichen Zustand zur optimierten Version der Software festgestellt werden kann. Vorteile der System Usability Scale sind sowohl die kurze Bearbeitungszeit als auch die Abstinenz längerer Vorbereitungsprozesse. Aufgrund der allgemein gestellten vorgeschlagenen Fragen von Brook ergeben sich außerdem breit gefächerte Anwendungsmöglichkeiten der Methode. Sie kann sowohl für klassische Anwendungssoftware als auch für Webseiten oder sogar Hardware genutzt werden. Nachteilig zu sehen sind einerseits fehlende Hinweise auf konkrete Probleme des getesteten interaktiven Systems, andererseits besteht keine Möglichkeit, einzelne Tester isoliert zu betrachten. Weiterhin lassen sich zwei Systeme nicht direkt vergleichen, da die anschaulichen Vor- bzw. Nachteile der beiden individuellen Systeme nicht direkt ersichtlich sind.

Bezüglich der Auswertung des Protokolls sowie der während der Tests aufgenommenen Screen-Captures liegt das Hauptaugenmerk einerseits auf der gemessenen Zeit pro Use-Case, andererseits auf der Anzahl an Fehlern, welche bis zum erfolgreichen Abschluss der Aufgaben gemacht wurden. Diese beiden Datentypen liefern, zusätzlich zu den von einzelnen Probanden subjektiv beeinflussten Daten der SUS-Tests, objektive und messbare Werte, welche eine weitere direkte Vergleichsmöglichkeit der beiden Softwareversionen darstellen. Diese Daten zeigen im Gegensatz zu den mittels SUS-Methodik erhobenen Daten konkrete Fehlerquellen beziehungsweise Verbesserungsvorschläge auf. Die nachträgliche Erfassung und Katalogisierung aller von den Testpersonen begangenen Fehler resultieren in klar definierten Schwächen der Navigation, des Aufgabenflusses sowie der Verständlichkeit des Systems. Außerdem deckt die Zeit pro Use-Case mögliches Optimierungspotential hinsichtlich der Anordnung von Funktionen sowie eine eventuell unzulängliche Erreichbarkeit derer auf.

Als abschließendes subjektives Abnahmekriterium ist das Feedback von Pilotkunden zu nennen, welches bereits vor der Integration des Autors in die Gebrauchstauglichkeitsanalysen des Projekts eingeholt wurde. Die darin enthaltenden Kritikpunkte bezüglich Usability wurden in die konzeptionelle Umsetzung eines optimierten Nutzerinterfaces miteinbezogen. Ein Ziel dieser Nutzerstudien ist es, möglichst weniger (bzw. weniger gewichtige) negative Kritikpunkte in zukünftigen Kundenrückmeldungen zu finden.

Im Falle einer signifikanten Verbesserung der Nutzbarkeit ist eine wireframekonforme Umsetzung innerhalb der Applikation zwingend notwendig, um die ermittelten Daten auf das fertige Produkt anwenden zu können. Eine abschließende Evaluierung der umgesetzten Verbesserungen und dem daraus resultierenden von Grund auf umgestalteten User Interfaces ist wünschenswert, allerdings aus zeitlichen Gründen im Rahmen des Praktikums des Autos leider nicht möglich.

## Durchgeführte Arbeiten

**NEUER SCREENSHOT**

**ERKLÄREN WAS ICH VERBESSERT HABE**

**UMSETZUNG (PROGRAMMIERTECHNISCH) UMREIßEN**

*Gehen Sie an dieser Stelle im Detail auf alle von Ihnen im Zusammenhang mit dem beschriebenen Projekt durchgeführten Arbeiten ein. Beschreiben Sie die einzelnen Schritte, welche zur Erreichung Ihres Projektziels geführt haben.*

*Beachten Sie, dass Sie bei Softwareprojekten die Testergebnisse (ggf. Erläuterung Testszenarien) und die Usability (ggf. Erläuterung User Tests) erörtern.*

## Ergebnis

**SCREENSHOTS ALTE VERSION, NEUE VERSION**

**TABELLE ZU TESTERGEBNISSEN**

*Dokumentieren Sie das Ergebnis des Projektes. Zeigen Sie Ihre erreichten Projektergebnisse anhand von Screenshots und Fotos. Gehen Sie insbesondere darauf ein, in welchem Umfang die Ziele des Projektes (auch hinsichtlich der Abnahmekriterien) erreicht wurden.*

## Gestaltungsqualität

*Erläutern Sie die Gestaltung der gewählten Form der Arbeit (Broschüre, PDF, Homepage, interaktives Dokument). Begründen Sie, warum Sie sich für diese Art der Visualisierung entschieden haben.*

## Zusammenfassung

*Fassen Sie das Projekt zusammen, indem Sie den Nutzen des Projektes für die Endanwender erläutern. Geben Sie einen Ausblick auf mögliche zukünftige Folgeprojekte und Weiterentwicklungen.*

# Fazit

*Ziehen Sie ein persönliches Fazit zu Ihrem Praxissemester sowie über die Tätigkeit im Unternehmen als solche. Gehen Sie auf die positiven und negativen Aspekte Ihrer Tätigkeit ein. Stellen Sie außerdem dem Mehrwert heraus, der sich für Ihre weitere berufliche Zukunft aus der Durchführung des Praxissemesters ergibt. Gehen Sie insbesondere auf die Sicherung der Nachhaltigkeit ein und erläutern Sie das Lessons Learned. Reflektieren Sie insbesondere das Praxissemester hinsichtlich der bisherigen Studieninhalte!*

**Umfang:** ca. 1 – 2 Seiten

# Anhang: Materialien

*Nehmen Sie weitere Materialien in den Anhang auf, welche für das Verständnis Ihrer Arbeit essentiell sind. Hängen Sie insbesondere zentrale Passagen aus Ihrem Programmcode an, sofern diese für Ihre Arbeit von zentraler Bedeutung sind. Hängen Sie weitere Darstellungen, Illustrationen und Screenshots an, um Ihre konkrete Vorgehensweise im Detail zu erläutern.*

*Referenzieren Sie die jeweiligen Materialen bei Bedarf im Hauptteil der Arbeit.*

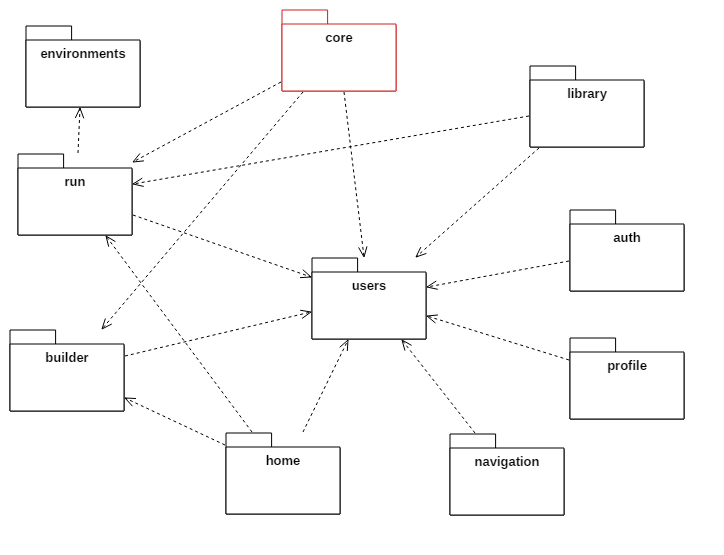


Abbildung 1: Frontendarchitektur vor Refactoring

Abbildung 2: Frontendarchitektur nach Refactoring

**Umfang:** je nach Bedarf

# Literaturverzeichnis

*Bitte geben Sie Ihre Quellen an. Formattieren Sie die Literaturangaben konsistent entsprechend eines Standards Ihrer Wahl (z.B. MLA, APA, ISO 690).*

*Zitieren Sie die Quellen im Hauptteil der Arbeit nach der amerikanischen Zitierweise.*

1 <https://www.fraunhofer.de/de/ueber-fraunhofer/profil-selbstverstaendnis/zahlen-und-fakten.html>

2 <https://www.fraunhofer.de/de/forschung/leistungsangebot/referenzbeispiele.html>

3 <https://www.fraunhofer.de/de/ueber-fraunhofer/profil-selbstverstaendnis/leitbild.html>

<https://www.dragandbot.com/#vision> 3.1

**Umfang:** je nach Bedarf

# Eidesstattliche Versicherung

**Name:**  Kammermann

**Vorname:**  Marc

**Matrikel-Nr.:**  2140119

**Studiengang:** Computervisualistik und Design

Hiermit versichere ich, Marc Kammermann, an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken ent­nommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

**Auszug aus dem Strafgesetzbuch (StGB)**

§156 StGB: Falsche Versicherung an Eides Statt

Wer von einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

Ort, Datum Unterschrift