

Konzeption und Evaluation  
von Benutzerkonditionierung in Virtual Reality

**Bachelorarbeit**im Rahmen der Bachelorprüfung  
zum Erlangen des akademischen Grades  
„Bachelor of Science“

Studiengang Angewandte Informatik

**Robert Zlomke**Matrikelnummer: 193 045

Erstprüferin: Prof. Dr. W. Issen-Schaftlich  
Zweitprüferin: Dr. Re Cherche

Heilbronn, April 2021

# Kurzfassung

Virtuelle Umgebungen zu Fuß zu erkunden ist die realistischste und natürlichste Schnittstelle. Zeitgleich ist dies auch technisch und logistisch am anspruchsvollsten.

Allgemein wurden in bisherigen Arbeiten sich auf die Fortbewegung in übereinstimmenden oder leeren Umgebungen konzentriert. Weniger wurde darauf geachtet, wie sich eine Änderung der Nichtübereinstimmung zwischen der physischen Umgebung und ihrer virtuellen Darstellung auf die Benutzer auswirkt.

**Schlagwörter**: Wissenschaftliche Arbeit, Dokumentvorlage, Hochschule

# Abstract

Exploring virtual environments on foot is the most realistic and natural interface. At the same time, this is also the most demanding technically and logistically. In general, previous work has focused on locomotion in coincident or empty environments. Less attention was paid to how changing the mismatch between the physical environment and its virtual representation would affect users.

**Keywords:** academic research and writing, style sheet, university

# Inhaltsverzeichnis

Kurzfassung 2

Abstract 2

Inhaltsverzeichnis 3

Abbildungsverzeichnis 4

Vorwort 5

1 Einleitung 6

2 Theoretische Grundlagen 7

2.1 Virtual Reality 7

2.1.1 Locomotion 7

2.2 Operante Konditionierung 7

2.3 Technologien 7

2.3.1 Unity 3D 7

2.3.2 Oculus Quest 7

2.3.3 Visual Studio 8

2.3.4 Programmiersprache C# 8

2.3.5 Git Versionskontrolle 8

3 Technische Umsetzung 10

3.1 Einbindung der Oculus Quest in Unity 3D 10

3.2 Entwicklung des Malus 10

3.3 Entwicklung der Szenarios 10

3.4 Datenerfassung 10

4 Evaluation 11

5 Fazit 12

Anhang A: Beispiele für die Gliederung von Diplomarbeiten 13

A.1 Literaturarbeiten 13

A.2 Systementwicklungen 13

Anhang B: Checkliste zur Bewertung wissenschaftlicher Arbeiten 14

Quellenverzeichnis 16

Eidesstattliche Erklärung 17

# Abbildungsverzeichnis

Abbildung 1: Fakultative und optionale Bestandteile   
einer wissenschaftlichen Arbeit 10

Abbildung 2: Beispiel Abkürzungsverzeichnis 12

# Vorwort

Dieses Dokument stellt den Versuch dar, den Studierenden im Studiengang Software Engineering das Erstellen wissenschaftlicher Arbeiten zu erleichtern.

Es ist im Sommer 2007 entstanden und basiert auf Ausarbeitungen vieler Kolleginnen und Kollegen, insbesondere Prof. Dr. Wolf-Fritz Riekert (Riekert, 2001) und Prof. Dr. Angela Koch (Koch, o.J.).

Über Kritik, Verbesserungsideen etc. freue ich mich – am besten per Mail an marsden@hs-heilbronn.de

# Einleitung

Der Überblick über Ihre wissenschaftliche Arbeit ist der Teil, den Sie[[1]](#footnote-1) als letztes schreiben. Er dient als vorausschauende Orientierung (advanced organizer) über die gesamte Arbeit, d.h. er teilt den Lesenden mit wie der Text strukturiert ist, was sie in welchem Kapitel erwartet etc. Vorher wird meist Bezug genommen auf die Ausgangslage, d.h. die Ist-Situation, auf deren Basis des sinnvoll erscheint, sich mit dem Thema zu beschäftigen. Im vorliegenden Fall könnte das folgendermaßen aussehen:

Mittlerweile liegen zahllose exzellente Leitfäden und Bücher vor, in denen Studierenden Hinweise dazu gegeben werden, wie wissenschaftliche Arbeiten zu verfassen sind. Gleichwohl gibt es auch für den Studiengang Software Engineering an der Hochschule Heilbronn den Bedarf, hierzu eine Unterlage zur Verfügung zu stellen. Denn gerade bei der Vielzahl von Handreichungen zu diesem Thema wird es für die Studierenden zunehmend unübersichtlich und schwierig, die existenziell wichtigen Punkte zu identifizieren und aus den Teils widersprüchlichen Angaben herauszufinden, welche Regelungen in ihrem Studiengang gelten und welche Schwerpunkte hier gelegt werden.

Deshalb wird im folgenden in möglichst kompakter Form auf die Gliederung wissenschaftlicher Arbeiten und Zitierweisen eingegangen. Das Thema Recherche wurde bewusst außen vorgelassen: Um hier konkrete Hinweise geben zu können, müssen die spezifischen Zugangsmöglichkeiten zu Literaturdatenbanken etc. für das Fach Software Engineering an der Hochschule Heilbronn berücksichtigt werden. Diese Möglichkeiten erweitern sich erfreulicherweise an unserer Hochschule derzeit relativ schnell, so dass aktuelle im Rahmen der Vorlesungsfolien darauf verwiesen wird.

Im vorliegenden Dokument wird in Kapitel 2 zunächst kurz die Zielsetzung dieser Arbeit skizziert, dann werden im Kapitel 3 stichwortartig die grundlegenden Anforderungen an wissenschaftliches Arbeiten dargestellt. Kapitel 4 geht auf die Gliederung wissenschaftlicher Arbeiten ein und zeigt auf, welche Teile vorhanden sein müssen und welche optional sind, außerdem wird skizziert, was sich hinter den einzelnen Elementen verbirgt. In Kapitel 5 werden grundlegende Informationen zu Zitaten und zum Zitieren gegeben, außerdem wird der APA-Stil vorgestellt, der im Studiengang Software Engineering verwendet werden sollte, falls es keine andere Instruktion oder Vorgabe gibt. Es folgen eine kurze Zusammenfassung und ein Ausblick in Kapitel 6. Die Anlagen umfassen Mustergliederungen und eine Checkliste zur Beurteilung wissenschaftlicher Arbeiten.

# Theoretische Grundlagen

## Virtual Reality

### Locomotion

## Operante Konditionierung

## Technologien

Für die Umsetzung des Projekts werden versch Technologien benutzt, die im Nachfolgenden erläutert werden.

### Unity 3D

Unity 3D ist eine Laufzeit- und für Spiele und andere 3D und anderer interaktive 3D-Grafik-Anwendungen. Ebenso das meistgenutzte Tool im Bereich VR-Entwicklung. In Szenen werden die 3D Umgebungen erstellt. Diese besteht zum größten Teil aus Objekten, Materialien bzw. Texturen, Animationen, Audio und Lichtquellen. So entsteht eine virtuelle Welt. Der Android Support ist sehr umfangreich, was der Entwicklung mit der Oculus Quest zugutekommt.

### Oculus Quest

[Alle Infos zur Oculus Quest - technische Daten, Preis und Laufzeit (vr-legion.de)](https://vr-legion.de/news/alle-infos-zur-oculus-quest-technische-daten-preis-und-laufzeit/)

Bei der Oculus Quest handelt es sich um eine mobile VR Brille. Sie kann also genutzt werden, ohne zusätzlich einen leistungsstarken Computer oder Konsole zu benötigen. Ebenso werden keine externen Sensoren oder Kameras für das Tracking gebraucht.

Die Auflösung des Displays ist 1.600x1.440 Pixel pro Auge mit einer Wiederholrate von 72Hz. Das Tracking passiert über die Brille selbst. Auf dieser befinden sich an der Vorderseite 4 Kameras, die mit Weitwinkel arbeiten. Mit 6DoF, in Englisch Six Degrees of Freedom, ist die Nutzung aller sechs Freiheitsgrade möglich. Somit stehen viele Steuermöglichkeiten zur Verfügung und eine Bewegung im virtuellen Raum gegeben. Dafür elementar ist die Bedienung mit den zwei dazugehörigen Controllern.

### Visual Studio

[Neues in Visual Studio 2019 | Microsoft Docs](https://docs.microsoft.com/de-de/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019)

Visual Studio 2019 ist eine Entwicklungsumgebung mit vielen Tools und Dienste für Entwickler. Die eingesetzte Programmiersprache ist hierbei C#.

Die Software eigenen Refactorings vereinfachen die Organisation des geschrieben Codes durch vorgeschlagene Aktionen. IntelliCode von Visual Studio gibt dem Entwickler Unterstützung mit dem Einsatz von künstlicher Intelligenz. Beinhaltend sind unter anderem die Bereitstellung von Codevervollständigung und die Suche nach Codeproblemen.

Mit der integrierten Codebereinigung ist es Möglich mit einem Klick Warnungen und Vorschläge zu ermitteln und zu beheben.

Mit Debugging kann präzise nach Laufzeitfehlern gesucht werden.

### Programmiersprache C#

[Einführung in C# - Leitfaden für C# | Microsoft Docs](https://docs.microsoft.com/de-de/dotnet/csharp/tour-of-csharp/)

[Programmiergrundlagen: Objektorientierte Programmentwicklung (entwickler.de)](https://entwickler.de/online/development/einfuehrung-programmierung-objektorientierte-programmentwicklung-197372.html)

C# ist eine objektorientierte und moderne Programmiersprache. Hier werden sogenannte Objekte zur Kapselung von Funktionalität und Daten genutzt. Dies wird erweitert durch die Möglichkeit der Vererbung und Polymorphie.

### Git Versionskontrolle

[Was ist Git? (dev-insider.de)](https://www.dev-insider.de/was-ist-git-a-850847/)

Git ist eine kostenlose Open Source Software. Sie dient bei der Softwareentwicklung zur verteilten Versionskontrolle. Die Nutzung wird heutzutage in Unternehmen und von privaten Entwicklern genutzt. Durch die Plattformunabhängigkeit lässt sich Git in fast jeder Entwicklungsumgebung integrieren und nutzen. Die erste Version des Tools wurde im Jahre 2005 veröffentlicht.

Der Nutzen einer Versionskontroller besteht darin, Änderungen verteilt dem Softwareprojekt hinzufügen zu können. Diese werden zusätzlich protokolliert und sind somit nachvollziehbar. Es kann zu einem späteren Zeitpunkt auch auf frühere Versionen zugegriffen werden.

Die Verwaltung geschieht über dezentrale Git-Repositories, im Gegensatz zu anderer Software zur Versionskontrolle, welche mit einer zentralen Datenbank arbeiten. Jeder Entwickler, der an einem Projekt mit Git arbeitet, besitzt eine lokale Kopie des Repositories. Diese wird durch Programmieren weiterentwickelt und die Änderungen mit den anderen Entwicklern des Projekts geteilt.

Ein weiterer großer Bereich ist das Aufteilen und Zusammenführen von verschiedenen Entwicklungsständen. Hierbei kann man ein Softwareprojekt in Unterverzweigungen, den Branchen aufteilen. Dies wird vor allem genutzt, um parallel zum Hauptzweig an verschiedenen Features zu arbeiten. Zum Schluss werden die Branches wieder mit dem Hauptzweig zusammengeführt.

Es gibt viele Webanwendungen, welche Git nutzen und weitere Funktionen hinzufügen. Die Verwaltung von Softwareprojekten wird damit noch übersichtlicher, zugänglicher, transparenter und für andere Entwickler online verfügbar. Der Entwicklungsfortschritt kann grafisch und in Statistiken dargestellt werden. Auch ein eigenes Wiki über das Projekt kann für die Community erstellt und öffentlich zur Verfügung gestellt werden. Die Webseite GitHub ist ein Beispiel für solch eine Plattform.

# Technische Umsetzung

## Einbindung der Oculus Quest in Unity 3D

Um selbst entwickelte grafische Anwendung bzw. Spiele in Unity 3D ausführen oder testen zu können, muss eine aktive Verbindung von der Oculus Quest zum Computer und der Entwicklungsumgebung hergestellt werden.

Um die Verbindung zum PC erfolgreich herstellen zu können, wird die Software Oculus Link benötigt. Diese wird vom Unternehmen selbst online angeboten. Erforderlich ist ebenso das Einrichten der Oculus Quest Brille mit einem Facebook Account.

In Unity 3D ist es nötig, den Oculus Support einzurichten. Dazu gibt es aktuell noch zwei Möglichkeiten, welche ihre Vor- und Nachteile bieten. Das veraltete XR-Plugin Tool ist schon standardmäßig integriert und bietet dazu die Möglichkeit, OpenVR zu nutzen. Dies wird benötigt, wenn weitere VR-Brillen in der Entwicklungsumgebung genutzt werden sollen. Ein Beispiel ist die HTCVive. Da die Umsetzung einzig mit der Oculus Quest durchgeführt wird, viel die Entscheidung auf das neue XR-Management. Dies befindet sich in ständiger Weiterentwicklung, somit wird in naher Zukunft ebenso OpenVR unterstützt werden. Dies könnte wichtig sein, falls das Projekt weiterentwickelt werden soll.

Ebenso benötigt wird das Asset Oculus Integration, welches sich über den integrierten Asset Store finden und importieren lässt. Darin befinden sich einige Prefabs und Skripte, die benötigt werden, um grundlegende Interaktionen wie das Umschauen und das Gehen in der Virtuellen Welt ermöglichen. Dazu muss das Prefab OVRPlayerController vom Assets Order per Drag and Drop in die Hierarchie der Szene gezogen werden. Zusätzlich wird das Skript CharacterCameraContraint benötigt, welches noch hinzugefügt werden muss.

Ist die Oculus Quest erfolgreich mit Oculus Link verbunden, kann die VR-Brille direkt in der Unity Entwicklungsumgebung über den Play Mode ausgeführt werden. Dies wird vor allem in der Entwicklung benötigt. Für die spätere Evaluation läuft die Anwendung jedoch Standalone, da ohne Kabel und angeschlossenen Computer ein freieres Bewegen der Probanden möglich ist. Dazu werden in den Grafikeinstellungen einige Änderungen vorgenommen und die Plattform wird zu Android gewechselt. Ebenso muss der Account als Developer freigeschalten werden und die Oculus Quest im Entwicklermodus laufen.

## Entwicklung des Malus

Dieser Teil wurde zunächst separat zum Hauptprojekt entwickelt und nach der Fertigstellung integriert. Ziel ist es, einen visuellen und akustischen Malus auszulösen, sobald der Proband vom Weg abkommt. Visuell sichtbar sein soll ein rotes, pulsierendes Blinken im kompletten Sichtfeld. Ebenso soll neben dem visuellen Reiz ein akustisches Piepen ertönen. Die Frequenz beide Stimuli ist zeitgleich und wird bei steigender Dauer schneller.

Für die Umsetzung in Unity 3D wurde das UI Objekt Canvas implementiert. Dies ist dazu da um Inhalte wie z.B. Bilder, Buttons für ein Menü oder Text anzeigen zu lassen. Eine Besonderheit in der Entwicklung einer VR-Anwendung im Vergleich zu einer normalen 3D Umgebung ist, dass das Canvas mit dem Kamera Objekt des OVRPlayerController verknüpft werden muss. Sonst besteht das Problem der fehlenden Anzeige in der VR-Brille.

## Entwicklung der Szenarios

## Datenerfassung

# Evaluation

# Fazit

In diesem Dokument wurden Hinweise zum formalen und inhaltlichen Aufbau von wissenschaftlichen Arbeiten im Studiengang Software Engineering an der Hochschule Heilbronn dargestellt: Es wurden verschiedene Elemente von wissenschaftlichen Arbeiten und ihre Anordnung in einer Gliederung vorgestellt sowie der Umgang mit Zitaten und Quellen erläutert. Diese Hinweise sind ein Teil der Lehre im Fach „Arbeitstechniken in der IT“ und erleichtern es den Studierenden, die Qualität ihrer wissenschaftlichen Arbeiten zu optimieren. Im Idealfall kann es so durch eine Automatisierung der Einhaltung der formalen Kriterien auch gelingen, die bei der Betreuung von wissenschaftlichen Arbeiten investierte Zeit der Dozierenden noch stärker auf inhaltliche Fragen zu fokussieren.

Rückmeldungen zu Inhalten dieses Dokuments werden gerne angenommen und für weitere Verbesserungen genutzt.

# Anhang A: Beispiele für die Gliederung von Diplomarbeiten

Die nachfolgenden Gliederungen stellen Vorschläge aus Riekert (2001, S. 38) dar.

## A.1 Literaturarbeiten

1. Überblick (oder: Zusammenfassung, „Executive Summary“, alles Wichtige für den „Manager“ oder Schnellleser)
2. Fragestellung (oder: Ziele, Ausgangspunkt, Motivation)
3. Übersicht über den Stand der Wissenschaft und Technik (Beschreibung der Lösungsansätze, Beispiele etc. in einzelnen Abschnitten)
4. Bewertung der einzelnen untersuchten Ansätze, Beispiele etc., Identifikation von Defiziten
5. Synthese: Erstellung einer Gesamtschau, allgemeine Prinzipien, Beschreibung einer eigenen Sicht auf das Problem, evtl. auch eigene Vorschläge
6. Zusammenfassung (Erklärung des Nutzens), Ausblick

Anhang: eventuell recherchierte Texte, Produktbeschreibungen, etc.

## A.2 Systementwicklungen

1. Überblick (oder: Zusammenfassung, „Executive Summary“, alles Wichtige für den „Manager“ oder Schnellleser)
2. Problemstellung (oder: Ziele, Ausgangspunkt), Vorgesehener Benutzerkreis, Bedürfnisse der Benutzer
3. Stand der Technik (Wie wird das Problem bisher gelöst, wo sind die Defizite)
4. Gewählter Lösungsansatz (allgemeines Prinzip, welche Werkzeuge, z.B. Programmiersprachen werden verwendet)
5. Beschreibung der durchgeführten Arbeiten
6. Ergebnis (z.B. Screenshots mit Erläuterungen)
7. Zusammenfassung (Erklärung des Nutzens), Ausblick

Anhang: evtl. (ausgewählte) Programmbeispiele

Evtl. Diskette/CD-ROM als Beilage

# Anhang B: Checkliste zur Bewertung wissenschaftlicher Arbeiten

Hier eine Checkliste zur Bewertung von wissenschaftlichen Arbeiten. Bei den Kriterien auf der ersten Seite handelt es sich um „K.O.-Kriterien“, d.h. sie müssen erfüllt sein, sonst ist mit Nicht-Bestehen oder massiver Abwertung in der Note zu rechnen.

|  |  |
| --- | --- |
| **Stil- und Sprachregeln** |  |
| Korrekte Rechtschreibung (durchgängig neue oder alte) |  |
| Korrekte Grammatik und Zeichensetzung |  |
| Schreibweisen von Begriffen werden durchgängig verwendet (z.B. nicht einmal Versionskontrolle und dann Versions-Kontrolle) |  |
| Keine Ich-Form verwendet, keine direkte Ansprache der Lesenden |  |
| Keine Umgangssprache verwendet |  |
| Ausdrücke, die aus mehreren Wörtern zusammengesetzt sind, sind verbunden (als ein Wort oder mit Bindestrich), wenn mindestens eines der Wörter deutsch ist (also z.B. Software-Engineering-Methoden, Serveranwendungen) |  |
| Wortwahl und Ausdrucksweise eindeutig verständlich, prägnant und treffend |  |
| Sätze klar, inhaltlich aussagefähig und in sich logisch |  |
| Satzverknüpfungen sind sprachlich und logisch korrekt, spiegeln in lückenloser Form dem Untersuchungsziel adäquate Gedankenabläufe |  |
| **Darstellungen und Verzeichnisse** |  |
| Darstellungen (Abbildungen, Tabellen) korrekt durchnummeriert und inhaltlich bezeichnet |  |
| Erforderliche Verzeichnisse (Inhalts-, Abkürzungs-, Symbol-, Abbildungs-, Tabellen-, Literatur-/Quellenverzeichnis) korrekt angelegt und an der jeweils richtigen Stelle der Arbeit platziert |  |
| **Formale Anforderungen** |  |
| Deckblatt, die Textvorlaufseiten, alle Textseiten und die Textnachlaufseiten in richtiger Aufteilung (Rand, Zeilenabstände) gut lesbar (Größe, Konturierung) gestaltet und in richtiger Form nummeriert |  |
| Eventuell vorgegebene Seitenzahl eingehalten |  |
| Eventuell geforderte eidesstattliche Erklärung korrekt verfasst, datiert und eigenhändig mit Vor- und Zunamen auf allen einzureichenden Exemplaren unterschrieben |  |
| **Literaturbearbeitung und Zitierweise** |  |
| Qualitativ angemessene Literatur in gebührendem Umfang herangezogen |  |
| Alle Quellenangeben im Verzeichnis und im Verzeichnis nur verwendete Quellen |  |
| Literatur korrekt ausgewertet (ohne Verfälschungen, auf letztem Stand, primär) |  |
| Kritische Auseinandersetzung mit der Literatur |  |
| Richtiges Zitieren (eindeutige Erkennbarkeit übernommenen und eigenen Gedankenguts, durchgängige Belegmethode, Seitenzahlen bei direkten Zitaten) |  |
| Adäquate Zitierweise (kein unnötiges Zitieren, Ausmaß wörtlichen Zitierens) |  |
| Vollständigkeit der Angaben zu den verschiedenen Quellen |  |

|  |  |
| --- | --- |
| **Fragestellung** |  |
| Fragestellung klar formuliert |  |
| Fragestellung themenadäquat, d.h. sie bezieht sich ausschließlich auf das Thema |  |
| Fragestellung dem Typ der wissenschaftlichen Arbeit entsprechend |  |
| **Gliederung** |  |
| Formal korrekte Gliederung (konsequente Gliederungs-Klassifikation, tatsächliche und vollständige Untergliederung, richtige Zuordnung von Ober- und Unterpunkten, Kriterienreinheit der Untergliederungen, angemessene Gliederungstiefe etc.) |  |
| Gliederung inhaltlich verständlich und in Bezug auf das Thema aussagefähig |  |
| **Behandlung der Fragestellung** |  |
| Arbeit zeigt keine themenfremden oder unnötigen Abschnitte |  |
| Alle relevanten Punkte bearbeitet, d.h. keine ausgelassen oder nur partiell behandelt |  |
| Argumentationen, Beleg- und Beweisketten (statt nur Behauptungen, Mutmaßungen oder Spekulationen) |  |
| Beleg- und Beweisketten gut belegt, lückenlos und schlüssig |  |
| Argumentation überzeugend und beweiskräftig |  |
| Wissenschaftliches Niveau, keine Selbstverständlichkeiten oder Trivialitäten |  |
| Keine ungerechtfertigte Wiederholungen |  |
| Prinzip der Nachvollziehbarkeit beachtet |  |
| **Ergebnisse** |  |
| Klare Formulierung der Ergebnisse |  |
| Ergebnisse beantworten die Fragestellung der Arbeit |  |
| Ergebnisse sind in sich widerspruchsfrei |  |
| Ergebnisse sind folgerichtig basierend auf Argumentationen, Beleg- und Beweisketten |  |
| **Definitionen, Prämissen, Untersuchungsdesigns** |  |
| Definitionspflichtige Begriffe klar und problemstellungsgemäß gefasst, konsequent durchgehalten |  |
| Prämissen und im Laufe der Arbeit vollzogenen Prämissenänderungen sind klar angezeigt und Prämissenunterschiede bei Literaturbezügen werden beachtet |  |
| Empirische Arbeit: Untersuchungs- und Auswertungsdesign klar und vollständig dargestellt |  |
| **Eigenständigkeit** |  |
| Eigenüberlegungen in Form eigener Ansätze, Umsetzungen eigener Ideen |  |
| Literaturlücken registriert und zu schließen versucht |  |
| Widersprüche und Fragwürdigkeiten in der Literatur herausgearbeitet, kommentiert und aufzulösen versucht |  |
| Eigenständigkeit hinsichtlich der Problembearbeitung |  |
| Eigenständigkeit hinsichtlich der Darstellung/Illustration, der Verdichtung und Verknüpfung des gesammelten Materials |  |
| Eigenständigkeit hinsichtlich der Wiedergabe und Kommentierung der Literatur |  |

# Quellenverzeichnis

**Verwendete Quellen**

Koch, Angela (o.J.). Hinweise zur Anfertigung der Diplomarbeit – Ein Leitfaden. Hochschule Heilbronn. http://www.hs-heilbronn.de/Members/koch/diplomarbeiten/LeitfadenfuerDiplomarbeiten2.pdf, Abruf 2007-09-30.

Lorenzen, Klaus F. (2002). Wissenschaftliche Anforderungen an Diplomarbeiten und Kriterien ihrer Beurteilung. Hochschule für Angewandte Wissenschaften Hamburg. http://www.bui.haw-hamburg.de/pers/klaus.lorenzen/ASP/wisskrit.pdf, Abruf 2007-09-30.

Lorenzen, Klaus F. (2003): Zitieren und Belegen in wissenschaftlichen Arbeiten. Hochschule für Angewandte Wissenschaften Hamburg. http://www.bui.haw-hamburg.de/pers/klaus.lorenzen/ASP/zitierenbelegen.pdf, Abruf 2007-09-30.

Riekert, Wolf-Fritz (2001). Diplomarbeiten u.a. wissenschaftliche Arbeiten / Theses. Hochschule der Medien Stuttgart. http://v.hdm-stuttgart.de/~riekert/theses/thesis.doc, Abruf 2007-09-30.

Schräder-Naef, Regula (2003). Rationeller Lernen lernen. Ratschläge und Übungen für alle Wissbegierigen. (21. Auflage) Weinheim: Beltz.

**Weiterführende Quellen:**

Burmester, Michael (2004). Zitieren – Skript für Propädeutik 1. Hochschule der Medien Stuttgart. http://v.hdm-stuttgart.de/%7Eriekert/lehre/zitieren.pdf, Abruf 2007-09-30.

Deininger, Marcus; Lichter, Horst; Ludewig, Jochen & Schneider, Kurt (2002). Studien-Arbeiten: ein Leitfaden zur Vorbereitung, Durchführung und Betreuung von Studien-, Diplom- und Doktorarbeiten am Beispiel Informatik. (4. Auflage) Stuttgart: Teubner.

Shaw, Mary (2003). Writing Good Software Engineering Research Papers: Minitutorial. In: Proceedings of th 25th International Conference on Software Engineering ICSE '03, IEEE Computer Society, p. 726-736

# Eidesstattliche Erklärung

Hiermit erkläre ich eidesstattlich, dass die vorliegende Arbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt wurde, insbesondere, dass ich alle Stellen, die wörtlich oder annähernd wörtlich oder dem Gedanken nach aus Veröffentlichungen und unveröffentlichten Unterlagen und Gesprächen entnommen worden sind, als solche an den entsprechenden Stellen innerhalb der Arbeit durch Zitate kenntlich gemacht habe, wobei in den Zitaten jeweils der Umfang der entnommenen Originalzitate kenntlich gemacht wurde. Ich bin mir bewusst, dass eine falsche Versicherung rechtliche Folgen haben wird.

Ort, Datum Unterschrift

1. Bitte beachten Sie, dass die in diesem Dokument verwendete direkte Ansprache der Leserin oder des Lesers in wissenschaftlichen Arbeiten ebensowenig vorkommen sollte wie die „Ich“-Form. Und mit dieser ersten und letzten Fußnote möchte ich Sie auch darauf hinweisen, dass Fußnoten in der Informatik absolut unüblich sind. [↑](#footnote-ref-1)