

---

# **Titel der Abschlussarbeit | Projektseminararbeit | Seminararbeit**

Untertitel der Arbeit

Diplom|Bachelor|Master|Staatsexamens|Seminararbeit  
im Studiengang Bezeichnung des Studienganges am  
Institut für Informatik und Wirtschaftsinformatik der  
Universität Duisburg-Essen

**Vorname\_Stud Nachname\_Stud  
Matrikelnummer**

**Essen, 2. April 2012**

Betreuer: Name\_Betreuer  
Erstgutachter: Name\_Erstgutachter  
Zweitgutachter: Name\_Zweitgutachter

## **Eidesstattliche Erklärung**

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe alle Stellen, die ich aus den Quellen wörtlich oder inhaltlich entnommen habe, als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Essen, am 2. April 2012

## **Zusammenfassung**

Dieses Dokument soll einige Hilfestellungen zu Aufbau, Gestaltung und Inhalt Ihrer Abschlussarbeit geben. Bitte beachten Sie, dass es Vorlagen für Word und LATEX gibt, die bereits die passenden Formatvorlagen definieren oder leicht angepasst werden können. An dieser Stelle muss eine Zusammenfassung Ihrer Abschlussarbeit auf Deutsch stehen, die maximal eine halbe Seite lang ist.

## **Abstract**

In addition to the German “Zusammenfassung”, an English abstract is required as well. Authors writing their thesis in English should swap the positions of Abstract and Zusammenfassung.

## **Danksagung**

Dieses Dokument entstand aus einer Formatvorlage für Diplomarbeiten an der TU Darmstadt aus der Arbeitsgruppe von Prof. Dr. Mühlhäuser, der dankenswerter Weise seine Zustimmung zur Weiterverwendung und Erweiterung dieser Vorlage gegeben hat. Inzwischen sind zahlreiche Änderungen und Erweiterungen eingeflossen, welche die Informatik-Studiengänge am Campus Essen berücksichtigen.

# Abbildungsverzeichnis

# Tabellenverzeichnis

1	Möglicher Zeitplan für eine Abschlussarbeit (6 Monate Dauer) . . .	2
2	Die für ihre Arbeit relevanten Dateien . . . . .	13
3	Maketargets . . . . .	14
4	Positionierungsparameter für Floats . . . . .	18

# Inhaltsverzeichnis

<b>Eidesstattliche Erklärung</b>	<b>I</b>
<b>Zusammenfassung</b>	<b>II</b>
<b>Abstract</b>	<b>III</b>
<b>Danksagung</b>	<b>IV</b>
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Tabellenverzeichnis</b>	<b>VI</b>
<b>1 Organisatorische und rechtliche Hinweise</b>	<b>1</b>
1.1 Allgemein . . . . .	1
1.2 Richtlinien für Abschlussprüfungen . . . . .	1
1.2.1 Durchführung der Abschlussarbeit . . . . .	1
1.2.2 Vorgeschriebene Form . . . . .	1
1.2.3 Einreichen der Arbeit . . . . .	2
1.3 Zeitplan . . . . .	2
<b>2 Die schriftliche Ausarbeitung</b>	<b>3</b>
2.1 Inhalt der Arbeit . . . . .	3
2.2 Aufbau der Arbeit . . . . .	5
2.3 Zur Leserschaft der Arbeit . . . . .	6
2.4 Vorbereitung . . . . .	6
2.5 Plagiatismus und gute wissenschaftliche Praxis . . . . .	7
2.6 Stil . . . . .	8
2.7 Schwarze Liste . . . . .	9
2.8 Zahlen, Ziffern und Maßangaben . . . . .	10
2.9 Überprüfen auf Fehler . . . . .	11
<b>3 Gestaltung</b>	<b>13</b>
3.1 Dateien . . . . .	13
3.2 Verwendung des Makefiles . . . . .	14
3.3 Verwendung von $\text{\LaTeX}$ . . . . .	16
3.3.1 Dokumentenstruktur . . . . .	16
3.3.2 Aufzählungen . . . . .	16
3.3.3 Labels . . . . .	17
3.3.4 Floats . . . . .	17
3.4 Allgemeines Schriftbild . . . . .	21
3.5 Präsentation der Arbeit . . . . .	21
<b>4 Arbeiten mit Literatur</b>	<b>24</b>
4.1 Literatursuche . . . . .	24



4.2	Literaturbeschaffung . . . . .	25
4.3	Literaturverweise . . . . .	25
4.3.1	Erstellen einer Literaturdatenbank . . . . .	25
4.3.2	Zitieren aus der Literaturdatenbank . . . . .	26
<b>5</b>	<b>Weitere Aspekte</b>	<b>28</b>
5.1	Themenfindung . . . . .	28
5.2	Deutsch oder Englisch? . . . . .	28
5.3	Externe Arbeiten . . . . .	28
5.4	Bezahlung . . . . .	28
5.5	Rechte an den Ergebnissen . . . . .	29
5.6	Projektberichte/Seminarausarbeitungen . . . . .	29
	<b>Literaturverzeichnis</b>	<b>30</b>

# 1 Organisatorische und rechtliche Hinweise

Diese Anleitung wendet sich in erster Linie an Bachelor- und Masterabsolventen am ICB. Ziel der Anleitung ist es, den Absolventen Hinweise und Hilfestellungen zum Verfassen von schriftlichen Ausarbeitungen, insbesondere ihrer Abschlussarbeit zu geben.

## 1.1 Allgemein

Die Abschlussarbeit ist für die meisten Studierenden das erste umfangreiche Werk, das sie im Lauf des Studiums verfassen. In der Informatik ist die Abschlussarbeit oft mit der Entwicklung eines Programms verbunden.

Diese Anleitung geht davon aus, dass Sie auch ein Programm erstellen und in Ihrer Abschlussarbeit beschreiben. Theoretische Arbeiten oder Literaturarbeiten (die in der Informatik unüblich sind) erfordern zum Teil einen anderen Aufbau; einige der hier angeführten Hinweise sind deshalb für derartige Arbeiten nicht in vollem Umfang verwendbar.

## 1.2 Richtlinien für Abschlussprüfungen

Achten Sie auf die Bestimmungen der für Sie gültigen Prüfungsordnungen. Hier werden ergänzende Hinweise gegeben.

### 1.2.1 Durchführung der Abschlussarbeit

Laut den üblichen Klauseln in den Prüfungsordnungen ist die Abschlussarbeit eine schriftliche Prüfungsarbeit, in der die Studierenden den Nachweis erbringen sollen, dass sie in der Lage sind, Probleme der Informatik selbständig zu bearbeiten. Nach den Prüfungsordnungen beträgt die Frist für die Anfertigung der Abschlussarbeit drei, vier oder sechs Monate, je nachdem ob es sich um eine Bachelor-, Diplom- oder eine Masterarbeit handelt. Beachten Sie, dass die jeweilige Frist nur die eigentliche Abschlussarbeit umfasst und nicht die im Rahmen der Arbeit nötige Einarbeitung in Werkzeuge oder existierende Programme.

Mit Zustimmung des Prüfungsausschusses können Sie Ihre Abschlussarbeit auch außerhalb der Universität durchführen. In einem solchen Fall muss allerdings die Betreuung durch einen Hochschullehrer gesichert sein. Sollten Sie also eine externe Arbeit durchführen wollen, sprechen Sie sich zuerst mit Ihrem gewünschten Betreuer ab.

### 1.2.2 Vorgeschriebene Form

Die Abschlussarbeit ist eine wissenschaftliche Arbeit. Sie müssen sie daher mit einem Verzeichnis aller benutzten Quellen und Hilfsmittel versehen.

Sie müssen Ihre Arbeit ebenfalls mit einer Erklärung versehen, dass Sie die Arbeit selbständig verfasst haben. Fügen Sie diese Erklärung am besten ganz am Ende ein (oder auch wie in dieser Vorlage am Anfang). Derzeit üblich und künftig verbindlich ist auch die zusätzlich Abgabe in digitaler Form (CD, DVD), ordentlich beschriftet und unter Verwendung einer Papier- oder Plastikhülle in die Arbeit eingeklebt.

### 1.2.3 Einreichen der Arbeit

Sie müssen drei Exemplare Ihrer Abschlussarbeit beim Prüfungsamt fristgerecht und in gebundener Form einreichen. Je ein Exemplar geht an Erstgutachter und Zweitgutachter, die innerhalb von ca. 6-8 Wochen ihre Gutachten erstellen sollen. Das dritte Exemplar dient zur Dokumentation Ihrer Prüfungsleistung und wird in der Regel mit den Prüfungsakten aufbewahrt.

## 1.3 Zeitplan

Der folgende Zeitplan soll nur einen groben Rahmen vorgeben und bezieht sich auf eine Abschlussarbeit mit Implementierungsanteilen von 6 Monaten Dauer. Soweit sinnvoll, können sich einzelne Phasen auch überschneiden und geringfügig im zeitlichen Rahmen ändern. Zusätzlich sollten regelmäßige Treffen mit den Betreuern vorgesehen werden, um den Fortschritt und mögliche Probleme zu besprechen.

Wenn Sie wenig Implementierungserfahrung haben (was eigentlich nicht der Fall

Monat	Aufgabe	Anmerkung
0	Einarbeitung	Mögliche Einarbeitung in Werkzeuge, Programmiersprachen, etc.
Start	Besprechung der Aufgabenstellung	
1		Beginn mit der Literaturarbeit: Einlesen in Standards und vergleichbare Forschungsarbeiten.
2		Ergebnisse der Literatursuche, Beginn der Entwicklung von Konzepten, Architektur und ähnliches.
2 (Mitte)	Beginn der Implementierung	
3 (Ende)	Prototyp	
5 (Mitte)	Implementierung abgeschlossen	
6 (Mitte)	Vortrag	Vortrag über die Ergebnisse der Abschlussarbeit, ca. 30 Minuten, davon 10 Minuten Diskussion.
6 (Ende)	Abgabe der schriftlichen Arbeit beim Prüfungsamt	Die Gutachter erstellen in 6-8 Wochen das Erst- und das Zweitgutachten.

Tabelle 1: Möglicher Zeitplan für eine Abschlussarbeit (6 Monate Dauer)

sein darf!), sollten Sie schon während der Einarbeitung prüfen, wie Sie mit den technischen Teilen vorankommen.

## 2 Die schriftliche Ausarbeitung

Die schriftliche Ausarbeitung dient folgenden Zwecken:

- Erläuterung der Aufgabenbeschreibung und der Zielsetzung
- Beschreibung des Grobkonzeptes, der Lösungsansätze, ggf. der entwickelten Algorithmen oder der Systemarchitektur (dies sind Beispiele)
- Eine Bedienungsanleitung für das im Rahmen der Abschlussarbeit entstandene Programm zu bieten
- Als Dokumentation für das Programm zu dienen, um damit den Grundstein für weitere Arbeiten daran zu setzen
- Als eine Grundlage für die Beurteilung durch den Betreuer zu dienen

Diese Anleitung enthält Empfehlungen, Richtlinien und Tips für die Vorbereitung, Gliederung und Gestaltung. Diese Regeln sind nicht verbindlich, ihre Einhaltung kann aber die Qualität der Ausarbeitung verbessern und damit die Wahrscheinlichkeit einer guten Beurteilung erhöhen.

### 2.1 Inhalt der Arbeit

Die Abschlussarbeit muss alles enthalten, was für das Verständnis Ihrer Arbeit erforderlich ist. Sie soll vollständig sein, kann aber Grundlagenwissen voraussetzen. Der vollständige Programmtext des erstellten Systems hat in der Abschlussarbeit nichts zu suchen. Allenfalls sollte der Quelltext mit Dokumentation in Absprache mit dem Betreuer getrennt auf CD/DVD oder anders abgegeben werden.

Versuchen Sie, das in Ihrer Abschlussarbeit beschriebene System von mehreren Seiten zu beleuchten. Insbesondere sollten die folgenden Punkte behandelt werden:

- **Aufgabe:** Worin bestand die gestellte Aufgabe? In welches Teilgebiet der Informatik gehört sie? Wurden die in der Aufgabe gestellten Anforderungen erfüllt, übertroffen, oder mussten während der Arbeit Abstriche gemacht werden?
- **Überblick:** Beschreiben Sie die wesentlichen Funktionen des Programms. Wozu dient es? Durch welche besonderen Eigenschaften zeichnet es sich aus? Was leistet das Programm, was leistet es nicht?
- **Benutzung:** Stellen Sie das Programm aus der Sicht eines Benutzers dar. Zeigen Sie an konkreten Beispielen, wie es zur Lösung einer Aufgabe verwendet werden kann. Vermeiden Sie langwierige Erklärungen von Menübefehlen und ähnlichen Details der Bedienung.
- **Ergebnisse:** Zeigen Sie an einfachen Beispielen, welche Ergebnisse Ihr Programm liefert.
- **Vergleich mit anderen Programmen:** Gibt es bereits Programme, die ähnliches leisten? Worin bestehen die wesentlichen Unterschiede zu Ihrer Arbeit? Was hat Ihr Programm Neues zu bieten?

- **Konzepte:** Beschreiben Sie die hinter der Implementierung stehenden Grundlagen und technischen Konzepte, auf die Sie aufbauen. Geben Sie eine Gliederung des Programms an und stellen Sie Funktionsblöcke und Datenfluss dar. Bei objektorientierten Programmen gehört hierher auch ein Überblick über die Klassenhierarchie und evtl. über das Zusammenspiel der Objekte zur Laufzeit.
- **Schnittstellen:** Beschreiben Sie Berührungspunkte mit der Außenwelt. Zeigen Sie, wie Ihr Programm mit anderen Programmen zusammenarbeitet, wie seine Ergebnisse in anderen Programmen weiterverarbeitet werden können.
- **Implementierungsdetails:** Wählen Sie ein paar interessante Detailprobleme aus und beschreiben Sie deren Lösung. Achten Sie dabei darauf, die Rolle der Details für das Gesamtsystem darzustellen, damit der Leser die Zusammenhänge erkennen kann.
- **Anleitung für Erweiterungen:** Zeigen Sie Erweiterungsmöglichkeiten auf und beschreiben Sie, wie Erweiterungen und Verbesserungen angebracht werden können. Besonders anschaulich fallen Erweiterungsanleitungen aus, wenn sie an einem Beispiel erklärt werden.
- **Technische Daten:** Führen Sie Maßzahlen an, die dem Leser die Abschätzung von Umfang und Leistungsfähigkeit Ihres Programms erlauben. Beispiele für solche technischen Daten sind Programmumfang, Laufzeiten und Speicherbedarf. Stellen Sie die Daten grafisch dar. Erleichtern Sie dem Leser das Verständnis der Daten, indem Sie sie interpretieren. Begründen Sie Unterschiede und Ausreißer, weisen Sie auf besonders gute und besonders schlechte Ergebnisse hin, und zeigen Sie die Grenzen des Programms auf.
- **Kritische Würdigung:** Versuchen Sie, Ihre eigene Arbeit objektiv zu beurteilen. Worauf sind Sie besonders stolz, was könnte verbessert werden, was würden Sie anders machen, wenn Sie dieselbe Arbeit noch einmal angehen würden? Welche Werkzeuge haben Sie für die Entwicklung benutzt, wie haben sie sich bewährt?

Die Reihenfolge der hier angeführten Punkte hat sich bei der Beschreibung von Arbeiten, deren Augenmerk auf der Programmierung lag, bewährt, muss aber nicht sklavisch eingehalten werden.

Beim Schreiben der Abschlussarbeit ist es nützlich, wenn Sie auf die erbrachte Leistung stolz sind und Ihre Begeisterung dem Leser vermitteln wollen. Eine rein technische Beschreibung liest sich manchmal langweilig. Ein lebendiger Schreibstil, der den Leser mit rhetorischen Fragen in Probleme und ihre Lösung einbezieht, kann eine Programmbeschreibung zu einem spannenden Werk machen. Wenn Sie sich stilistisch nicht absolut sicher fühlen, so behalten Sie besser einen „defensiven“ und technisch-orientierten Schreibstil bei. Sie sollen schließlich keinen Unterhaltungsroman schreiben!

## 2.2 Aufbau der Arbeit

Die Ausarbeitung sollte in deutscher oder englischer Sprache verfasst werden. Eine typische Abschlussarbeit mit 6-monatiger Bearbeitungsdauer umfasst etwa 25.000-35.000 Worte (etwa 75-100 Seiten). Beachten Sie dabei, dass dies lediglich Richtwerte sind und im Zweifelsfall eine exakte und knappe Ausarbeitung überflüssigem Geschwafel vorzuziehen ist. Bei einer Bearbeitungsdauer von 3 Monaten (wie bei einer Bachelorarbeit) reduziert sich der Umfang auf etwa 15.000-18.000 Worte (40-50 Seiten).

Die Ausarbeitung sollte folgende Struktur aufweisen:

1. Inhaltsverzeichnis (inkl. Verzeichnisse von Abbildungen, Tabellen, etc.)
2. Einleitung und Motivation: Warum gibt es dieses Abschlussarbeitsthema? Beschreibung der Problemstellung, gestellte Aufgabe, Ergebnisse, Überblick über die Abschlussarbeit (Kurzbeschreibung der Kapitel).
3. Grundlagen: Theoretische und technische Grundlagen der Arbeit, verwandte Arbeiten, Berührungspunkte zu anderen Fachgebieten, verwendete Ansätze, Methoden und/oder Modelle (Sprachen, Entwurfsmethoden, Datenmodelle, Analysemethoden, Formalismen).
4. Benutzung: Gedankenmodell des Programms, typischer Programmablauf, gelieferte Ergebnisse.
5. Programmstruktur: Design des Programms, Gliederung des Programms in Funktionsblöcke, Programmhierarchie, Datenfluss, Schnittstellen.
6. Implementierung: Technische Beschreibung der Problemlösung, verwendete Datenstrukturen, interessante Algorithmen.
7. Technische Daten: Skalierbarkeit der Lösung, Angaben zur Laufzeit des Programms, Programmgröße und Speicherbedarf.
8. Kritische Würdigung: Vergleich mit verwandten Arbeiten, Diskussion offener Punkte und insbesondere auch Ihre persönliche Bewertung.
9. Zusammenfassung
10. Literaturverzeichnis
11. Anhänge

Die hier angeführte Gliederung ist nicht unbedingt äquivalent zur Kapitelstruktur Ihrer Arbeit. So kann es beispielsweise sinnvoll sein, die Implementierung in mehrere Kapitel zu unterteilen oder die technischen Daten und die Beurteilung zu einem Kapitel zusammenzufassen. In Einzelfällen kann auch eine andere Reihenfolge angebracht sein, beispielsweise wenn eine Klassenbibliothek entwickelt wurde, deren Benutzung Kenntnis ihrer Implementierung voraussetzt.

Achten Sie auf jeden Fall darauf, dass grundlegende Punkte zuerst behandelt werden, so dass Sie in späteren Kapiteln darauf aufbauen können.

Sollten Sie in Ihrer Arbeit weitere allgemeine Konventionen als die in Kapitel 3 empfohlenen Gestaltungsrichtlinien verwenden, so sollten Sie diese in einem Vorwort einführen.

Ob Anhänge sinnvoll sind, hängt vom Thema Ihrer Arbeit ab. Kurze Programmstücke (bis ca. 2 Seiten) können direkt in die entsprechenden Kapitel eingesetzt werden. Lange Beispielprogramme, die den Lesefluss stören würden und für das Verständnis eines Kapitels aber unbedingt erforderlich sind, sollten in einen Anhang verbannt werden. Bei Bildschirmmasken, Klassendiagrammen oder ähnlichen liegt die Grenze bei etwa zwei Abbildungen pro Seite.

## 2.3 Zur Leserschaft der Arbeit

Die Arbeit richtet sich an fortgeschrittene Studierende der Informatik oder informatiknaher Fächer, die mit den Grundlagen vertraut sind, jedoch nicht mit dem speziellen Thema. Denken Sie immer an den Leser; versuchen Sie sich in die Lage eines Kollegen zu versetzen, der mehr erfahren möchte. Unnötige Grundlagen und ausschweifende Erklärungen langweilen den Leser. Zu knappe Erklärungen wesentlicher Dinge hinterlassen das Gefühl der Unvollständigkeit.

Tatsächlich wird ihre Arbeit *typischerweise nur von den Gutachtern/Betreuern und Bearbeitern paralleler oder später aufbauender Abschlussarbeiten gelesen*, nicht hingegen von Personen, die den Inhalt der einschlägigen Standardvorlesungen nicht kennen. Der Inhalt sollte auf eine Leserschaft hin ausgerichtet sein, welche diese Vorkenntnisse hat. Von daher ist es überflüssig, diesen Stoff mehr oder weniger langatmig aufzubereiten; entsprechende Abschnitte der Arbeit können das Ergebnis nur verschlechtern. Wo möglich, sollte bei allgemeinen Begriffsdefinitionen auf eine allgemein zugreifbare Quelle (z.B. Lehrbuch oder vorhandene Dokumentation) verwiesen werden. Standardstoff sollte nur dann wiederholt werden, wenn dies wegen aufbauender spezieller Themen unbedingt notwendig ist.

Die Arbeit soll also vollständig sein, kann aber Grundlagenwissen voraussetzen. Das richtige Maß zwischen Kürze und Ausführlichkeit zu finden ist eine Kunst. Versuchen Sie nicht, durch leeres Geschwätz, aufgeblähte Bildschirmschnappschüsse oder seitenlange Programmlisten Seiten zu schinden.

## 2.4 Vorbereitung

Es ist empfehlenswert, die Teile der Arbeit in derselben Reihenfolge zu schreiben, in der sie später gelesen werden. Auf diese Weise steigt die Wahrscheinlichkeit, dass die Arbeit flüssig gelesen werden kann, dass sie keine Brüche enthält und keine unbekannten (z.B. erst später definierten) Begriffe verwendet werden. Allerdings setzt diese Arbeitsweise voraus, dass zuvor ein Plan der schriftlichen Arbeit ausgearbeitet wird.

Stellen Sie als erstes das Inhaltsverzeichnis auf. Beschränken Sie sich auf ein- und zweistufige Überschriften, und ergänzen Sie jeden Abschnitt um die geschätzte Seitenanzahl sowie Stichworte zum Inhalt.

Die Schätzung des Umfangs gibt einen Anhaltspunkt für die Schwerpunkte der Arbeit. Wenn die Seitenanzahl des Hauptteils unter der von nebensächlichen Abschnitten (z.B. der Einleitung) liegt, sollten Sie die Gliederung neu überden-

ken. Wenn einzelne Kapitel zu umfangreich werden, sollten Sie eine Aufteilung in kleinere Kapitel in Betracht ziehen.

Die Stichworte sollten so gewählt werden, dass Sie sich beim Schreiben noch daran erinnern, was Sie sich dabei gedacht haben. Empfehlenswert sind auch rhetorische Fragen, die Sie im betreffenden Abschnitt beantworten wollen. Überprüfen Sie anhand der Stichworte, ob die einzelnen Kapitel und Abschnitte aufeinander aufbauen, so dass möglichst selten Vorwärtsverweise auf folgende Kapitel notwendig werden.

Es ist empfehlenswert, schon während der Implementierungsarbeit Notizen zu sammeln, in denen der Projektverlauf, aufgetretene Probleme und ihre Lösungen festgehalten werden. Es ist auch nützlich, über Irrwege Buch zu führen. Jeder falsche und später verworfene Lösungsansatz vermittelt Einsichten, die dem Leser der Abschlussarbeit von Nutzen sein können.

## **2.5 Plagiatismus und gute wissenschaftliche Praxis**

Plagiatismus ist ein großes Problem, das Sie ernst nehmen müssen. Es ist zwar meistens unvermeidlich, dass Sie auf existierenden Ergebnissen aufbauen, diese müssen aber ordentlich zitiert sein und durch Referenzen abgedeckt werden. Außerdem darf dieser reproduzierte Anteil nicht zu groß sein; wenige Seiten mit reproduktivem Charakter sollten meistens genügen.

Beachten Sie bitte, dass Plagiatismus zur Abwertung der Arbeit dient, wobei in der Regel die Note "nicht bestanden" (5,0) vergeben werden muss. Je nach Umfang des Plagiatismus kann (oder muss) der zuständige Prüfungsausschuss damit befasst werden, der weitere Sanktionen verhängen kann. Im Prinzip ist sogar eine Exmatrikulation möglich oder (als Variante) der Ausschluss von allen weiteren Prüfungsversuchen.

Sollten Sie aus anderen Quellen Bestandteile oder Zitate in Ihre Arbeit übernehmen, die dabei nötigen Quellenangaben nicht oder grob unvollständig machen, verstoßen Sie damit gegen die gute wissenschaftliche Praxis. Die übernommenen Bestandteile sind damit Plagiate und gelten als Betrug.

Auf keinen Fall ist es statthaft,

- wörtlich abzuschreiben oder via cut & paste Passagen zu kopieren, ohne solche Abschnitte eindeutig als Originalzitate zu kennzeichnen. Solche Zitate können sinnvoll sein, dürfen aber nicht zum Aufblähen einer Arbeit zum Zwecke des Erreichens eines vorgegebenen oder vermuteten „Soll-Umfangs“ dienen.
- Zitate abzuändern, damit sie nicht mehr als Originalzitate gekennzeichnet werden müssen oder nicht mehr so leicht mittels Suchmaschinen aufzufinden sind. Inzwischen gibt spezielle elektronische Hilfsmittel zum Aufspüren von sogenannten Strukturplagiaten gibt.
- Ideen und Konzepte zu übernehmen, ohne Urheber zweifelsfrei zu benennen.

Wie Sie Zitate korrekt kennzeichnen, können Sie in Kapitel 4.3 nachlesen.



Zur Verdeutlichung der Problematik ein Zitat aus einer Resolution des Deutschen Hochschulverbandes: *“Schriftliche Arbeiten Studierender im Rahmen der universitären Ausbildung haben dem Anspruch guter wissenschaftlicher Praxis zu genügen. In einigen wissenschaftlichen Disziplinen stellt sich ein Teil der Studierenden diesem Anspruch nicht, sondern legt statt einer schriftlichen Arbeit, die auf eigener geistiger Leistung beruht, ein Plagiat vor, indem Texte Dritter ganz oder teilweise, wörtlich oder nahezu wörtlich übernommen und als eigene wissenschaftliche Leistung ausgegeben werden. Ein solches Vorgehen widerspricht nicht nur guter wissenschaftlicher Praxis, es ist auch eine Form des geistigen Diebstahls und damit eine Verletzung des Urheberrechts.”*

<http://www-e.uni-magdeburg.de/sturafwu/studium/prof-plagiat/plagiate.pdf>

## 2.6 Stil

Die inhaltlich beste Arbeit ist schwierig zu lesen, wenn man von einem mangelhaften Stil abgelenkt wird. Ein guter Schreibstil kann leider nicht im Schnellverfahren erlernt werden, sondern wird mühsam durch Erfahrung und Übung erworben. In der Schule wird zwar das Schreiben von Aufsätzen geübt, das technische Schreiben bleibt aber auf der Strecke. Die folgenden Hinweise sollen daher einige Anhaltspunkte für die Ausarbeitung bieten. Wenn Sie Ihre Arbeit auf Englisch verfassen, finden Sie auf den Webseiten der *Plain English Campaign* [1] weitere Anregungen für einen guten Schreibstil.

Benutzen Sie eine einfache, sachliche und klare Sprache. Vermeiden Sie geschwollene Ausdrucksweisen, “wohlklingende” Fremdwörter, zu lange und geschachtelte Sätze, die man zweimal lesen muss, um ihren syntaktischen Aufbau zu durchschauen, sowie lange Aufzählungen wie in diesem Satz.

Winzige Abschnitte mit nur wenigen Zeilen rechtfertigen keine eigene Überschrift. Überdenken Sie in solchen Fällen die Gliederung neu. Vermeiden Sie zu tiefe Schachtelung der Kapitel. Dreistufige Überschriften (1.2.3) reichen in der Regel.

Auch innerhalb eines Abschnitts können und sollten Sie durch Absätze, Einrückungen und nicht nummerierte Überschriften für Struktur sorgen. Auf jeden Fall sollten lange Absätze in mehrere kleinere zerlegt werden.

*Wortwiederholungen* sind in literarischen Werken verpönt, in technischen Beschreibungen aber oft für das Verständnis wesentlich. Wenn in einer Arbeit die Wendungen “Eingabefenster”, “aktives Fenster”, “vorderstes Fenster” und “aktuelles Fenster” nebeneinander vorkommen, so vermutet der Leser einen Sinn dahinter und versucht vergeblich, die Unterschiede zu ergründen. Benutzen Sie daher konsequent denselben Begriff für dieselbe Bedeutung. Noch besser: Erklären Sie den Begriff bei oder vor seinem ersten Auftreten.

Wenn Sie Ihre Arbeit auf Deutsch verfassen, versuchen Sie, für englische Begriffe deutsche Entsprechungen zu finden (etwa “Datei” statt “File”). Schießen Sie dabei aber nicht über das Ziel hinaus. Manche Begriffe (etwa “Task” und “Thread”) sind schwer zu übersetzen, ohne den Sinn zu verfälschen. In diesen Fällen empfiehlt es sich, die eingebürgerten englischen Begriffe zu verwenden.

*Vermeiden Sie Jargon*, auch wenn es toll oder "cool" klingt und "in" ist. Beispielsweise lässt sich "generieren" durch "erzeugen" ersetzen, "handeln" (auch: "handlen") durch "behandeln" oder "bearbeiten". Ein anderes oft verwendetes Beispiel ist "updaten". Heißt das in der Vergangenheit "geupdatet" oder "upge-datet"? Diese Frage erübrigt sich, wenn man gleich "aktualisieren" verwendet.

*Gehen Sie sorgsam mit Abkürzungen um*. Wer weiß schon, was ADSP, TCL, LLC und PST bedeuten? Und wie spricht man sie aus ("Pst", "Pe-Es-Te" oder "Pi-Es-Ti")? Abkürzungen sind sinnvoll, wenn sie sich (wie etwa TCP und FTP) eingebürgert haben und ihre Langform zu umständlich wäre. In den meisten anderen Fällen sollten sie – zumindest beim ersten Auftreten – lieber ausgeschrieben werden. In jedem Fall sollte die Langform beim ersten Auftreten verwendet werden, gefolgt von der Abkürzung in Klammern: "Das Transmission Control Protocol (TCP) ...".

Ihre Arbeit muss nicht unbedingt ein Abkürzungsverzeichnis enthalten, oft ist das für den Leser dennoch sehr hilfreich.

Die *Ich-Form* gilt in technischen Werken als schlechter Stil. Versuchen Sie, auf unpersönliche Formulierungen auszuweichen (z.B. „Das System zerfällt in ...“ statt „Ich habe das System in ... zerlegt“). Bei der persönlichen Beurteilung der Arbeit kann die Ich-Form jedoch verwendet werden. Vermeiden Sie auch den Plural („Kommen wir nun zu...“), es sei denn, Sie stellen wirklich die Arbeit Ihrer Gruppe dar.

*Vermeiden Sie Füllwörter*. Durchsuchen Sie Ihre fertige Arbeit nach Wörtern wie „sehr“, „eigentlich“, „grundsätzlich“, „prinzipiell“, „ziemlich“, „relativ“, „nun“, „nämlich“ und „konkret“. Sie werden sehr bald feststellen, dass Sie diese relativ häufigen Wörter eigentlich grundsätzlich weglassen können, da sie eigentlich nämlich ziemlich nichtssagend sind. Ihre Aussagen werden dadurch exakter und schärfer.

## 2.7 Schwarze Liste

Einige Wörter und Formulierungen tauchen immer wieder in technischen Arbeiten (und leider auch in Werken angesehener Autoren) auf. Manche davon sind Modewörter, die grässlich klingen, falsch aus dem Englischen übersetzt sind oder in unpassendem Zusammenhang verwendet werden. Die folgende Liste enthält einige dieser Wörter mit Vorschlägen für bessere Formulierungen.

- „beinhalten“ ist ein grässliches Wort, vor allem wenn man es geschrieben sieht. Es kann durch „enthalten“, „umfassen“, „einschließen“ oder „abdecken“ ersetzt werden.
- Das ebenso hässliche Modewort „kreieren“ kann durch „erzeugen“ oder ein anderes der im folgenden Absatz angeführten Wörter ersetzt werden.
- „unterstützen“ bedeutet so viel wie „helfen“. Wenn ein Werkzeug eine Funktion „unterstützt“, dann hilft es nicht der Funktion, sondern es bietet die Funktion oder führt sie aus.

- „Instanz“ wird oft als Übersetzung des englischen „instance“ verwendet. Die richtige Übersetzung im objektorientierten Sinn lautet „Exemplar“, „Element“ oder einfach „Objekt“.
- „Kontrolle“ bedeutet vor allem Überprüfung. Leider kommt es oft im Sinne des englischen „control“ vor, das jedoch „Steuerung“ bedeutet.
- „besitzen“ klingt zwar vornehmer als das banale Wörtchen „haben“, es deutet aber auf Eigentum (des Besitzers) hin. Dinge haben kein Eigentum; sie können also nichts besitzen. Statt „Das Objekt besitzt einen Verweis auf. . .“ sollte es daher einfach „hat“ oder „enthält“ heißen.
- „Speicherverbrauch“ ist eine schlimme Sache. Wenn ein Programm tatsächlich Speicher verbraucht (wie ein Auto Benzin verbraucht), dann wird der Speicher des Computers immer weniger. Gemeint ist natürlich „Speicherbedarf“; ein Programm benötigt oder belegt Speicher.
- „Paradigma“ ist ein gewichtiger Begriff, den man besser meiden sollte. Leider hat es sich als Ausdruck für bestimmte Programmierstile und Entwicklungstechniken eingebürgert. Noch schlimmer wird es, wenn von einem „objektorientierten Programmierparadigma“ die Rede ist, denn das „Paradigma“ ist nicht objektorientiert (vergleiche „vierstöckiger Hausbesitzer“).
- Anglizismen sind in der Informatik oft nicht zu vermeiden; wenn es gute deutsche Begriffe gibt, sollten diese vorzugsweise verwendet werden. Statt garbage collection, hashing, queue und stack kann man auch Speicherbereinigung, Streuschlüsselverfahren, Schlange und Stapel verwenden. Es gibt natürlich viele Begriffe wie Client/Server, Grid, Peer-to-Peer, Routing, Thread, usw., usf., die man besser nicht übersetzen sollte.

## 2.8 Zahlen, Ziffern und Maßangaben

Im Fließtext werden natürliche Zahlen zwischen 0 und 9 üblicherweise durch Worte dargestellt, alle anderen Zahlen durch Ziffern (z.B. fünf Schritte, 42 Schritte). Sätze oder Überschriften sollten nicht mit einer Ziffer beginnen. Versuchen Sie, den Satz umzuformulieren, falls dies nicht möglich ist, schreiben Sie die Zahl aus.

Wenn Sie in einem Satz Zahlen kleiner als 10 mit Zahlen größer als 10 vergleichen, verwenden Sie durchgängig Ziffern (z.B. „in 4 von 20 Fällen“). Die Zahlen „Null“ und „Eins“ sind in Worten dargestellt verständlicher. Sie können deshalb auch in solchen Fällen ausgeschrieben werden (z.B. „eine von 50 Testpersonen“).

Bei Angabe des Datums sollte der Tag immer in Ziffern angegeben werden. Zwischen den Datumsbestandteilen wird ein Punkt gesetzt, führende Nullen sind nicht üblich. Schreiben Sie, wenn möglich, die Monatsangabe aus. Bei Zeitangaben werden Stunden und Minuten durch einen Punkt (11.30) oder eine Hochstellung voneinander abgehoben (11<sup>30</sup>).

Ganze Zahlen aus mehr als 3 Ziffern werden von der Endziffer aus in dreistellige Gruppen zerlegt. Gliedern Sie die Gruppen durch Zwischenräume. Eine Gliederung durch Punkte, Kommata oder Hochkommata sollte vermieden werden, weil

es dadurch zu Verwechslungen zwischen deutsch- und englischsprachigen Texten kommen kann. In  $\text{\LaTeX}$  verwenden Sie für den Zwischenraum ein Spatium ( $\backslash,$ ), in Word ein geschütztes Leerzeichen (Strg+Leerzeichen).

Nummern, die Sie verwenden, um Kundennummern, IDs oder ähnliches anzuzeigen, werden immer in Ziffern geschrieben. Sie können dabei nach Belieben gegliedert werden. Dabei sollten Sie allerdings etablierte Vorgehensweisen berücksichtigen:

- Telefon- und Faxnummern werden ausgehend von der letzten Ziffer in Zweiergruppen gegliedert. Trennen Sie die Ortsvorwahl durch einen Schrägstrich oder durch Klammerung ab, innerhalb von Vorwahlen wird nicht gegliedert (z.B. 0800/3 30 33 33). Sie können auch das internationale Telefonnummernformat verwenden. Vor die in Klammern gesetzte Ortsvorwahl wird dabei die durch ein Pluszeichen eingeleitete Landesvorwahl gestellt. Führende Nullen der Ortsvorwahl entfallen normalerweise (z.B. +49 (800) 3 30 33 33).
- IPv4-Adressen werden ausgehend von der höchstwertigen Stelle in vier Bytewerte zerlegt. Diese Bytewerte werden dezimal niedergeschrieben und durch Punkte getrennt (z.B. 127.0.0.1) ([2] S. 13).
- IPv6-Adressen werden ausgehend von der höchstwertigen Stelle in acht 16-Bit-Werte zerlegt. Diese werden hexadezimal codiert und durch Doppelpunkte getrennt dargestellt. Führende Nullen können weggelassen werden (z.B. 1080:0:0:0:8:800:200C:417A). Zusätzlich kann dabei eine einzige längere Gruppe von Nullen durch einen doppelten Doppelpunkt dargestellt werden (z.B. 1080::8:800:200C:417A) ([3] S. 6ff).
- Postleitzahlen werden nicht gegliedert.

Dezimalstellen werden im Deutschen laut Duden von den ganzen Zahlen durch Komma getrennt. Wenn Sie Ihre Arbeit auf Englisch schreiben, verwenden Sie statt dessen einen Punkt. Bei Zahlen kleiner als Eins wird auch die führende Null angegeben.

Zwischen einem Zahlenwert und einer vorhergehenden oder nachfolgenden Einheit oder Abkürzung darf kein Zeilenumbruch erfolgen (z.B. 11 km, € 5 000). Verwenden Sie ein geschütztes Leerzeichen, um dies zu erzwingen.

## 2.9 Überprüfen auf Fehler

Sobald Sie Ihre Arbeit fertig gestellt haben, überprüfen Sie Ihr eigenes Werk auf Fehler:

- Führen Sie eine Rechtschreibprüfung durch; nutzen Sie auch die Möglichkeiten dies automatisiert zu tun.
- Lesen Sie Ihre Arbeit selbst noch einmal durch und merzen Sie die „letzten Fehler“ aus.
- Wenn möglich, bitten Sie einen Kollegen, die Arbeit durchzulesen und dabei auf inhaltliche Mängel (die Sie selbst aus „Betriebsblindheit“ übersehen) zu achten.

Wenn Sie der Meinung sind, alles für eine gute Arbeit getan zu haben, geben Sie die „vorläufige Endfassung“ bei Ihrem Betreuer ab. Er wird die Arbeit gegebenenfalls durchsehen (nicht „korrigieren“) und Verbesserungsvorschläge anbringen. In welchem Ausmaß Sie bei der letzten Überarbeitung auf diese Vorschläge eingehen, bleibt Ihnen überlassen.

Beachten Sie, dass auch Ihr Betreuer für die Durchsicht der Arbeit einige Zeit benötigt. Beachten Sie, dass die Verantwortung für Fehler und Unzulänglichkeiten aller Art letztlich bei Ihnen als Autor der Arbeit liegt und nicht beim Betreuer oder gar dem Gutachter.

### 3 Gestaltung

Die äußere Form der Abschlussarbeit vermittelt einen ersten Eindruck von ihrer Qualität. Ein ansprechend gestaltetes Werk ist angenehmer zu lesen als ein lieblos formatiertes.

Es kommt nicht auf eine möglichst phantasievolle und individuelle Gestaltung Ihres Schriftstücks an, sondern darauf, dass die Arbeiten ein einheitliches Erscheinungsbild aufweisen. Gewünscht ist ein klares Schriftbild.

Die Verwendung dieser Vorlage ermöglicht ihnen, sich auf den Inhalt ihrer Arbeit zu konzentrieren, da der überwiegende Anteil ohne explizite Formatierung ihrerseits auskommen sollte.

#### 3.1 Dateien

Im Folgenden finden Sie eine Liste der Dateien, die Sie im Laufe ihrer Arbeit möglicherweise bearbeiten müssen. Hier nicht genannte Dateien sollten nicht bearbeitet werden, damit ihre Arbeit dem vorgegebenen Format entspricht.

Dateiname	Inhalt
variablen.tex	Enthält einige Textbausteine, die Sie entsprechend ihrer Arbeit anpassen sollten. Dabei ersetzen Sie den Text im jeweils zweiten geschweiften Klammersatz durch einen für sie zutreffenden Wert.
preamble.tex	preamble.tex enthält die nicht zu nummerierenden Kapitel Zusammenfassung, Abstract und Danksagung. Anders als im normalen Inhalt werden diese drei Kapitel nicht durchnummeriert, weshalb sie nicht mit <code>\chapter</code> ausgezeichnet sind. <code>\addchap</code> sollte im eigentlichen Text nicht verwendet werden. Beachten Sie, dass die Reihenfolge der Kapitel Abstract und Danksagung bei einer englischsprachigen Arbeit zu vertauschen ist.
studentcontent.tex	In dieser Datei sollten Sie den Fließtextanteil ihrer Arbeit platzieren.
bibliography.bib	Diese Datei soll ihre die Literaturlistenbank enthalten. Näheres dazu unter 4.3.
verzeichnisse.tex	Hier sind die Verzeichnisse gelistet, die ihre Arbeit verwenden soll. Wenn Sie neben Bildern und Tabellen andere Verzeichnisse wünschen, so müssen Sie diese hier aus den Kommentaren entfernen.

Tabelle 2: Die für ihre Arbeit relevanten Dateien

Stellen sie sicher, dass sie die Vorlage verwenden, die ihrem Betriebssystem angepasst ist. Dies erlaubt ihnen die Verwendung des beigefügten Makefiles.

### 3.2 Verwendung des Makefiles

Der Vorlage ist ein Makefile angefügt, der sämtliche notwendigen Arbeiten zum Erstellen eines PDFs erfüllen sollte. Anbei finden sie eine Liste der Maketargets, die definiert sind.

Target	Ergebnis
clean	Dieses Ziel löscht Dateien, die im Laufe der Ausführung von $\text{\LaTeX}$ erzeugt werden.
distclean	siehe <i>clean</i>
really-clean	Zusätzlich zu den Dateien, die mit <i>clean</i> entfernt werden, werden auch alle DVI, PS und PDF-Dateien entfernt.
pdf	Dieses Ziel erzeugt das normale PDF ohne Netzwerkoptimierung.
optimized	Dieses Ziel wird ausgeführt, wenn kein Ziel explizit vorgegeben wurde. Es resultiert in einem normalen PDF und einem PDF, dass für die Lesbarkeit über Netzwerkverbindungen optimiert wurde.

Tabelle 3: Maketargets

Unter Linux empfiehlt es sich, eine Konsole zu öffnen, zum Ordner der Vorlage zu wechseln und *make* aufzurufen. Unter Windows wird die Vorlage mit der *NMAKE.EXE* ausgeliefert, welche ähnliche Funktionen wie *make* bereit stellt. Der oben genannte Ablauf ist unter Windows ebenfalls möglich mit den entsprechenden Ersetzungen. Zusätzlich dazu ist es möglich, die *NMAKE.EXE* direkt per Mausklick auszuführen, was das Standard Maketarget ebenfalls auslöst.

```
tdr@iem:~/bmclatex$ make
rm -f *.toc *.log *.idx *.ind *.aux *.loa *.out *~ *.dvi */*.bak *.bbl \
    *.blg *.glo *.nlo *.ilg *.lof *.lot *.gls *.nls *.brf
pdflatex -interaction batchmode bmclatex.tex
This is pdfTeX, Version 3.1415926-1.40.10 (TeX Live 2009/Debian)
entering extended mode
bibtex -terse bmclatex
A level-1 auxiliary file: titelseite.aux
A level-1 auxiliary file: eidesstattlicheerklaerung.aux
A level-1 auxiliary file: preamble.aux
A level-1 auxiliary file: verzeichnisse.aux
A level-1 auxiliary file: studentcontent.aux
-- IEEEtran.bst version 1.12 (2007/01/11) by Michael Shell.
-- http://www.michaelshell.org/tex/ieeetran/bibtex/
-- See the "IEEEtran_bst_HOWTO.pdf" manual for usage information.

Done.
pdflatex -interaction batchmode bmclatex.tex
This is pdfTeX, Version 3.1415926-1.40.10 (TeX Live 2009/Debian)
entering extended mode
pdflatex -interaction batchmode bmclatex.tex
This is pdfTeX, Version 3.1415926-1.40.10 (TeX Live 2009/Debian)
entering extended mode
pdfopt bmclatex.pdf bmclatex-optimized.pdf
```

#### Quellcode 1: Beispiel für fehlerfreie Ausführung des Makefiles unter Linux

```
tdr@iem:~/bmclatex$ make
pdflatex -interaction batchmode bmclatex.tex
This is pdfTeX, Version 3.1415926-1.40.10 (TeX Live 2009/Debian)
entering extended mode
make[1]: *** [pdf] Fehler 1
```

#### Quellcode 2: Beispiel für einen Fehler unter Linux

Sollte es wie in Code 2 gezeigt zu einem Fehler kommen, so können sie in der Datei *bmclatex.log* nähere Informationen zum Fehler finden.



```
tdr@iem:~/bmclatex$ make clean
rm -f *.toc *.log *.idx *.ind *.aux *.loa *.out *~ *.dvi */*.bak *.bbl \
  *.blg *.glo *.nlo *.ilg *.lof *.lot *.gls *.nls *.brf
```

Quellcode 3: Ausführen des Targets *clean* unter Linux

Wenn sie ein spezifisches Target ausführen möchten, so können sie dies über *make <target>* wie in Code gezeigt tun. Alle gezeigten Beispiele gelten entsprechend unter Windows für *NMAKE.EXE*.

## 3.3 Verwendung von L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X erlaubt es, sich beim Erstellen der Arbeit auf den Inhalt zu konzentrieren, während Sie explizite Formatierungen (fett, kursiv etc.) im Text selten benötigen werden. Ein neuer Absatz im Dokument wird entweder durch zweifachen Zeilenumbruch oder durch `\\` begonnen.

Im Folgenden wird eine kurze Einführung in die Befehle und Umgebungen gegeben, die Sie im Laufe ihrer Arbeit möglicherweise verwenden werden.

### 3.3.1 Dokumentenstruktur

Die Strukturierung innerhalb dieses Dokumentes erfolgt mit dem folgenden Befehlen (sortiert von höchstem Rang in der Hierarchie zum niedrigsten Rang):

```
\chapter{Überschrift}
\section{Überschrift}
\subsection{Überschrift}
\subsubsection{Überschrift}
\paragraph{Überschrift}
\subparagraph{Überschrift}
```

Quellcode 4: L<sup>A</sup>T<sub>E</sub>Xbefehle zur Strukturierung von Dokumenten

Nur die ersten drei Ebenen werden im Inhaltsverzeichnis geführt, durchnummeriert und bei der Referenzierung über Labels (siehe Teil 3.3.3) verwendet.

### 3.3.2 Aufzählungen

Für Aufzählungen wird die `itemize`-Umgebung, für Nummerierungen die `enumerate`-Umgebung verwendet.

```
\begin{itemize}
  \item Ein Punkt
  \item Ein weiterer Punkt
  \begin{itemize}
    \item Unterpunkt
    \item Unterpunkt
  \end{itemize}
\end{itemize}
```

#### Quellcode 5: Aufzählung mit Stichpunkten

```
\begin{enumerate}
  \item Ein nummerierter Punkt
  \item Ein weiterer nummerierter Punkt
  \begin{enumerate}
    \item Nummerierter Unterpunkt
    \item Nummerierter Unterpunkt
  \end{enumerate}
\end{enumerate}
```

#### Quellcode 6: Eine durchnummerierte Aufzählung

### 3.3.3 Labels

Wenn Sie in einem Teil ihres Textes auf einen anderen Teil verweisen möchten, so können Sie dies mithilfe von Labels tun. Labels platzieren Sie im Fließtext oder innerhalb eines Floats (siehe Teil 3.3.4) mit `\label{schlüsselwort}`. Ein gesetztes Label kann im gesamten Text referenziert werden mit `\ref{schlüsselwort}` oder `\pageref{schlüsselwort}`. Bei Verwendung von `\ref{}` wird die Sektion ausgegeben, in der das Label sich befindet oder, wenn ein Float referenziert wird, die fortlaufende Nummer.

### 3.3.4 Floats

Floats werden in  $\text{\LaTeX}$  für Inhalte verwendet, bei deren Platzierung etwas Spielraum vorhanden ist, um im Ergebnis ein angenehmeres Gesamtbild der Arbeit zu ermöglichen. Wenn Sie die Position genauer kontrollieren wollen, so können dem Beginn der Floatumgebung einen Parameter mitgeben, der diese genauer spezifiziert (Beispiel: `\begin{bmceq}[h]`).

Parameter	Platzierung des Floats
t	Seite oben
b	Seite unten
p	Seite mit Floats gefüllt
h	Hier, wenn möglich
H	Hier, definitiv

Tabelle 4: Positionierungsparameter für Floats

Floats werden automatisch in entsprechenden Verzeichnissen geführt wie beispielsweise dem Tabellenverzeichnis. Desweiteren können Floats mit Unterschriften versehen werden, wovon Sie in ihrer Arbeit nahezu immer Gebrauch machen sollten. Diese sollten den Inhalt kurz und prägnant benennen und, wenn nötig, eine Legende zur Interpretation des präsentierten Materials enthalten. Sie sollten, wenn Sie Material so einbinden, im Text auch darauf Bezug nehmen. Dies wird erleichtert durch die Verwendung von Labels, die innerhalb eines Floats platziert werden können, um diese im Text zu referenzieren (Beispiel siehe Code 8). Das Label muss nach `\caption` platziert werden, sonst wird nicht die fortlaufende Nummer, sondern die Nummerierung des Abschnitts ausgegeben.

### Formeln

Mathematische Formeln können entweder im Fließtext (`$ \rho = \lambda / \mu $` wird zu  $\rho = \lambda / \mu$ ) oder in einer separaten Zeile platziert werden. In dieser Arbeit sollten Sie für die Platzierung in einer neuen Zeile die `equation*`-Umgebung gekapselt in einer `bmceq`-Umgebung nutzen.

```
\begin{bmceq}
\begin{equation*}
a^2 + b^2 = c^2
\end{equation*}
\caption{Der Satz des Pythagoras}\label{eqn:pythagoras}
\end{bmceq}
```

Quellcode 8: Eine hervorgehobene Gleichung

```
$ a^2 + b^2 = c^2 $
```

Quellcode 7: Formel im Fließtext

## Programmcode

Programmcode (unter zwei Seiten Länge) kann im Fließtext eingebunden werden mit der `verbatim`-Umgebung, gekapselt in der `code`-Umgebung.

```
\begin{code}
\begin{verbatim}
public static void main(String[] args) {
    System.out.print("Hello World");
}
\end{verbatim}
\caption{Ein einfaches Hello World Programm}\label{lst:helloworld}
\end{code}
```

Quellcode 9: Ein Beispiel zum Einbinden von Quellcode

## Abbildungen

Gestalten Sie ihre Bilder schlicht und verzichten Sie auf Schnörkel, die den Leser vom Wesentlichen ablenken könnten. Sie sollten Grafiken verwenden, wenn Sie eine Information illustrieren oder Daten abbilden wollen. Dekorative Grafiken haben in wissenschaftlichen Arbeiten keinen Platz. Nummerieren Sie Ihre Bilder fortlaufend und versehen Sie sie mit Unterschriften und Legenden. Verweisen Sie im Text auf die Bilder.

```
\begin{figure}
\centering
\includegraphics[scale=1.0]{icblogo.jpeg}
\caption{Das ICB-Logo}\label{fig:icblogo}
\end{figure}
```

Quellcode 10: Einbindung einer Grafik

Beachten Sie, dass Graustufen und Farben in Bildern bei der Vervielfältigung unzureichend wiedergegeben werden. Oft gehen feine Schattierungen verloren. Verwenden Sie andere Formen der Hervorhebung in Bildern, etwa Umrahmung oder stärkere Linien. Achten Sie bei Bildschirmfotos auf derartige Schattierungen. Im Zweifelsfall probieren Sie einfach aus, ob Details eines Bildes beim Kopieren verloren gehen.

Bei technischen Zeichnungen und Diagrammen achten Sie darauf, dass Sie eine Legende angeben, die alle Zeichnungselemente erklärt.

```

\begin{figure}
\centering
\includegraphics[scale=1.0]{informatikranking.jpeg}
\textit{Legende: grün, gelb und rot bezeichnen jeweils sehr gute,
gute oder mäßige Bewertungen bezüglich der jeweiligen
Kriterien. (Quelle: \url{www.das-ranking.de}, Stand: 2008)}
\caption{Ranking von Informatik-Standorten; Kriterium: Studiensituation insgesamt}
\end{figure}

```

Quellcode 11: Einbindung einer Grafik mit Legende

## Tabellen

Die Formatierung der Tabellen ist freigestellt. Sie sollte allerdings innerhalb des Dokuments einheitlich sein und am besten ebenfalls über Formatvorlagen geregelt werden. Auch Tabellen sollten beschriftet sein. Tabellen im Fließtext erstellen Sie in einem `table`-Float. Im Regelfall sollten Sie die `tabularx`-Umgebung benutzen, die die gesamte Breite der Seite ausnutzt.

```

\begin{table}
\centering
\begin{tabularx}{\textwidth}{|l|X|X|}
\hline
Spaltenkopf 1 & Spaltenkopf 2 & Spaltenkopf 3\\
\hline
Zeilenkopf 1 & Daten & Daten\\
\hline
Zeilenkopf 2 & Daten & Daten\\
\hline
Zeilenkopf 3 & Daten & Daten\\
\hline
\end{tabularx}
\caption{Eine Beispieltabelle}\label{tab:beispiel}
\end{table}

```

Quellcode 12: Eine Tabelle in  $\LaTeX$ 

Dieses Beispiel ergibt eine dreispaltige Tabelle mit Trennlinien, deren Text überall linksbündig geschrieben ist. Die mittlere und rechte Spalte werden gedehnt, um die Tabelle auf Seitenbreite zu erweitern.

Inbesondere bei Tabellen kann es geschehen, dass sie sich über mehrere Seiten erstrecken, was mit Floats allerdings nicht möglich ist. Wenn Sie dies wünschen, können Sie eine `longtable` verwenden.

```

\begin{longtable}{|l|l|l|}
\hline
Spaltenkopf 1 & Spaltenkopf 2 & Spaltenkopf 3\\
\hline
Zeilenkopf 1 & Daten & Daten\\
\hline
Zeilenkopf 2 & Daten & Daten\\
\hline
Zeilenkopf 3 & Daten & Daten\\
\hline
\caption{Eine Beispieltabelle}\label{tab:beispiel}
\end{longtable}

```

Quellcode 13: Eine longtable in  $\text{\LaTeX}$

### 3.4 Allgemeines Schriftbild

Beachten Sie für Ihre Arbeit folgende grundlegende Tipps für die Gestaltung des Druckbilds:

- **Aufeinanderfolgende Überschriften** sollten vermieden werden. Beginnen Sie jedes Kapitel und jeden Abschnitt mit einer kurzen Einleitung oder einer Überleitung vom vorangegangenen Text.
- Variablen und andere **Programmfragmente** sollten im Fließtext durch eine andere Schriftart angedeutet werden. Wir empfehlen dafür die `verbatim`-Inlineumgebung.

```
\verb+myVariable+
```

- Für **Hervorhebungen** wird `\emph{Text}` (Beispiel: *Hervorgehobener Text*) empfohlen.
- **Fettdruck** wirkt aufdringlich und sollte daher nur dort benutzt werden, wo diese Wirkung erwünscht ist, zum Beispiel bei der Definition neuer Begriffe.
- **Versalien** (GROSSBUCHSTABEN) sollten zur Hervorhebung vermieden werden, weil sie ein unruhiges Schriftbild bewirken.
- **Fußnoten** sind in anderen Disziplinen (vor allem in den Rechts- und Wirtschaftswissenschaften) durchaus üblich<sup>1</sup>, in der Informatik sollte man damit sparsamer und mit Augenmaß umgehen. Wichtige Erläuterungen gehören in den Fließtext.

### 3.5 Präsentation der Arbeit

Im Rahmen Ihrer Abschlussarbeit sollen Sie Ihre Arbeit auch vor einem Publikum durch einen Vortrag präsentieren. Ziel einer solchen Präsentation ist

<sup>1</sup>In den Rechts- und Wirtschaftswissenschaften werden Fußnoten vor allem für Literaturverweise verwendet. Folgen Sie stattdessen den Empfehlungen aus Abschnitt 4.3 zum Verweisen auf Literatur.

es, den Inhalt der Arbeit dem Publikum darzulegen. Ob eine (Abschluss-)Präsentation Ihrer Arbeit in die Bewertung der Arbeit einfließt oder sogar separat mit Credits und einer Note versehen wird, wird von der jeweils gültigen Prüfungsordnung geregelt. Die Durchführung einer Abschlusspräsentation gehört zum guten akademischen Stil, auch wenn sie nicht als Prüfungsleistung vorgesehen ist. In künftigen Prüfungsordnungen (ca. ab dem Jahr 2011) wird die Abschlusspräsentation verbindlich hinzukommen. Der Zweck einer Präsentation während(!) des Ablaufs der Arbeit ist vielmehr, Ihnen für den weiteren Verlauf der Arbeit Hilfestellung zu geben.

- Die Zusammenfassung in einer relativ kurzen Präsentation hilft Ihnen, sich (wieder) des Kerns Ihrer Arbeit bewusst zu werden.
- Fragen und Anregungen aus dem Publikum helfen Ihnen, interessante Teile der Arbeit zu finden, die in der Abschlussarbeit besonders erwähnt werden sollen, oder verwandte Ansätze zu finden, die Sie in Ihrer Arbeit mit aufnehmen können.
- Auch im späteren Berufsleben müssen Sie häufig Ihre Arbeit kompakt zusammenfassen. Beim Abschlussvortrag steht Ihnen eine Umgebung zur Verfügung, in der Sie dies „gefahrlos“ üben können.

Eine Präsentation sollte deshalb aus folgenden zwei Phasen bestehen:

- Präsentation des neuen Materials
- Wiederholung, wie etwa Kerngedanken am Anfang und am Ende eines Unterpunktes nennen, ein Beispiel einbauen. Assoziation mit Bekanntem: eine Brücke zu Alltagssituationen schlagen oder Vergleichbares in „Informatik-Basiswissen“ finden

Ein Rahmen für den Aufbau eines Fachvortrags kann z.B. so aussehen:

- Motivation - Problem - Lösungsidee - Methode - Wiederholung oder Beispiel.
- Die Gliederung des Vortrags soll bereits während der Stoffsammlung geschehen.
- Inhalt sorgfältig auswählen! Für Vorträge gilt: Weniger ist mehr.
- Bilder können ein wichtiger Bestandteil eines Vortrages sein, jedoch sollten Sie sorgfältig vorbereitet, sauber ausgeführt und sachgerecht eingesetzt werden. Die wesentliche Bildinformation sollte in etwa zehn Sekunden zu erfassen sein. Überfrachtete Bilder lenken vom Vortrag ab.

Hier sollen noch einige Tips zur Präsentation des Vortrags selbst gegeben werden:

- Üben Sie den Vortrag vorab laut. Es hat sich bewährt, sich den eigenen Vortrag auf Audio oder Video aufzunehmen, oder einem Freund/Freundin vorzutragen. Dies gibt Sicherheit, hilft bei der Zeiteinteilung und bringt noch zu wenig vorbereitete Stellen im Vortrag ins Bewusstsein.
- Faustregel: Zu jeder wesentlichen Aussage gehört (mindestens) eine eigene Folie, zu jeder Folie eine Aussage.

- Nur Kernaussagen, nicht die fortlaufenden Texte auf den Folien präsentieren.
- Für Folien große und lesbare Schrift verwenden.
- Die Folien fortlaufend nummerieren, damit man sich in der nachfolgenden Diskussion oder bei Zwischenfragen darauf beziehen kann.
- Faustregel: pro Seite zwei bis vier Minuten reden (20 Minuten → 7 Seiten / Folien), wobei die Verweilzeit natürlich vom Inhalt der Folie abhängt und somit die Abweichungen von dieser Regel beträchtlich sein können.
- Im letzten Drittel des Vortrags eine Abkürzung vorsehen, die eingeschlagen wird, wenn nach 50 bis 70% der Zeit weniger als die entsprechende Anzahl von Folien gezeigt ist. Temposteigerung ist meist kein Ausweg, um Zeit einzusparen!
- Sollten Sie in der anschließenden Diskussion bereits bestimmte Fragen erwarten, können Sie dafür zusätzliche Folien vorbereiten, die nach dem Ende des Vortrags einsortiert werden. Vor allem, wenn Sie sich nach dem Übungsvortrag entschließen, Folien zur Zeiteinsparung wegzulassen, ist so auch Ihre Vorbereitung möglicherweise doch nützlich.
- Beim Vortragen aufrechte, entspannte Haltung einnehmen, weder gehemmt noch aggressiv.
- Nicht ständig auf- und abgehen, sondern nur kontrollierte Ortsveränderungen vornehmen.
- Die Hände für unterstützende Gesten frei halten. Ein Zeigestock oder Laserpointer kann sinnvoll/notwendig sein.
- Als Gesten nicht zu kleine, fahrige Bewegungen verwenden, aber auch nicht zu weit ausholen. Es gibt schließlich Kulturkreise, in denen große Gesten als Bedrohung empfunden werden.
- Laut, deutlich und langsam sprechen, Sprechpausen einlegen, sich um Lebendigkeit der Sprache bemühen.
- Kontakt zum Publikum aufbauen, in das Publikum schauen, nicht in Richtung der Vortragsmedien (Projektionsfläche, Tafel, Notebook) sprechen.
- Nicht einzelne Personen im Publikum fixieren, sondern Blick wandern lassen (bekannte oder freundliche Gesichter suchen) und in den Gesichtern lesen.



## 4 Arbeiten mit Literatur

Wissenschaftliche Literatur ist die wichtigste Informationsquelle für Ihre Arbeit. Hier eine kurze Charakterisierung der Literatur:

- **Bücher** befassen sich meist mit einem größeren Problembereich, der systematisch und vollständig behandelt wird. Der Nachteil von Büchern ist, dass sie meist nicht die allerneuesten Erkenntnisse behandeln. Dafür enthalten Bücher meist eine ganzheitliche Präsentation des Stoffgebietes. Der Inhalt wird von Experten vor dem Erscheinen geprüft.
- **Artikel in Fachzeitschriften und Artikelsammlungen** sind wissenschaftlich höchst wertvoll und präsentieren oft neue Resultate für ein Fachpublikum. Die Artikel werden von Experten vor dem Erscheinen geprüft und oft dem Autor zur Überarbeitung zurückgegeben. Dieser Review-Vorgang kann sich über mehrere Monate ziehen und sorgt meist für sehr hohe Qualität (allerdings zu Lasten der Aktualität).
- **Artikel in Konferenzbänden** haben ähnliche Eigenschaften wie Artikel aus Zeitschriften, jedoch ist die Qualitätskontrolle nicht ganz so perfekt. Der Grund dafür ist, dass der Artikel zu einem bestimmten Datum fertig sein muss, um rechtzeitig zur Konferenz zu erscheinen. Die Qualität zwischen einzelnen Konferenzen variiert stark.
- **Technical Reports** enthalten meist neueste Inhalte, manchmal Vorabversionen für oben genannte Artikel. Technical Reports durchlaufen selten eine externe Qualitätskontrolle, trotzdem kann ihre Qualität und vor allem ihre Ausführlichkeit besser sein als die von Artikeln in Zeitschriften oder Konferenzbänden, wo die Autoren ein Seitenlimit beachten müssen.

In der Regel nennt man nur Artikel „Publikationen“, die eine Qualitätskontrolle bestanden haben. „Papers“ ist ein Überbegriff, der alle oben aufgeführten Publikationsarten umfasst.

### 4.1 Literatursuche

Die einzelnen Schritte bei der Suche nach Literatur (nach [4]):

1. Fragen Sie Personen, die im entsprechenden Gebiet arbeiten. Ebenso finden Sie in den Bibliotheken des Fachbereichs fachspezifische Bücher.
2. Betrachten Sie die Literaturliste in den vorhandenen relevanten Publikationen.
3. Betrachten Sie das Inhaltsverzeichnis und den Index vorhandener relevanter Bücher, um zusätzliche Schlüsselworte für die weitere Suche zu finden.
4. Suchen Sie im Schlüsselwortkatalog von Bibliotheken, z.B. hier <http://www.ub.uni-duisburg-essen.de/>
5. Durchsuchen Sie „Review Journals“, wie etwa Computer Abstracts, ACM Computing Reviews, ACM Guide to Computing Literature.
6. Beachten Sie vor allem die digitale Bibliothek von acm unter [www.acm.org](http://www.acm.org), die aus der Domäne der Universität Duisburg-Essen komplett zugreifbar

ist. Auch die digitale Bibliothek von IEEE ist, zumindest teilweise, kostenfrei verfügbar.

Viele Artikel sind im Internet frei zugänglich. Eine spezielle Suchmaschine dafür ist CiteSeer, auf die unter <http://citeseer.nj.nec.com/cs> zugegriffen werden kann. Diese Suchmaschine gibt auch an, wie oft ein Artikel von anderen zitiert worden ist („Citations“), was eine wichtige Maßzahl für die Qualität eines Artikels ist. Bei der Suche im Internet sollten Sie bei speziellen Zeichen im Autorennamen auch nach Varianten davon suchen: Müller, Mueller, Muller. Manchmal wird das ß übrigens falsch als B erfasst („Guido RöBling“).

## 4.2 Literaturbeschaffung

- **Bücher:** Bibliotheken. Bei Konferenzbänden oder Artikelsammlungen nach den Namen der Editoren der Ausgabe suchen, nicht nach den Namen der einzelnen Artikelverfasser.
- **Fachzeitschriften:** Zentral- und Fachbereichsbibliotheken
- **Fernleihe:** Für nicht vorhandene Bücher oder einzelne Artikel nicht vorhandener Zeitschriften oder Bände.
- **WWW:** Seiten der Autoren oder Suchmaschinen für Literatur, wie beispielsweise:
  - The Collection of Computer Science Bibliographies <http://liinwww.ira.uka.de/bibliography/index.html>
  - Libraries <http://wwwbs.cs.tu-berlin.de/bibliotheken/index.html>
  - Literatursuche im Internet <http://www.wi-inf.uni-essen.de/suche/literatur.html>

## 4.3 Literaturverweise

Diese Vorlage verwendet BibTex zur Erzeugung des Literaturverzeichnisses, wobei die Einträge nach den Vorgaben der IEEE formatiert sind.

### 4.3.1 Erstellen einer Literaturdatenbank

Bibtex generiert die Einträge des Literaturverzeichnisses aus einer Datenbank. Diese Datenbank wird in dieser Vorlage in *bibliography.bib* geführt. Es handelt sich hierbei um eine Textdatei, die sie entweder mit einem gängigen Texteditor oder mit einem dedizierten Literaturverwaltungsprogramm wie JabRef (<http://jabref.sourceforge.net/>) bearbeiten können.

Neben umfangreichen Daten wie Art der Literatur (Artikel, Buch, Konferenzbeitrag, ...) und Daten über die Literatur (Titel, Autorname, Monat/Jahr der Publikation, ...) muss jeder Eintrag mit einem einzigartigen Key versehen werden, damit er in  $\text{\LaTeX}$  referenziert werden kann.

Eine Methode zur Wahl des Keys und zur Benennung von Dateien in der verwendeten Literatur ist die Verwendung des ersten Buchstaben der Nachnamen der Autoren, gefolgt vom Jahr der Veröffentlichung. Bei mehr als drei Autoren werden nur die Nachnamen der ersten drei Autoren verwendet, gefolgt von

einem ”+“. Bei Schlüsselkollisionen wird ein ”-“ an die Jahreszahl angehängen, gefolgt von einer Zahl, die entsprechend bei folgenden Kollisionen inkrementiert wird (Beispiel: BD11, BD11-2, BD11-3, ...).

Beispiel:

Becke, Martin ; Dreibholz, Thomas : Multi-Path Transport. (2011) →BD11  
Dreibholz, Thomas ; Adhari, Hakim ; Becke, Martin ; Rathgeb, Erwin P. : Simulation and Experimental Evaluation of Multipath Congestion Control Strategies. (2012) →DAB+12

Dieses Schema ist nur eine Möglichkeit, die allerdings nicht zwingend notwendig ist und von der Sie auch jederzeit abweichen können, soweit Sie es für sinnvoll erachten.

Bei mehreren Autoren müssen die Namen der Autoren im Eintrag mit ”and“ voneinander getrennt werden.

Beispiel:

Thomas Dreibholz and Hakim Adhari and Martin Becke and Erwin P. Rathgeb

Bei Titeln kann es vorkommen, dass innerhalb des Titels Buchstaben unabhängig von üblicher Gross- und Kleinschreibung explizit gross geschrieben werden müssen. Diese Buchstaben müssen in der Datenbank mit geschweiften Klammern umgeben werden.

Beispiel:

{T}hinking, {S}peaking, {W}riting. {B}asic {W}orking {T}echniques for {S}tudents of {M}athematics and {C}omputer {S}cience.

Der **Umfang des Literaturverzeichnisses** und somit auch ihrer Literaturdatenbank muss einer akademischen Arbeit angemessen sein. Als Verfasser einer Abschlussarbeit müssen Sie nicht nur aus Vorlesungsmanuskripten, Lehrbüchern und Programmierhandbüchern schöpfen, sondern auch aktuelle wissenschaftliche Arbeiten z.B. aus der digitalen Bibliothek von [www.acm.org](http://www.acm.org) zur Kenntnis nehmen und möglichst auch lesen. Diese Quellen gehören in das Literaturverzeichnis Ihrer Abschlussarbeit. Wenn Ihr Literaturverzeichnis lediglich auf Vorlesungsmanuskripte, Programmierhandbücher, Produktkataloge und Internetadressen verweist, so dokumentiert das eine oberflächliche Auseinandersetzung mit dem behandelten Thema.

#### 4.3.2 Zitieren aus der Literaturdatenbank

Während Literaturdatenbanken häufig das komplette gesichtete Material führen, das Sie in der Vorbereitung ihrer Arbeit gelesen haben, so wird in der Arbeit hinterher meist nur ein Bruchteil davon auch verwendet.

Quellenangaben werden in  $\text{\LaTeX}$  mit `\cite{key}` platziert. Dies sorgt sowohl für eine Quellenangabe an der aktuellen Position als auch für einen Eintrag der Quelle im Literaturverzeichnis. Bei zukünftiger Referenzierung des gleichen Schlüssels wird automatisch auf den entsprechenden Eintrag im Literaturverzeichnis verwiesen, was eine unerwünschte doppelte Benennung derselben Quelle verhindert.

## 5 Weitere Aspekte

### 5.1 Themenfindung

Es ist sinnvoll, sich möglichst frühzeitig (einige Monate) vor dem geplanten Beginn der Abschlussarbeit mit potentiellen Betreuern in Verbindung zu setzen.

In einem ersten Gespräch können die interessanten Themenbereiche diskutiert werden. Ergebnis dieses ersten Gesprächs sind typischerweise 1 - 3 mögliche Themengebiete für die Arbeit und eine Liste von Gebieten, in denen noch Vorkenntnisse fehlen.

In der folgenden „Konkretisierungsphase“ werden das bzw. die angedachten Themen schrittweise verfeinert und auf ihre Durchführbarkeit untersucht. Da in den meisten Fällen im Rahmen der Abschlussarbeit eine Software neu entwickelt bzw. eine vorhandene Software modifiziert wird, kann man sich an Vorgehensweisen aus der Softwareentwicklung orientieren. Ziel ist eine Entscheidung für eine der angedachten Alternativen und eine Zerlegung der Gesamtaufgabe in Teilaufgaben, für die sich die Aufwände halbwegs sicher schätzen lassen.

Während der Konkretisierungsphase müssen ferner ggf. fehlende Vorkenntnisse erworben werden; der Zeitaufwand hierfür sollte nicht unterschätzt werden.

### 5.2 Deutsch oder Englisch?

In der Regel lassen die Prüfungsordnungen eine Wahl zwischen Deutsch und Englisch als Sprache für die schriftliche Ausarbeitung zu. Wenn Deutsch Ihre Muttersprache ist, so haben Sie den Vorteil, dass Sie Ihren Arbeitsfortschritt und die Ergebnisse wesentlich differenzierter darstellen können als Ihnen das im Englischen möglich sein wird. Bei sehr technischen oder formalen Themen kann es dennoch günstiger sein, die Arbeit in englischer Sprache zu verfassen. Es kann auch eine wichtige Erfahrung sein, ein größeres Schriftstück in Englisch zu verfassen. Unterschätzen Sie aber bitte nicht den Mehraufwand, den auch Ihr Betreuer mittragen muss. Besprechen Sie diesen wichtigen Aspekt deswegen bereits in der Phase der Themenfindung.

### 5.3 Externe Arbeiten

Prinzipiell können Abschlussarbeiten auch bei externen Einrichtungen (Forschungsinstitute, andere Fachbereiche, öffentliche Einrichtungen, Industrie) durchgeführt werden. Hier ist von allen Seiten besondere Aufmerksamkeit geboten. Ein wichtiger Aspekt ist, dass die Aufgabenstellung zwar sinnvoll und nützlich für den externen Partner sein soll, aber auch den Ansprüchen an eine wissenschaftliche Abschlussarbeit genügen muss. Unbedingt notwendig ist ein „interner“ Betreuer an der Universität, der ein am betreffenden Studiengang beteiligter Hochschullehrer (Professor/in) ist und für die Begutachtung inkl. Benotung verantwortlich ist.

### 5.4 Bezahlung

Grundsätzlich erhalten Studierende für die Erstellung der Abschlussarbeit keine(!) Vergütung. Das Erstellen der Arbeit und bezahlte Tätigkeiten müssen deut-

lich getrennt werden. Manchmal zahlen Firmen eine Aufwandsentschädigung oder eine Praktikantenvergütung, welche die durch zeitweise Anwesenheit am Firmenstandort entstehenden Zusatzaufwände für Fahrt, Unterkunft und Verpflegung abdecken kann.

## **5.5 Rechte an den Ergebnissen**

Der Autor einer Abschlussarbeit hat Autorenrechte und nicht-ausschließliche Nutzungsrechte, d.h. prinzipiell hat auch die Universität (der Betreuer der Arbeit) ein kostenloses und unwiderrufliches Nutzungsrecht an den Ergebnissen der Arbeit zu Zwecken von Forschung und Lehre.

## **5.6 Projektberichte/Seminarausarbeitungen**

Diese Vorlage kann natürlich auch für andere Arbeiten benutzt werden. In der Regel werden Sie zum Beispiel für ein sog. „Seminar“ nicht nur eine Präsentation abhalten müssen, sondern es wird auch eine schriftliche Ausarbeitung des Themas verlangt, die sich in Form und Inhalt (aber natürlich nicht im Umfang) an den hier zusammengefassten Ansprüchen orientieren muss.

Bei einem (Haupt-)Seminar (3 Leistungspunkte, 90 Arbeitsstunden) wird meist eine Ausarbeitung im Umfang von 15-20 Seiten, die Erstellung einer Präsentation („Folien“) und ein Vortrag (20-30 Minuten) mit anschließender Diskussion erwartet.

Ein Projektseminar (9 Leistungspunkte, 270 Arbeitsstunden) ist bezüglich der Ausarbeitung natürlich umfangreicher, insbesondere werden meist Aufwände bezüglich Implementierung, Dokumentation, Test, Erprobung, Experimenten oder Messungen hinzukommen.

# Literaturverzeichnis

- [1] Plain english campaign. [Online]. Available: <http://www.plainenglish.co.uk/>
- [2] R. Braden. (1989, Oktober) Requirements for Internet Hosts – Communication Layers. Internet Standard 3. Network Working Group. [Online]. Available: <http://rfc.net/std3html>
- [3] R. Hinden and S. Deering. (1998, Juli) IP Version 6 Addressing Architecture. Request for Comments 2373. Network Working Group. [Online]. Available: <http://rfc.net/rfc2373.txt>
- [4] B. Buchberger, "Thinking, Speaking, Writing. Basic Working Techniques for Students of Mathematics and Computer Science." Begleitmaterial zu seiner Vorlesung Praktische Beweistechnik und wissenschaftliches Arbeiten im Bereich des Symbolic Computation", Universität Linz, 1992.
- [5] M. Deininger, H. Lichter, and L. Jochen, *Studien-Arbeiten - ein Leitfaden zur Vorbereitung, Durchführung und Betreuung von Studien-, Diplom- und Doktorarbeiten am Beispiel Informatik*, 3rd ed., Vdf Hochschulverlag AG an der ETH Zürich und B.G. Teubner, Stuttgart, 3. Auflage, Vdf Hochschulverlag AG an der ETH Zürich und B.G. Teubner, Stuttgart, 1996.
- [6] H. F. Ebel and C. Bliefert, *Vortragen in Naturwissenschaft, Technik und Medizin*, VCH Verlagsgesellschaft mbH, Weinheim, 1992.
- [7] S. P. Jones, J. Launchbury, and J. Hughes, "How to give a good research talk," in *ACM SIGPLAN Notices*, vol. 3, 1993.
- [8] Bereinigte Sammlung (Ordnungen, Studien- und Prüfungsordnungen). Universität Duisburg-Essen. [Online]. Available: [www.uni-duisburg-essen.de/zentralverwaltung/bereinigte\\_sammlung\\_ordnung.shtml](http://www.uni-duisburg-essen.de/zentralverwaltung/bereinigte_sammlung_ordnung.shtml)
- [9] (2007, Juni) Empfehlungen des Fakultätentags Informatik für Abschlussarbeiten in Informatikstudiengängen an Universitäten. [Online]. Available: <http://www.ft-informatik.de/->Dokumente>