

# Hinweise zur Durchführung von Abschlussarbeiten

(Generiert mittels pandoc per Markdown-Original)

Prof. Dr.-Ing. Christoph P. Neumann

## Inhalt

<b>WARNHINWEIS</b>	<b>1</b>
<b>Die wichtigsten Hinweise</b>	<b>1</b>
<b>Ausarbeitung</b>	<b>3</b>
IMRE . . . . .	3
Zielführung & Ergebnis . . . . .	3
LaTeX . . . . .	4
Zeichnungen/Grafiken . . . . .	5
<b>Orga</b>	<b>5</b>
Leistungen . . . . .	5
Erste Schritte . . . . .	8
Zwischenspiel . . . . .	8
Selbstkorrektur / Checkliste . . . . .	8
Vorkorrektur / Vorab-Review . . . . .	8
Abgabetag . . . . .	10
Verteidigung (Abschlussvortrag) . . . . .	10
FAQ . . . . .	11
Touch-Down Checkliste . . . . .	11
<b>Wissenschaftliche Suchmaschinen</b>	<b>13</b>
<b>Wissenschaftliches Schreiben</b>	<b>13</b>
Vermeidbare Fehler (aka “Kindergartenfehler”) . . . . .	13
Stilistisch . . . . .	16
Inhaltlich . . . . .	16
Typographie und Gestaltung . . . . .	16
Grobstruktur nach Prof. Neumann . . . . .	16
Allgemeine Ratschläge . . . . .	17
Spezielle Hinweise . . . . .	18

## WARNHINWEIS

Die Formatierungen sind ohne Gewähr. Dieses Dokument hatte ich zunächst in einem Wiki verfasst und später exportiert. Dabei ist viel Formatierung verloren gegangen. Dieses Dokument ist außerdem etwa im Jahr 2009 entstanden und daher können URLs und Teile des Inhalts mittlerweile veraltet sein.

## Die wichtigsten Hinweise

### Kompakte und gute Einführung

Kompakter Leitfaden von Deininger et al. zur Erstellung von Abschlussarbeiten am Beispiel Informatik:  
<https://amzn.to/37wKapC> (Werbung)

## Wissenschaftliche Vorgehensweise

Zur allgemeinen wissenschaftlichen Vorgehensweise lesen Sie bitte von Alan R. Hevner (2007) die Publikation “A three cycle view of design science research” aus dem Scandinavian Journal of Information Systems 19(2), 4.

<https://community.mis.temple.edu/seminars/files/2009/10/Hevner-SJIS.pdf>

## Begleitend dokumentieren

Nicht erst kurz vor Schluss! Vier Wochen reichen aller Erfahrung nach nicht, um eine ordentliche Ausarbeitung anzufertigen.

## Einleitung früh schreiben

Daran zeigt sich, ob man die Aufgabenstellung verstanden hat. Es macht auch klar, was man nicht machen muss.

## Plan machen und dokumentieren

Schritte und Termine; mit Betreuer besprechen.

## Literatur suchen und auswerten

- Was andere schon gemacht haben, sollte man nicht noch einmal machen, sondern benutzen.
- Wissenschaftliche Literatur! Nicht c't, heise, Wikipedia. Zeigen, dass man die einschlägigen Publikationsplattformen (Zeitschriften, Konferenzen) kennt.
- Ordentlich zitieren.

## Entwurfsentscheidungen begründen

Alternativen nennen und bewerten.

## Keine “How I did it”-Arbeit schreiben

Eine Schreibhaltung “How I did it?” führt i.d.R. zu geringem allg. Erkenntnisgewinn. Sie müssen das unbedingt vermeiden!

Das ist v.a. bei Abschlussarbeiten mit Fokus auf einen SW-Prototypen besonders relevant. Viele Anfänger werden hier dazu verführt. Genausowenig ist eine Abschlussarbeit ein Projektbericht oder ein Praktikumsbericht.

Ein plakatives Mantra um dem entgegenzutreten: Schreiben Sie eine “How to do it?”-Arbeit! Also einen “Leitfaden nach Hr. Mustermann”, der für alle vergleichbaren Problemstellungen Gültigkeitsanspruch haben sollte.

Die Grenzen dieser “vergleichbaren Problemstellung” dürfen je nach Level (Bachelorarbeit, Masterarbeit, Dissertation) unterschiedlich eng gesteckt sein. Insbesondere bei Bachelorarbeiten in Kooperation mit Unternehmen dürfen Sie die Grenzen sehr eng stecken! Dennoch ist die innere Haltungsfrage (etwa “How to do it?”) stets entscheidend.

Vergessen Sie auch zwischen Idee (Einleitung) und Umsetzung (Architektur/Implementierung) nicht Ihr Lösungskonzept! Auch der Leitfaden von Deininger et al. warnt Sie davor.

Obacht: Wenn wir Dozenten von Ihnen Erkenntnisgewinn fordern, dann hüten Sie sich vor **kontraproduktiver Ehrlichkeit** und **Selbstanklage**. Ich versuche das mal zu erklären: Das Mindestniveau des niedergeschriebenen Erkenntnisgewinns muss sich an der vorgesehenen Zielgruppe orientieren und soll an ihr gemessen nicht trivial sein. Bei Bachelorarbeiten ist die Mindestkompetenz der Zielgruppe durch Ihre “peer group” gegeben: eine Person mit der Lebenserfahrung von ca. 23 Lebensjahren und mit mind. sechs Semestern Informatik oder KI als Vorwissen; allerdings hat eine solche Person die vorgesehene Intensität (i.S.v. ECTS-Workload) in sein Studium eingebracht, d.h. sie hat dabei bspw. ihre Vorlesungen regelmäßig besucht und ihre Übungen lückenlos und selbstständig bearbeitet; es gibt daher eine statistische Diskrepanz zum durchschnittlichen Bearbeiter. Seien Sie sparsam damit in Ihrer Abschlussarbeit Erkenntnisgewinne zu formulieren, die eine solche Person bereits vorab haben sollte (Konjunktiv!). Beweisen Sie still die Fähigkeit zur Reflektion: Es geht beim geforderten Erkenntnisgewinn nicht um Ihre eigene ego-zentrische Perspektive (“How I did it?”), sondern um eine angemessen-allgemein-zentrische Perspektive (“How to do it?”). Beispiele: Schreiben Sie im Architekturkapitel die Architektur so auf, wie Sie zuletzt implementiert wurde und nicht etwa einen naiven nicht-tragfähigen Erstentwurf vor

Coding-Beginn (keine kontraproduktive Ehrlichkeit). Schreiben Sie im Testing-Abschnitt nicht, dass Ihre Unit-Tests für Sie selbst hilfreich geworden waren um Ihren eigenen Code zu verbessern oder dass Sie gemerkt haben, dass Sie eigentlich zu spät mit dem Testing begonnen haben (keine Selbstanklage).

### Regelmäßige Treffen mit dem Erstprüfer

- Mindestens einmal im Monat, am Anfang eher jede Woche
- Formelles Minimum (sonst abgabeverhindernd): Drei Treffen mit dem Erstprüfer – am Anfang, in der Mitte und kurz vor dem Ende

## Ausarbeitung

### IMRE

Das IMRE Konzept benennt vier Inhaltsblöcke (Grobklassifikation!), die durch eine wissenschaftliche Arbeit behandelt werden müssen. Es dient bei Publikationen der Grobgliederung (und oftmals sogar als direkter Aufbau) des Inhalts. Bei Bachelor- und Masterarbeiten sowie Survey-Publikationen und Dissertationen dient IMRE nicht einer direkten Gliederung, jedoch zur geschärften Wahrnehmung beim Aufbau der eigenen Arbeit und der inhaltlichen Trennung von Kapitelinhalten.

1. I: Idea
  - “Einleitung” => grobe Idee
  - [tlw. auch “IST-Analyse” => Grundlage der Idee/Problematik]
  - “Fachliche Anforderungsanalyse” => konkrete Idee, bzw. Problem
  - [tlw. auch “Fachkonzept” => Lösungsansatz]
2. M: Method(s) [oder auch: “Material & Methods”]
  - “Methodik” => Vorgehen
  - “Grundlagen” => grundsätzliche Lösungskompetenzen
  - “Verwandte Arbeiten” => wie nähern sich Andere dem Problem
3. R: Result
  - “Fachkonzept” => Lösungsansatz verständlich, nicht-technisch beschreiben
  - “Grobarchitektur” => Lösungsansatz grob-technisch (verständlich!) beschreiben
  - “Systementwurf” => Prototyp
  - “Technische Umsetzung” => ausgewählte Implementierungsdetails
  - [in Ausnahmefällen sogar Teile von “Stand der Technik” -> Lösungsbausteine als der Gegenstand der Betrachtung und Recherche]
4. E: Evaluation
  - “Evaluation” im engeren Sinn => rückblickend, quantitativ
  - “Diskussion” => kritisches reflektieren
  - “Ausblick” => vorwärtsblickend
  - (Disclaimer: “Discussion and Future Work” gehören nur im weitesten Sinn zur Evaluation; oft führen Publikationen die unmittelbar nach IMRE gegliedert sind ein “Discussion and Future Work” als zusätzliches fünftes Kapitel auf, und belassen damit «E» auch tatsächlich im engeren Sinn)

### Zielführung & Ergebnis

- Kein Botschaft an den Leser “Wir haben System X um Feature Y erweitert”. Das führt zu einer “How I did it!” Beschreibung ohne allgemeinen Wert.
- So gut wie möglich betonen “What do we want to demonstrate?” und daraus automatisch “What is the general interest?” oder auch “Why use foo to do bar?” (Methode/Artefakt-Struktur/...)
- Das “What do we want to demonstrate?” zunächst nicht als Aussagen sondern besser als Fragestellungen formulieren; Fragestellungen auf welche die Arbeit eine Antwort liefert
- Bei Use Case Beschreibungen nicht nur den Ablauf sondern wenn möglich eine User Story daraus machen, d.h. die Akteure und ihre Handlungsmotivation in den Ablauf einweben. Nicht “und dann passiert dies und dass” sondern “jetzt soll diese Aufgabe erfüllt werden (dieser Effekt bewirkt werden), dazu wird folgendes getan”

## LaTeX

Die Abschlussarbeiten werden i.d.R. in LaTeX erstellt.

Vorteile:

- Einmalige Einarbeitungszeit, max. 1 Woche, geringe Funktionalitätssuche während der Arbeit
- Keine Formatierungsprobleme gegen Ende der Arbeit, sowie problemlose Indizierung
- Konzentration gilt beim Editieren dem Inhalt und wird nicht durch die Formatierung abgelenkt
- Schönes und anspruchsvolles Layout ;)

Für die erfolgreiche Verwendung des Templates unter Windows benötigen Sie folgende Programme bzw. Umgebungen:

- **MikTeX** (<http://www.miktex.org/>): Umgebung für LaTeX. Stellen Sie nach der Installation den Package Manager auf einen Webmirror (FTP) Ihrer Wahl ein, damit zusätzlich benötigte Packages automatisch installiert werden können.
- **Sumatra PDF** (<https://www.sumatrapdfreader.org/>): PDF Viewer. Für LaTeX-Zwecke besser geeignet als Adobe Reader, weil er aus TeXnicCenter heraus an die korrekte Stelle springt... sehr bequem!
- !! Sowohl MikTeX als auch Sumatra PDF als ersten (**vor** TeXnicCenter) installieren, da sich dann TeXnicCenter beim ersten Start gleich für beides konfiguriert
- **TeXnicCenter 2.0** (<https://www.texniccenter.org/>): LaTeX-Editor. Hierbei müssen Sie die übliche Konfiguration durchführen (wo liegt miktex, wo liegt der pdf-viewer etc.) und sofern Sie auch z.B. ein Abkürzungsverzeichnis verwenden möchten, den Autoclean um entsprechende Einträge erweitern. Laden Sie die .tcp-Datei in TeXnicCenter und versuchen Sie sie anschliessend nach PDF zu rendern. Hierbei müssen Sie bis zu **3mal den Renderingvorgang** durchlaufen lassen bis alle Verweise aufgelöst und Hyperlinks korrekt eingebunden wurden. Beim Übersetzen kann es dazu kommen, dass weitere, noch nicht vorhandene Packages verlangt werden. Sofern Sie den MikTeX Package Manager richtig eingestellt haben, werden diese fehlenden Packages vom eingestellten FTP-Server runtergeladen. Nach dem Rendern sollten Sie beim Template 0 Errors und 0 Warnings haben (außer ggf. badbox-Warnings, die meist harmlos sind).
- BibTeX vs. **BibLaTeX**: Ich selbst hab v.a. mit BibTeX Erfahrung. BibLaTeX ist moderner. Ein wichtiger praktischer Unterschied ist, dass BibTeX den Eintrag "year" verwendet und BibLaTeX stattdessen den Eintrag "date" verwendet, dabei muss "date" in ISO8601 formatiert sein!
- **JabRef** (<https://www.jabref.org/>): Verwaltung der Literatur als BibTeX/BibLaTeX. Die verwendete Literatur können Sie im BibTeX- oder BibLaTeX-Format leicht in LaTeX selbst verwenden. Zur bequemen Editierung der Bib-Einträge empfehle ich Ihnen die Verwendung von einem Editor, z.B. JabRef. Bitte nehmen Sie auch nicht verwendete, aber bearbeitete Literatur mit in Ihr eigenes Literaturverzeichnis auf. In der Ausarbeitung selbst erscheinen jedoch nur die, welche auch wirklich im Text referenziert wurden. Ein Tip von mir wäre noch zu jedem gelesenen Paper eine kurze Zusammenfassung zu schreiben, da dies Ihnen ein späteres Auffinden (z.B. nach 3-4 Monaten) enorm erleichtert. Notieren Sie auch gleich ab Beginn alle Ihre Papers und Bücher als Bib-Einträge. Das Thema BibTeX/BibLaTeX ist nicht idiotensicher, und die Verwendung von JabRef entbindet Sie nicht davon das BibTeX- bzw. BibLaTeX-Format auch nativ verstehen zu müssen.
- Wenn Sie sich für eine BibLaTeX-basierte Tool-Chain entscheiden, dann unterstützt auch JabRef das technisch. Aber! Sie haben trotzdem einen Stolperstein: Sie müssen fast immer die aus Scholar o.ä. importierten/übernommenen Einträge erweitern. Die enthalten meist nur das Jahr. Sind nämlich meist dem Ursprung nach BibTeX. Sie müssen also dann das Year zum Date machen, durch Ergänzung von Monat+Tag. JabRef macht das nicht automatisch für Sie, denn welchen -MM-DD sollte es denn nehmen? Tja und welche nehmen Sie? Da meist nur die Jahresangabe auch tatsächlich im generierten LaTeX-Text auftaucht, können Sie auf gut Glück "-01-01" anhängen; Sie müssen aber nachkontrollieren. Wenn Sie das Jahr ohne Ergänzung übernehmen, dann wirft biblatex eine Warning (Obacht: bibtex- und biblatex-Warnings NIE ignorieren!) und im Literaturverzeichnis fehlt die komplette Jahresangabe!! Das wäre schlecht.
- (Beachten Sie auch die Literaturverzeichnis-Hinweise im Abschnitt "Kindergartenfehler", bspw. wg. Groß-/Kleinschreibung)

Anstelle einer lokalen Infrastruktur gibt es auch Cloud-basierte LaTeX-Umgebungen:

- **Overleaf** (<https://www.overleaf.com/>): Kostenlos für LaTeX-Dokumente mit einer "Compile timeout" von 1 Minute

Machen Sie sich bitte frühzeitig mit LaTeX vertraut um Probleme noch rechtzeitig aufzudecken, damit Sie nicht gegen Ende evtl. in Stress kommen.

## Zeichnungen/Grafiken

- **draw.io/diagrams.net** (<https://app.diagrams.net>): Universelles Zeichenwerkzeug, u.a. auf Augenhöhe mit Microsoft Visio.
- Entweder als Cloud-WebApp
- Oder als Desktop-Anwendung (Unter Windows bspw. mittels chocolatey: `choco install drawio`)
- **Cloudcraft** (<https://www.cloudcraft.co/>): Visualisierung von Cloud-Architekturen. Als Cloud-WebApp.
- **PlantUML** (<https://plantuml.com/de/>): Zur Erstellung von **UML**-Diagrammen auf Basis einer textuellen Notation. Bspw. Sequenzdiagramme, Klassendiagramme und Komponentendiagramme. Basiert auf Java und GaphViz/dot, bzw. anstelle von GraphViz mittlerweile auch optional auf Basis von Smetana.
- Kommandozeilen-Anwendung (Unter Windows bspw. mittels chocolatey: `choco install plantuml`)
- **Adobe XD** (<https://www.adobe.com/de/products/xd.html>): Es gibt noch (Stand Februar 2022), etwas versteckt, eine kostenlose Version namens **Starter Pack** zum Download mit Adobe ID aber ohne Abonnement (dafür aber mit etwas reduzierten Features): <https://www.adobe.com/products/xd/pricing/start-plan.html>
- **MathCha.io** (<https://www.mathcha.io>): Eine Cloud-WebApp zum zeichnen von **mathematischen** Grafiken. Erlaubt den Export nach SVG und sogar nach Tikz (Latex) Format.
- **Microsoft PowerPoint**: Unterschätzen Sie PowerPoint nicht als **universelles** Zeichenwerkzeug! Es ist ebenfalls mein Lieblingswerkzeug für **Poster** (DIN A1).
- **Microsoft Visio**: Ein Klassiker. Kommerziell. Eine Desktop-Anwendung. Mittlerweile durch draw.io ersetzbar.
- **Adobe Illustrator**: 1) Man kann damit gut zeichnen. 2) Illustrator erlaubt das Öffnen von PDFs und eignet sich dadurch gut zur Extraktion von Grafiken, ggf. in hochauflösendem Vektorformat.

## Orga

### Leistungen

#### Bachelorseminar

- Der Vollständigkeit halber erwähnt, wird hier nicht weiter detailliert

#### Lösungskonzept

- Es muss ein IT-fachlicher Erkenntnisgewinn, der für jeden Leser nachvollziehbar ist, erreicht werden (die berühmte wissenschaftliche ‘Contribution’)
- Bei prototypische Implementierungen muss die Ausarbeitung v.a. das Design darstellen (Subsystem-Strukturen, Zusammenhänge, Verantwortlichkeiten, Schnittstellen, Datenstrukturen, etc.) und weniger die Konstruktion (Implementierung; v.a. ‘private’ Code, Helferklassen, etc.)
- **ACHTUNG**: Oft vergeben wir in Ausschreibungen eher spezifische Aufgaben anstatt wissenschaftliche Fragestellungen — um das Ziel möglichst konkret zu halten und damit sich der Bearbeiter nicht im Fachgebiet verliert; dennoch muss jeder Bearbeiter beweisen (= in der Ausarbeitung entsprechend argumentieren), dass er die Aufgabe und die darin enthaltenen Probleme als beispielhaft begreift und die Aufgabe einer höheren Fragestellung unterzuordnen vermag

#### Literaturrecherche

- Verwalten von bibliographischen Einträgen
- Natives BibTeX-Format (bspw. JabRef) bzw. Werkzeuge wie Mendeley, EndNote oder Citavi [Export-Möglichkeit nach BibTeX ist obligatorisch]
- Web: **Google Scholar** (<https://scholar.google.com/>), **ResearchGate** (<https://www.researchgate.net/>), **CSAuthors** (<https://www.csauthors.net/>), **CiteSeer** (<https://citeseerx.ist.psu.edu/index>) sowie die drei Informatik-Primärquellen: **SpringerLink** (<https://link.springer.com/>), **ACM** (<https://dl.acm.org/>), **IEEE** (<https://ieeexplore.ieee.org/Xplore/home.jsp>)

- Applikation: **Publish or Perish** (<https://harzing.com/resources/publish-or-perish>) auf harzing.com (Autoren)
- Suche in den Literaturverweisen gefundener Quellen
- Suche auf den Homepages der Autoren
- Bei Bezugsschwierigkeiten (bspw. kostenpflichtig): Autoren per Mail kontaktieren und höflich um eine elektronische Fassung bitten (funktioniert sehr häufig!)
- **HowTo Literaturrecherche?** Ist sehr einfach: 1) Sie können über Stichworte suchen. Oder 2) über Autoritäten, d.h. Sie erlangen Kenntnis darüber, dass eine Person im Fachgebiet wichtig ist, dann erschließen Sie sich die Liste all seiner Publikationen. Oder 3) navigierend, durch die Literaturverweise bereits gefundener Quellen. Alle drei Verfahren wenden Sie wiederholt und ineinander verschränkt an. Und zwar so lange bis Sie signifikant häufig auf Publikationen treffen, die Sie bereits gesehen haben. Das wars, das ist der ganze iterative Algorithmus mit fuzzy termination.
- Anmerkungen zum HowTo: Dabei verfeinern Sie auch Ihren Pool an Stichwörtern und mehrten Ihren Pool von Autoritäten. Meistens ist ein Paper von mehreren Personen geschrieben. Nehmen Sie sich also jeden Autor des Papers heraus und suchen bei ihm mindestens in der Zeit etwas nach vorne. Wieso nicht auch nach hinten? Meistens sind vergangene Autorenwerke durch Selbstzitate in den referenzierten Quellen bereits subsumiert. Also natürlich müssen Sie bei jedem Autor auch nach hinten in der Zeit gehen, aber explizit wichtig ist es unter den Nachfolge-Publikationen fischen zu gehen.
- Anmerkungen aus Workload-Sicht: Meist finden Sie zu jedem beliebigen Thema durch diesen Algorithmus mehrere Hundert relevante Publikationen. In einer Dissertation werden Sie diese tatsächlich lesen, in einer Bachelorarbeit werden Sie Workload-getrieben Ihre Fähigkeit zum Querlesen trainieren (müssen) um effizient die überdurchschnittlich wichtigen Publikationen/Autoritäten/Stichwörter zu extrahieren. Das bereitet Sie auf die Dissertation vor, um dort später priorisieren zu können.
- Anmerkungen zur Anti-Verzettlungsmentalität: Die Spreu vom Weizen trennt sich natürlich primär durch den Inhalt. Dazu muss man aber die Publikation erstmal lesen. Ein weniger aufwändiger Lackmustest für die Qualität (und generische Relevanz) einer Publikation ist die Häufigkeit mit der andere Wissenschaftler die Quelle zitieren. Die Datengrundlage für diese Metrik ist heute durch Google Scholar et al. leicht erschließbar.
- Nochmal **für Dummies**: Am Anfang machen Sie eine erste Stichwortsuche und fischen sich eine vielversprechende wissenschaftliche Quelle heraus. Diese Quelle dient als Samen. Werfen Sie einen Blick auf alle dort referenzierten weiteren Quellen. Gehen Sie alle Autoren durch und werfen Sie einen Blick auf deren Publikationsliste. Rekalibrieren Sie Ihren Kenntnisstand und Ihre Stichwörter. Rinse and Repeat.
- Lesen Sie dazu auch den PhD Comics Strip No. 286: <https://phdcomics.com/comics/archive.php?comid=286>
- Weiterführende Methoden zur Qualitätsbewertung von wissenschaftlichen Publikationen: **Impact Factor** und **Rankings** von Publikationsorganen
  - Impact Factor: [https://en.wikipedia.org/wiki/Impact\\_factor](https://en.wikipedia.org/wiki/Impact_factor)
  - Autor-Metriken #1: <https://harzing.com/pophelp/metrics.htm>
  - Autor-Metriken #2: <https://harzing.com/publications/white-papers/reflections-on-the-h-index>
  - A. W. Harzing, R. van der Wal (2008) Google Scholar as a new source for citation analysis?, Ethics in Science and Environmental Politics, vol. 8, no. 1, pp. 61-73. URL: <http://www.int-res.com/articles/esep2008/8/e008p061.pdf>

### Prototypische Implementierung [i.d.R.]

- Abhängigkeiten zu 3rd-party Bibliotheken nachvollziehbar
- Build-Umgebung nachvollziehbar (z.B. per maven)
- Selbstverständlich: Modularisierung, Entkopplung von SW-Bausteinen, Konfigurierbarkeit, Anwendung von Entwurfsmustern
- Regelmäßige Kontrolle (bspw. per Maven-Reports) von Test-Code-Coverage, Code-Style, Code-Metriken und Javadoc-Vollständigkeit
- Dokumentation vollständig: mit Sorgfalt gepflegte exportierte Subsystem-Schnittstellen, Subsystem-interne public-Schnittstellen sind zumindest knapp kommentiert, Referenzen auf andere Klassen (als Klassen-Member) dokumentieren das Zusammenspiel (den Zweck, die Kollaboration) zwischen beiden Klassen

- Testklassen (z.B. per JUnit) vollständig: exportierte Subsystem-Schnittstellen werden mindestens mit einem Aufruf getestet

### **Demonstration** [ggf.]

- Ablauf der Demonstration ist strukturiert und eingeübt; der Ablauf macht klar welche Eigenschaft veranschaulicht wird
- Beispieldaten sind mit Liebe erstellt

### **Ausarbeitung**

- Die Länge der Arbeit ist kein Maß für deren Güte
- Gezählt wird von Introduction bis Conclusion (und nicht die vielen umrahmenden Seiten wie Abstracts, TOC, Appendix, Bibliography, etc. pp.)
- Richtlinie für Bachelorarbeit ~50 Seiten
- Richtlinie für Masterarbeit ~80 Seiten
- Bei Recherche- und vergleichenden Evaluations-Arbeiten ist die Seitenzahl naturgemäß höher, da die Aufarbeitung des State-of-the-Art sehr leicht Seiten produziert. Kurze, zielgerichtete Zusammenfassungen sollten in der Arbeit vollständig erfolgen auch wenn dadurch ein höheres Seitenvolumen zu Stande kommt. Zusätzlich muss aber der “synthetische Teil” (der Erkenntnisgewinn aus dem Konzeptvergleich) noch signifikanten Anteil haben, unabhängig von der Seitenzahl in der ersten Hälfte. Das ist der kreative, konzeptionelle Teil. Mehr Seiten sind auch fair, denn bei der Recherche-Arbeiten fängt man deutlich früher an Bilder und Texte für die Arbeit zu produzieren.
- Bei Recherche-Arbeiten kriegt man keine Lorbeeren auf die Anzahl gelesener Paper! Es ist selbstverständlich, dass diese Anzahl hoch ist! Entscheidend ist die Fähigkeit deren Inhalt adäquat auf den Punkt zu bringen, die Zusammenhänge zwischen den Papern zu erkennen und aufzuzeigen (u.a. Begriffskonsolidierung) und aus der Gesamtschau der gelesenen Konzepte eigene Schlüsse ableiten zu können (u.a. Begriffs- und Konzeptbildung).
- Vollständigkeit und Nachvollziehbarkeit der Betrachtung muss gegeben sein
- Vollständigkeit der Betrachtung kann nur durch explizite und nachvollziehbare Abgrenzung eingeschränkt werden
- Bindung? Wird in den Formalia der EMI-Fakultät geregelt (z.B. im “Merkblatt Bachelorarbeiten”: “Format: DIN A 4, broschiert Arbeiten in Schnellheftern, in Ordnern und in anderer Form werden nicht angenommen.”)

### **Abschlussvortrag**

- Länge: {20 min Vortrag + 10 min Demo} oder {30 min Vortrag}
- Anschließend an den Vortrag Zeit für Publikumsfragen
- Bitte selbstständig um einen Termin kümmern

**Frage:** “In den Bewertungskriterien sind ja viele Merkmale aufgezählt. Was beeinflusst die Note letztlich am meisten?”

- Entscheidend ist die Systematik der Beschreibung und die Ausdruckskraft bildlicher Veranschaulichungen.
- (Dabei selbstverständlich ist eine gewisse inhaltlich Tiefe; man muss im Kern der Arbeit eigene Erkenntnisse systematisch darstellen, keine Banalitäten; reines Wiederkäuen gibt es zwar, aber nur im Related Work in maßvollem Umfang.)
- Am wichtigsten in der Informatik ist das Herausstellen von analytischen Fähigkeiten und die Fähigkeit zur Abstraktion (auf unterschiedlichen Ebenen).

**ACHTUNG:** den Titel können wir vor der Anmeldung am Prüfungsamt (PA) noch (gemeinsam) ändern; bspw. kürzen. Danach ist der Titel aber stets in der Langfassung zu verwenden, und mit das wichtigste an der Arbeit, weil das PA den Titel später prüft. (Und das ganze Deckblatt und die Unterschrift bei der Selbsterklärung.) Der Titel muss mit der Anmeldung übereinstimmen!! Der korrekte Titel ist quasi formal das wichtigste für euch. (Sie sollten allerdings auch Deckblatt und Selbsterklärung unverändert übernehmen, bspw. keine Ergänzung von Firmenlogos auf dem Deckblatt.)

**Bonus** sammeln: Das Schreiben eines internationalen wissenschaftlichen Beitrags adelt Abschlussarbeiten!

- Paper schreiben hilft dabei sich auf das Big-Picture zu konzentrieren und bei der Arbeit später eine reine “How I did it”-Systemdokumentation zu vermeiden

- Bei Publikationen steht der Student als Erstautor und der Professor als letzter Autor, ggf. der industrielle Betreuer als Zweitautor. Es wird vom Student erwartet, dass er die erste Rohfassung selbstständig beiträgt. (Kommt es später, nach Abgabe der Arbeit, zur Publikation der Inhalte ohne Mitwirkung des Studenten am Paper-Schreibprozess, ist er nicht Koautor sondern wird in den Acknowledgements erwähnt.)
- Der Student darf dann auch mitfahren (Workshop/Konferenz). Das Department unterstützt Studenten dabei i.d.R. finanziell, auch wenn ein gewisser Eigenanteil verbleibt.

## Erste Schritte

- Exposé schreiben
  - Bei Kooperationen mit der Industrie: Exposé vor dem ersten Treffen zw. Erstprüfer, Industriebetreuer und Bearbeiter; als Grundlage
  - LaTeX-Umgebung aufsetzen
- Mit GIT vertraut machen (ggf. vor mir das git-Repo auf GitHub anlegen lassen)
- [Frühestens jetzt beginnt die eigentliche Bearbeitungszeit]

## Zwischenspiel

- Denken Sie dreieinhalb Monate vor Abgabetermin daran, dass Sie sich beim **Arbeitsamt** melden! Ansonsten entsteht Ihnen eine Sperrfrist für die Bezüge, falls Sie eine Lücke zwischen Studium und Berufseinstieg haben.
- Denken Sie daran, Ihre Bewerbungsunterlagen rechtzeitig, also parallel zur Abschlussarbeit, fertig zu stellen. Unterschätzen Sie nicht den dafür nötigen Aufwand! Bei Bachelorarbeiten bieten sich häufig spätestens die Weihnachtsferien an. (Gehen Sie daher davon aus, dass Sie die Weihnachtsferien zur Entspannung nutzen und für Ihre Bewerbungsunterlagen; planen Sie bitte auf keinen Fall Tätigkeiten für Ihre Abschlussarbeit in den Weihnachtsurlaub.)
- **Lesen Sie dieses Hinweis-Dokument erneut.** Genau genommen: Lesen Sie es während der Arbeit drei mal: Einmal zu Beginn. Einmal, ungefähr jetzt, vor dem Beginn der Schreibtätigkeit und noch ein letztes Mal zwei Wochen vor Abgabetermin!

## Selbstkorrektur / Checkliste

- Wurden alle **Kindergartenfehler** (s. unten) ausgebügelt!?
- Ist das Literaturverzeichnis mind. 3 Seiten lang?
  - Bei Bachelorarbeiten ist das ein absolutes Minimum - es sollte eigentlich umfangreicher sein
  - Bei Masterarbeiten sollte das Literaturverzeichnis deutlich umfangreicher sein

## Vorkorrektur / Vorab-Review

- Es gibt KEINE Vorkorrektur durch den Erstprüfer oder Zweitprüfer
- Einzelne Abschnitte können dem Erstprüfer dennoch zum Vorab-Review vorgelegt werden, allerdings in Summe nicht mehr als 10 Seiten (also 20% von geplanten 50 Seiten)
- WICHTIG: bereits während ihr schreibt: erklärt die Arbeit / Fragestellung / Konzepte euren Freunden und reflektiert daran, welche Aspekte bei der Beschreibung besonders relevant sind
- Das Vorab-Review ist ein nicht-garantiertes (Betreuer!) Angebot, das auch freiwillig (Student!) angenommen wird.
  - (übergibt mir eure Fassung einfach als PDF per Mail; der Ausdruck der Vorfassung soll euch nichts kosten daher mach ich das selbst)
- Ich limitiere die Review-Zeit auf 6h pro Abschlussarbeit (damit sind Reviews von 10 Seiten problemlos abbildbar)
  - Die Qualität und Lesbarkeit beeinflusst stark den Aufwand des Reviews
  - Ihr könnt unfertige Kapitel rausnehmen, müsst das aber nicht
  - Bitte Leseanweisungen/Prioritäten geben



- Die Vorfassung muss allerspätestens zwei Wochen vor offizieller Abgabe zum Vorab-Review durch den Erstprüfer vorgelegt werden.
  - Idealerweise wird das Angebot für ein Vorab-Review deutlich früher in Anspruch genommen.
  - Eine Woche brauche ich um es mir terminlich einzutakten.
  - Eine Woche braucht der Student um Änderungen noch einpflegen zu können und um die Anmerkungen/Erkenntnisse auf die anderen Teile der Arbeit zu übertragen
  - Die Initiative liegt seitens des Studenten (dass die Vorkorrektur durchgeführt wird): Wenn die Vorfassung zu spät bei mir ankommt (bspw. erst eine Woche vor Abgabe) lehne ich ein Vorab-Review und -Feedback auch ab.
- **Wichtig I:** Bitte vor Überreichung einer Vorfassung unbedingt nochmal den Abschnitt **“Kindergartenfehler”** (s. unten) durchgehen und diese vermeiden!
- **Wichtig II:** Vorher dafür sorgen, dass alles was ich lesen soll schon von einem Freund/Bekannten vorkorrigiert wurde!
- Mindestens eine Woche vor de-facto Zieltermin, also fünf Wochen vor der de-jure Abgabe (und zwei/drei Wochen vor meinem Review) Freunde/Bekannte darauf vorbereiten, dass sie euch die Arbeit vorkorrigieren sollen. **Jetzt bereits einen konkreten Termin mit ihnen vereinbaren** damit es am Ende nicht bei einem Lippenbekenntnis verbleibt!
  - Etwa vier Wochen vor de-jure Abgabe (zwei Wochen vor meinem Review) die Arbeit an die ausgewählten Freunde/Bekannte geben und zwei Tage später das Feedback einholen.
  - Sucht euch Leute die das ernst nehmen. Ein aufmunterndes “passt schon” als Feedback ist völlig nutzlos.
  - Vater und Mutter können zwar die Rechtschreibung korrigieren, aber ...
  - Echtes Review: Ihr braucht jemanden mit Informatik-Fachkenntnisse der euch inhaltlich kritisiert: Verständlichkeit, Aufbau (makro), Logik in der Argumentation (mikro)
  - Ein guter Reviewer muss mindestens 50 mal anmerken: “Häh?”, “unklar”, “Begriff fällt vom Himmel”, “konfus”, “stärker strukturieren”, “stimmt nicht”, “Logik?”, “Sinn?”, ...
  - **Sagt dem Reviewer das!!** Oft fehlt der rote Faden in Ausarbeitungen. Ein Reviewer muss dazu angeregt werden dem Autor aufzuzeigen wo dieser rote Faden verloren geht
- **Wichtig IV:** Für Recherche-lastige Abschlussarbeiten (u.a. auch Konzept-lastige oder Vergleich-lastige) gilt:
  - Eure Kollegen mit Prototyp-lastigen Themen schaffen sich Strukturen während sie programmieren (Package-Aufteilung, Interface-Gestaltung, etc. pp.) — wenn Sie zu schreiben anfangen haben sie sich Teile ihrer Arbeit bereits monatelang vorstrukturiert
  - Ihr tut dieselbe Strukturierungsaufgabe mit Konzepten (Gruppierungen, Überschriften, Begriffsbildung, etc. pp.)
  - Vorsicht: Verlasst euch nicht darauf, dass ich euch dabei umfassend helfe. Diese Abstraktionsleistung und analytische Leistung ist immerhin die Quintessenz der Arbeit, die ihr selbstständig leisten sollt.
  - Starthilfe: Lesen Sie im Kapitel “Wissenschaftliches Schreiben” den Abschnitt “Spezielle Hinweise”, denn hier findet ihr noch eine Reihe von Methodenbeispielen für Vergleiche und Funktionsumfangsdarstellungen
  - Bei analytischen Vergleichen solltet ihr euch mindestens mit der **Stiftung-Warentest** und deren Systematik und Darstellungsweisen auseinandersetzen, weil die meisten deutschsprachigen Leser diese kennen werden (Die Stiftung-Warentest nachzuahmen ist kein Erfolgsgarant für eine Abschlussarbeit, aber die Stiftung-Warentest nicht zu kennen oder zu ignorieren wäre fahrlässig inkompetent)
  - Botschaft: bei Recherche-lastigen Arbeiten muss ein privates Review möglichst früh beginnen (auf keinen Fall erst kurz vor de-jure Abgabetermin)
  - Tipp: schreibt so schnell wie möglich Text zu Teilaspekten der Aufgabenstellung — und gebt das dann einem Reviewer (auf Inhaltskorrektur)
- **Wichtig X:** Sorgt dafür, dass euer Literaturverzeichnis in der Arbeit von euch selbst korrekturgelesen ist (das wird oft vernachlässigt)
- Review-Zeichen:

- Eine einfache «Unterringelung» zeigt eine stilistisch unschöne oder schwer zugängliche Formulierung an.
- Ein Unterringelung mit «Fragezeichen» zeigt eine inhaltlich unklare/schwammige/unpräzise Formulierung an.
- Ein Unterringelung mit «Blitz» zeigt einen inhaltlichen Fehler an.
- Ein «Fehlt»-Zeichen (V) zeigt die Absenz eines wichtigen Aspekts an.
- Checkliste (**darüber ist mir Rechenschaft abzulegen**):
  1. Hab ich alle Fehler- und Stilfragen (**“Kindergartenfehler”**) beachtet?
  2. Welcher Freund/Bekannter hat die Arbeit bereits **inhaltlich** vorkorrigiert?
  3. Hab ich das **Literaturverzeichnis** in einen vorzeigbaren Zustand gebracht?

## Abgabetag

- Lesen Sie ein paar Tage vorher nochmals die \*\*Checkliste der EMI-Fakultät!
- Denken Sie vor Ihrem Abgabetag daran, dass Sie die von mir unterschriebene **Annahmebestätigung** benötigen!
- Per Formalia der Fakultät EMI: Die benötigten Fassungen abgeben
- Die elektronische, finale Fassung der Ausarbeitung als PDF per Mail an den Erstprüfer schicken – kann mittlerweile auch über Moodle abgegeben werden, dann bitte trotzdem kurze Info-Mail an den Erstprüfer, dass Upload vollzogen wurde
- Den Termin für den Abschlussvortrag spätestens jetzt einplanen
- Eine CD mit den elektronischen Quellen vorbereiten (sollte i.d.R. erst nach der Abgabe erstellt werden, siehe Touch-Down-Checkliste) – kann mittlerweile auch über Moodle als ZIP abgegeben werden, dann bitte trotzdem kurze Info-Mail an den Erstprüfer, dass Upload vollzogen wurde

## Verteidigung (Abschlussvortrag)

- Der Abschlussvortrag ist nicht die Präsentation im Rahmen des Bachelorseminars vor Ihren Kommilitonen während eines Semesters. Der Vortrag im Rahmen der Verteidigung erfolgt vor dem Erstprüfer und Zweitprüfer, kurz nach Abgabe, vor dem Abschluss der Korrektur/Benotung.
- Umfang von 20 bis 30 min
- Die Hörer abholen! Motivation des Projekts! (Einbettung in den Kontext, besondere Ziele, Nutzen für den Anwender)
  - Vorsicht vor Comics und Zitaten (kommt oftmals als Tipp): es ist ein wissenschaftlicher Vortrag und keine Kunden/Chef/Teambespaßung (ein erhabenes/interessantes Zitat ist natürlich trotzdem zum Auftakt möglich)
- Es ist trotz der Seriösität eines wissenschaftlichen Vortrags nur ein semi-formaler Rahmen
  - Anzug ist nicht nötig (sehen wir aber dennoch oft; wenn ihr euch damit wohl fühlt)
  - Die Ausdrucksweise sollte zwar sachlich sein, d.h. aber nicht, dass sie deswegen steif sein muss
- Methodik/Vorgehen zur Lösung
- Ergebnisteil: Konzeptionelle Ebene entscheidender als die Technische
  - Wegen Zeitknappheit auf die Kernkonzepte beschränken
- Foliengestaltung
  - Möglichst graphisch
  - Öfter auch mal zweispaltig (bspw. bei Gegenüberstellungen; aber auch dadurch zwei simple Folien zusammenfassen und dadurch die semantische Dichte erhöhen)
  - Für die Einbindung von Live-Web-Seiten in PowerPoint empfehle ich Poll Everywhere for Windows PowerPoint:  
<https://www.polleverywhere.com/app/powerpoint/win>
- Abschluss:

- Gerne eine Folie davor auch “Things Not Presented...” mit den Konzepten, Ansätzen, Erkenntnissen und Ergebnissen, die im Vortrag nicht dargestellt wurden aber in der Arbeit behandelt wurden (Auswahl so treffen, dass die Folie interessant ist und anregt die Arbeit zu lesen)
- “Fitness for Use” vs “Future Work”: was funktioniert bereits, was könnte man noch verbessern
- Die “Contribution” (Schlüsseleigenschaften) zusammenfassen, i.S.v. “USPs”: was ist das Besondere am Ergebnis, bspw. im Vergleich zu anderen Ansätzen
- Vorsicht vor: Pauschalisierungen, Anpreisungen, Parolen (sachlich bleiben)
- Demo gut vorbereiten
  - Achtung: Netzwerk/Firewall! (evtl. vorher mit mir absprechen)
  - Die Zuschauer abholen: anfangen, so wie auch ein End-Anwender anfangen würde
  - Immer erst die Funktionalität aus Sicht des End-Anwenders zeigen, dann technische Stärken/Details ergänzen
  - Die Demo kurz und übersichtlich halten; sich ein eigenes Storyboard/Drehbuch skizzieren
  - Nicht alle Funktionen müssen demonstriert werden (nur die wichtigen zeigen, ansonst evtl. Fragen abwarten)
  - Man muss deutlich mehr Demo geistig vorbereiten (in petto haben), als man dann zunächst von sich aus vorzeigt
  - Den Source-Code zu demonstrieren ist in der Regel nicht nötig (und man sollte ihn nur zeigen, wenn man darauf vorbereitet ist, die Zuschauer spannend durch den Code zu führen; kein spontanes/nicht-wirklich-vorbereitetes “ich könnte auch den Code zeigen”)
  - Tipp: eigene Agenda-Folie für die Demo erstellen (oder auf Flipchart vor Ort vorbereiten)
- Wichtig: mir deutlich vorher Bescheid geben, wenn ein Laptop gestellt werden muss!

## FAQ

- Alle Druck-Fassungen bunt? Nein, eine bunte Fassung reicht.
  - Vorsicht: kann recht teuer sein (~30€)! Vorher mit dem Copy-Shop verhandeln.
  - Evtl. alle Fassungen schwarz/weiß drucken lassen, dann die bunten Seiten einmal extra drucken (bspw. per PDF-Tools seitenweise Extrakte). Anschließend einsortieren, danach erst binden lassen.
  - **WICHTIG:** bei LaTeX darauf achten, dass dem **xcolor**-package die Option “**cm**yk” mit übergeben wird
    - \* (bspw. `\usepackage[pdftex,cm,yk,fixpdftex,hyperref]{xcolor}`)
    - \* **Nur dadurch** ist schwarzer Text auch **Druckschwarz** (100%-K-only); ansonsten ist es **RGB-Schwarz**, was im Druck/Copyshop als Farbe gilt, wodurch alle Seiten als Farbseiten zählen würden, was die Sache teuer machen würde!
    - \* **ACHTUNG:** Weder die Vorlage von Prof. Aßmuth noch die Vorlage von Prof. Frenzel beinhalten diese wichtige Einstellung! In beiden taucht das xcolor-package auf, aber nur mit `\usepackage[svgnames,table,hyperref]{xcolor}`; ergänzen Sie dort “cm”yk”!
- Deckblatt: “Mit Klarsicht oder einfach auf Karton gedruckt oder ...?” Völlig egal: Deckblatt sichtbar, egal wie!
  - “Welche Farbe soll das Deckblatt haben?” Eine schöne! ;-)
- Papiergewicht? 80g/m<sup>2</sup> reicht, also normales Büropapier
  - Wer Spaß an Ästhetik hat: ich nehme auch schwereres Papier welches oft glatter/hochwertiger ist (90g/m<sup>2</sup> oder 100g/m<sup>2</sup> sind manchmal beliebt)

## Touch-Down Checkliste

Tendenziell ist das nach dem Abgabetag:

- Die Abschlussarbeit benennen: als PDF in Form von **BT\_\_Nachname\_\_Vorname.pdf** (BT = Bachelor Thesis) oder **MT\_\_Nachname\_\_Vorname.pdf** (MT = Master Thesis)
- Eine **SelfRef-BiBTeX-Datei** anlegen, namensgleich zur PDF-Datei, nur mit **.bib** Dateiendung: enthält den BibTeX-Eintrag zur eigenen Arbeit, zur Wiederverwendung durch nachfolgende Arbeiten
- Quellen ablegen

- Die **TeX-Quellen** (oder die Word-Datei)
- **Alle Grafiken** zur Abschlussarbeit ablegen.
- Insbesondere die Quelldateien der Grafiken (Zeichenprogrammdateien, z.B. Visio, dia, etc.) für die spätere Wiederverwendung der Grafiken. \* Aber auch UML-Projektdateien bspw. von StarUML oder Visual Paradigm als Basis für verwendete Grafiken. \* Auf eine sinnvolle Benennung der Grafiken achten!
  - Alle elektronisch verfügbaren **Literaturquellen** (PDFs, HTML-Dokumente, etc.) der Bibliography
- Den Build-Prozess dokumentieren (build.txt)
  - Der build-Prozess muss automatisiert durchführbar sein (bspw. per ant oder maven).
  - Ungewöhnliche Aspekte sind im Maven-Modul als readme.txt zu dokumentieren
- Den Installationsprozess dokumentieren (install.txt)
  - Stolpersteine und Inbetriebnahme
  - Kurze Bemerkung dazu im Maven-Modul in Form einer readme.txt dokumentieren
- **Quellcode aufräumen**
- Software-Dokumentation vervollständigen
  - Bspw. in Java: JAutodoc hilft, aber zusätzlich Schlüssel-Klassen mit Sorgfalt beschreiben
  - Warnings reduzieren
    - \* bspw. in Java: unused imports » “Ctrl+Shift+O”, raw types et al. » @SuppressWarnings, ...
- Testklassen vervollständigen
  - Bspw. Maven Cobertura-Reports zur Test-Code-Coverage
  - Demodaten liegen in Java bspw. stets in src/test/resources
- Code Qualität (erst- oder letztmalig) prüfen, bspw. in Java:
  - Findbugs (am besten als Elipse Plug-In, aber auch als Maven report)
    - \* Maven JavaNCSS-Reports für Code-Metriken wie Commenting Source Statements (NCSS) und zyklomatische Komplexität
    - \* Maven PMD&CPD und Checkstyle zur Eliminierung typischer Quellcode-Fehler
- Termin für **Quellcode-Übergabe** vereinbaren
  - Am besten immer wieder begleitend zur Arbeit
  - Auf jeden Fall:
    - \* Bei genügend zeitlichem Freiraum drei Wochen vor der Abgabe.
    - \* Auf jeden Fall nach Abgabe der Arbeit, vor der Korrektur.
  - Eventuelle Nachbesserungen (Modulstruktur, build-Prozess, Installation und Quellcodedokumentation) sind nicht ausgeschlossen.
- Eine **CD** brennen und abgeben/nachreichen
  - Zweimal brennen: einmal für den Erstprüfer, einmal für den Bearbeiter (also euch selbst) — glaubt mir, ihr werdet einmal froh sein für das CD-Backup, das ihr euch selbst erstellt!
  - Etliches davon kann man erledigen, indem man einfach einen unveränderten git-Dump nimmt!
  - Inhalte:
    - \* Elektronische Originalfassung der Ausarbeitung (Word, Latex)
    - \* Elektronische Originale der Zeichnungen/Grafiken
    - \* Elektronische Druckfassung der Ausarbeitung (PDF)
    - \* Bibliographie-PDFs (“biblib”)
    - \* Sourcecode

- \* Kompilierte Version (inkl. Startskripten etc. pp.)
- Die CD kann mittlerweile auch über Moodle als per ZIP abgegeben werden! Sie sollten ein Exemplar der CD trotzdem physisch sich selbst JETZT brennen.
- Die **Präsentation** zur Abschlussarbeit ablegen.
  - Die Präsentation ist in der Regel erst Wochen nach der CD fertig
  - Sie muss nur am Ende im git-Repo stehen, selbstständig dran denken und per Mail nur kurz Bescheid geben
- Alle Dateien derart **privat sichern**, dass ein Löschen des git-Repos jederzeit nach dem Vortrag möglich ist
- Ausgeliehene **Bücher** zurückgeben
- Eine Liste-offener-Punkte als **Beiblatt** schreiben (formlos, bspw. Word-Datei)
  - Ebenfalls namensähnlich zur PDF-Datei und parallel abgelegt (bspw. BT|MT\_Nachname\_Vorname\_LoP.doc)
  - Eine Liste bekannter Bugs (tauch gerne noch bei der Vortragsvorbereitung/Demovorbereitung auf, wenn möglich natürlich noch ausbessern, wenn das im Aufwand vertretbar ist)
  - Darüber hinaus eine Übersicht bekannter “hackeliger” Benutzungs/Testszzenarien. Gemeint sind Szenarien in die man als Benutzer/Tester gerne mal reinstolpert, in denen es zum Fehler oder zur Fehl-/Nicht-Funktion kommt, die man aber durch Expertenwissen geschickt vermeiden kann/könnte (“dies und das funktioniert momentan nur genau dann, wenn man vorher XY beachtet”)
  - (darüber hinaus kann man hier auf konzeptioneller Ebene nochmal eine formlose Ergänzung zum “Future Work” machen)
  - Und wichtig ist im LoP-Beiblatt auch ein Lessons Learned-Abschnitt: wenn ihr dieselbe Aufgabe nochmal lösen müsstet, was würdet ihr evtl. anders gestalten / nochmal neu überdenken (das kann hier unverblümt erfolgen, oder technisch detaillierter, als unter “Discussion” der Ausarbeitung)
- Den Betreuer zum Abschluss **per Mail informieren**, dass diese Touch-Down-Checkliste erfolgreich abgehakt wurde

## Wissenschaftliche Suchmaschinen

Bei der Literaturrecherche für Examensarbeiten können u.a. folgende Suchmaschinen hilfreich sein (keine Rangfolge!):

- Google Scholar:  
<http://scholar.google.de/>
- ACM Digital Library:  
<http://portal.acm.org/dl.cfm>
- DBLP Computer Science Bibliography:  
<http://www.informatik.uni-trier.de/~ley/db/index.html>
- Karlsruher Virtueller Katalog:  
<http://www.ubka.uni-karlsruhe.de/kvk.html>
- Microsoft Windows Live Search Academic:  
<http://academic.microsoft.com/>

## Wissenschaftliches Schreiben

Lesen Sie sich die folgenden Hinweise dreimal durch: vor Beginn der Arbeit, vor Beginn der Schreibphase und vor Ihrer Abgabe!

### Vermeidbare Fehler (aka “Kindergartenfehler”)

- Falscher Einsatz von “das” und “dass”

- Schwankender Numerus, also eine Mischung von Einzahl und Mehrzahl
- Lange und falsch gebaute Sätze (Aussagen gehören in Hauptsätze; Nebensätze dienen der Präzisierung und Verknüpfung)
- Fehlende Kommata vor oder nach Relativsätzen und erweiterten Infinitiven; es darf nicht passieren, dass es wegen falscher Interpunktion mühsam ist einen Text zu lesen
- Bitte Grundkenntnisse der Textverarbeitung in TeX oder Word aneignen! (Bsp.: In \*.tex-Dokumenten trennt man Absätze mit Leerzeilen voneinander! Sie müssen in LaTeX nach einem Punkt, welcher der Abkürzung dient, das Leerzeichen escapen, sonst glaubt LaTeX, dass es sich um ein Satzende handelt und macht ein sichtbar falsches Spacing! etc. pp.)
- Einführen von Abkürzungen, nicht vorhanden oder nicht einheitlich
- Bilder:
  - Bilder, Tabellen, Codelistings, etc. die im Text nicht beschrieben werden? Gehen gar nicht!
    - \* Man spricht auch von benötigter *textueller Verankerung* der Bilder/Tabellen/Codelistings/...
    - \* Bilder et al. sind essentiell, müssen aber dennoch textuell beschrieben werden (!!)
    - \* Die Beschreibung der Bilder muss derart vollständig sein, dass die Arbeit auch verständlich wäre, wenn man alle Bilder entfernt (!)
    - \* Machen Sie das. Immer. Fragen Sie mich erst gar nicht nach Ausnahmen!
  - Bei Bildern ist es günstig, wenn die Verankerung und Teile der Beschreibung vor dem Bild stehen
  - Aussagekräftige Bildunterschriften verwenden
  - Wenn Sie Abbildungen mitten in einer Seite platziert, dann müssen diese immer zwischen zwei Absätzen stehen (sie dürfen einen Absatz nicht “brechen”)
  - Schlechte Qualität der Bilder (weil Screenshots oder aus dem Netz oder unpassende Schriftart, die im Kontext der Arbeit negativ hervorsteht.) Lösung: Selber nachzeichnen! Fei echt. (Alternative zum nachzeichnen: Professionelle Bildbearbeitung, bspw. mittels Krita, zur Entfernung von Kompressionsartefakten, bspw. durch White-Thresholds oder Level, sowie zum Hochskalieren auf mind. 300ppi in Druckmaßen und zum Nachschärfen, damit das Bild nicht matschig erscheint)
  - Zu kleine Schrift innerhalb der Bilder (Lesbarkeit muss gewährleistet sein! Besser zu groß als zu klein) (wichtig: ausgeglichenes Verhältnis zw. Schriftgröße und Symbolgröße)
  - Bilder, die nicht selbst gezeichnet wurden, müssen eine Quellenangabe in der Bildunterschrift haben, bspw. “(Quelle: [23], S. 82)”; auch Abbildungen, die selbst nachgezeichnet wurden, benötigen einen Hinweis auf den geistigen Urheber in der Bildunterschrift, bspw. “(Quelle: [42], eigene Darstellung)”; einzig selbstkonzipierte und selbstgezeichnete Abbildungen haben keine Quellenangabe
  - Legende, Legende, Legende! (Ausnahmen: vgl. eigener Abschnitt oben)
- Schritt-für-Schritt Beschreibungen, die nicht durch eine einfache Grafik oder ein Sequenzdiagramm visualisiert sind, sind zu vermeiden (Gewünscht: Aufbrechen des Texts durch Grafiken zur Erhöhung der Lesbarkeit)
- URLs im Fließtext eher vermeiden (bspw. durch Kapselung als Fußnote oder als Literaturreferenz)
- Literaturverzeichnis:
  - Form-Fehler vermeiden (bibtex macht gerne Großbuchstaben klein; zu verhindern bspw. per extra-geschweifte-Klammerung)
  - In einer deutschsprachigen Abschlussarbeit müssen Seitenangaben per “Seite”/“S.” und ggf. “f”/“ff” erfolgen und NICHT mit dem englischsprachige Stil “pp.”; entsprechend korrekt muss bibtex konfiguriert sein
  - Firmennamen als Autoren werden unabsichtlich gekürzt (gewisse Literaturverzeichnis-Stile listen die Vornamen von Autoren nur abgekürzt mit einem Buchstaben, bspw. “C. Neumann”. Wenn der Autor eines Werks keine natürliche Person ist, sondern eine Firma oder eine Foundation oder ein Standardisierungsgremium etc. pp., dann musst du ebenfalls mit extra-geschweifter-Klammerung arbeiten. Bsp.: Ein `author = {Cloud Native Computing Foundation}` wird im Text unabsichtlich zu “C. N. C. Foundation”, nur mit

- extra-geschweifter-Klammerung, also `author = {{Cloud Native Computing Foundation}}`, bleibt der Autorenname voll erhalten.)
- Fehlende Jahresangaben vermeiden (immer ein Erstellungsjahr recherchieren; wenn nicht anders möglich, dann Abrufdatum; Literaturangaben ohne Jahreskürzel sind unzulässig), kann auch durch BibLaTeX entstehen bspw. wegen fehlender Migration des bib-Eintrags von bibtex-“year” (YYYY) auf biblatex-“date” (YYYY-MM-DD)
- Konferenzorte als Ortsangaben verwenden, nicht den Sitz des Verlags (meist falsch bei BibTeX-Einträgen aus der ACM-Digital-Library)
- Literaturverzeichnis: Wikipedia unzulässig (weder Brockhaus noch Wikipedia sind wissenschaftlich zitierfähig; Enzyklopädien gelten als Tertiärliteratur)
- Rechtschreibfehler (Trick: verwende OOo/Word als Korrekturhilfe per TeX2RTF (Win) oder latex2rtf (Lin))
- Ein einzelner Satz sollte nicht als eigener Absatz stehen! Zu vermeiden sind auch Absätze, die mehrere Botschaften an den Leser vermischen.
  - Tipp: jeder Absatz sollte eine Botschaft haben – wird diese Botschaft präzise dargestellt, wird i.d.R. automatisch mehr als ein Satz benötigt
- Aufeinanderfolgende Überschriften, ohne Bindetext, darf es nicht geben. Überschriften dienen ausschließlich der Strukturierung, d.h. Kapitel müssen kurz eingeleitet werden so dass die Struktur und Zielsetzung beschrieben ist. Denkhilfe: der Text muss auch lesbar sein wenn alle Überschriften fehlen.
- Kapitel ohne Zusammenfassung. Fast alle Kernkapitel benötigen eine eigene Zusammenfassung als letzten Abschnitt. Nur kurze und kaum strukturierte Einleitungs-, Methodik- oder Zusammenfassungskapitel nicht.
- Verwendung von Bindestrichen in Wörtern:
  - Bindestriche genau da einsetzen, wo das Zusammenschreiben die Lesbarkeit erschweren würde
  - Am wichtigsten ist die konsequente Verwendung exakt einer Variante durchgängig im Text [Tipp: eigene Liste für solche Wörter erstellen]
  - Kombinationen aus englischen Wörtern oder aus englischem Wort und deutschem Wort werden genauso behandelt, wie rein deutsche Wortkombinationen (Duden Rechtschreibregeln 41, 42; Stand: 2014-04-09)
- Nach Abschnittsüberschriften immer erstmal Text. Niemals Elemente wie Grafiken, Listen, Tabellen im unmittelbaren Anschluss an eine Überschrift.
- Schreiben Sie die Namen Ihres Erst- und Zweitprüfers richtig! Fei echt.
- Schreiben Sie Amtsbezeichnung (“Prof.”) und genaue Grade (“Dr.-Ing.”) Ihres Erst- und Zweitprüfers in förmlichem Kontext richtig; förmlicher Kontext ist u.a. die Titelseite, aber auch jede Seite mit Formularcharakter, bspw. mit Zeilen wie “Prüfer:”, “Aufgabensteller:” oder “Betreuungsperson:”; den genauen Titel können Sie meist von deren Webseiten übernehmen; referenzieren Sie mich in förmlichem Kontext bspw. mit “Prof. Dr.-Ing. Christoph P. Neumann”; lassen Sie sich im Personenvergleich dabei nicht davon irritieren, dass die Bindestriche nach einem Dokortitel mal stehen und mal nicht, oder dass einem Professor ggf. auch ein Dokortitel fehlt; vertrauen Sie der Schreibweise auf der jeweiligen Hochschul-Homepage
- Schreiben Sie eine Danksagung nicht für Ihre Prüfer, Betreuer oder Reviewer alleine, das ist nicht nötig; Danksagungen in Bachelorarbeiten sind eher unüblich (bis hin zu umstritten oder mancherorts untersagt), während Danksagungen bei mehrjährigen Promotionen (Buch-Veröffentlichung!) üblicher sind; wenn Sie trotzdem eine Danksagung schreiben, dann sollten Sie natürlich Prüfer und Reviewer nicht weglassen, aber alles weitere dazu sagt Ihnen dann ein gegoogelter Leitfaden für Danksagungen in Abschlussarbeiten; referenzieren Sie Professoren dann im Fließtext nicht mit vollen Graden, bspw. mich nur noch mit “Prof. Christoph Neumann”
- Leerzeichen kontrollieren! (Bsp.: Fehlt in LaTeX gerne mal vor Zitatangaben, aus Flüchtigkeit!)
- Wortabstände kontrollieren! (Verlassen Sie sich nicht vollständig auf die automatische Silbentrennung; betrifft LaTeX und Word; aktivieren Sie in Word die automatische Silbentrennung überhaupt erst einmal; kontrollieren Sie die Arbeit optisch und helfen Sie durch Trennungsempfehlung von

Wörtern aus der Folgezeile; kontrollieren Sie auch ihr Literaturverzeichnis; beschäftigen Sie sich ggf. mit der Trennung von langen URLs in Word bzw. LaTeX – ohne dabei zusätzliche Trennstriche in die URL einzubringen, welche diese ungültig machen würde)

### Stilistisch

- Wortwiederholungen nicht zwanghaft vermeiden! Technische und konzeptionelle Texte dürfen (müssen sogar manchmal) der Klarheit wegen dieselben Begriffe häufiger wiederholen
- Radikales eliminieren von Weichmachern (engl: should, suggests, it is believed, ...)
  - Jeder Weichmacher der stehen bleibt, muss sofort im Text begründet werden!
- Weitreichendes eliminieren von Füllwörtern (engl: also, hence, besides, respectively, so long, all in all, ...)
- Sparsamer Gebrauch von Bindewörtern (engl: furthermore, nevertheless, moreover, ...)
- Nüchterne Begriffe verwenden; emotionale Begriffe vermeiden (engl: huge, attractive, ...)
- Anglizismen vermeiden
- EN-DE-Wortkombis mit Sorgfalt überdenken

### Inhaltlich

- Wenn etwas beschrieben wird, dann muss die Beschreibung verständlich und klar sein. NICHT nach dem Motto: ich fühle mich verpflichtet etwas zu erwähnen, hab aber eigentlich keine genaue Ahnung ...
- Begriffe (alle, alle) einführen! Niemals neue Begriffe vom Himmel fallen lassen (Korrekturkürzel “FvH” = “Fällt vom Himmel”)
- Keine zyklischen Definitionen (“Prozessmodellierung ist die Modellierung von Prozessen.” => Nullaussage)

### Typographie und Gestaltung

- Ein Minus (Bindestrich) taugt nicht als Gedankenstrich
- Overfull-Boxes = am Rand überstehende Wortenden (zu beheben oft nur per leichtem Umbau der Satzstruktur, dabei aber nicht den Textfluss oder Inhalt zerstören)
- Schlechte Qualität der Bilder (weil Screenshots oder aus dem Netz oder unpassende Schriftart, die im Kontext der Arbeit negativ hervorsteht.) Lösung: Selber nachzeichnen! Fei echt. Hatten wir oben schon.
- Ebenfalls nochmal: Zu kleine Schrift in den Bildern (Lesbarkeit muss gewährleistet sein! Besser zu gross als zu klein) (wichtig: ausgeglichenes Verhältnis zw. Schriftgröße und Symbolgröße)

### Grobstruktur nach Prof. Neumann

IMRE allein reicht nicht um eine Ausarbeitung zu strukturieren. Dennoch verwenden wir gleich I, M, R und E als Referenzsystem auf das in den verschiedenen Teilen einer Ausarbeitung unterschiedlich stark Bezug genommen wird. IMRE funktioniert v.a. dann wenn ich eine eng-gefasste Botschaft argumentieren will, wie dies in Publikationen oft der Fall ist. Bei Arbeiten die einer Systemkonstruktion-Aufgabenstellung **ganzheitlich** gerecht werden soll, ist also nun folgende ergänzende Sicht auf die Gliederung einer Arbeit hilfreich. Die skizzierten Teile sind nicht unmittelbar als Kapitel-Überschriften zu übernehmen.

#### 1. Prolog [5%-15%]

- **Einleitung, Vorgehen**, ggf. (fachliche) Grundlagen
- **ENTSCHEIDEND: Motivation** der spezifischen Problemstellung/**Fragestellung**
  - Die Formulierung sollte nicht imperativ (als Aufgabe: “Ich Bearbeiter muss!”) sondern in Frageform (“Wie löst man?”) erfolgen
  - In der Frage muss sich ein **Erkenntnisgewinn** für jeden Leser widerspiegeln
  - Die Arbeit muss am Ende dieser Fragestellung gerecht werden
  - **ACHTUNG**: Oft vergeben wir in Ausschreibungen eher spezifische Aufgaben anstatt breite Fragestellungen um das Ziel möglichst konkret zu halten und damit sich der Bearbeiter nicht im Fachgebiet verliert; **dennoch** muss jeder Bearbeiter beweisen (= in der Ausarbeitung entsprechend argumentieren), dass er die Aufgabe und die darin enthaltenen Prob-



leme als beispielhaft begreift und die Aufgabe einer höheren Fragestellung unterzuordnen vermag

- = I M - -: «Idea» pt.1 (Motivation) & «Methods»
- 2. Stand der Technik / Literaturlauswertung [20%-30%]
  - Das Feld abstecken: (fachliche) **Grundlagen**, (technische) **Grundlagen** (= Disziplinen aus der Informatik, welche für die Arbeit relevant sind) und **Methods** (Verfahrensebene der Konstruktion, seltener in der Informatik auch im Projektvorgehen) sowie **Related Work** (konkrete Publikationen mit ähnlicher Zielsetzung)
  - Manchmal eignen sich auch getrennte Kapitel für Grundlagen, Stand der Technik, Methoden und Verwandte Arbeiten, je nach Schwerpunkt
  - = - M - -: wenn überhaupt, dann kann man das als «Methods» begreifen — der Vergleich hinkt: eigentlich taucht das in IMRE höchstens begleitend auf
- 3. Konzeption [~30%]
  - Der wissenschaftliche **Beitrag**, die berüchtigte ‘Contribution’
  - Hier muss man sich der Fragestellung allgemein/konzeptionell annehmen und einen allgemeinen/nicht-technischen Erkenntnisgewinn produzieren
  - Die logische Einbettung des spezifischen Themas in das Forschungsgebiet
  - Bilder, Bilder, Bilder! (Freihand in Visio! UML ist hier weniger relevant!)
  - Problem-/Lösungsbereiche und Ergebnisbausteine
  - Kein Botschaft an den Leser “Wir haben System X um Feature Y erweitert”. Das führt zu einer “How I did it!” Beschreibung ohne allgemeinen Wert.
  - So gut wie möglich betonen “What do we want to demonstrate?” und daraus automatisch “What is the general interest?” oder auch “Why use foo to do bar?” (Methode/Artefakt-Struktur/...)
  - [hierzu unbedingt das Büchlein “Studien-Arbeiten” von M. Deininger et al. lesen, Kap. 5.2]
  - = I - R -: «Idea» (Lösungsansatz) & «Result» pt.1 (Konzept)
- 4. Prototypische Umsetzung [~30%]
  - Grob: die handwerkliche Leistung, auch «**Ingenieursleistung**»
  - Prototypaufbau und -funktionsweise
  - Jedoch liegt das Hauptaugenmerk nicht auf der konkreten Implementierung sondern v.a. auf dem **Design**
  - (In etwa: Design/Feinentwurf [15%-25%] und ausgewählte Details aus der Implementierung/Code-Ebene [5%-10%])
  - ggf. inkl. technischer Evaluation [5%-10%] des Prototypen mittels Software-Metriken (Cyclo, LoC, NoM, NoP, Calls-Fan-Out, ANDC, AHH, ...) und Test-Metriken (Code-Coverage); diese Form der Evaluation ist die weniger wichtige und kann ggf. auch entfallen; die eigentlich wichtige Evaluation erfolgt auf Konzeptebene
  - = - - R -: «Result» pt.2 (Konstruktion)
- 5. Evaluation [~5%]
  - Auf Ebene der eigentlichen Ergebnisse der Arbeit, also auf Ebene der Konzeption und des Erkenntnisgewinnes (u.a. “Fitness for Use” Ihres Prototypen!)
  - Das ist die eigentliche Evaluation i.S.v. IMRE
  - = - - - E: «Evaluation»
- 6. Epilog [10%]
  - **Diskussion, Ausblick, Zusammenfassung**
  - taucht in IMRE nicht wirklich auf
  - In der finalen Zusammenfassung müssen Sie Ihren Ergebnisbeitrag noch einmal konkret auf den Punkt bringen (i.d.R. ist das durch die Motivation und die Fragestellung in Kapitel 1 vorbereitet und bringt letztlich v.a. Konzept und ggf. auch Umsetzung in einen finalen kompakten Überblick)

**! Die Prozentangaben sind nur unverbindliche Richtwerte**

## Allgemeine Ratschläge

- Grundlagen: hier besteht die Gefahr den Faden zu verlieren. Am besten wird das nicht zu lang und hat immer klaren Bezug zur eigenen Arbeit.
- Verwandte Arbeiten: Hier verliert man als Leser ebenfalls leicht den roten Faden. Wichtig ist, dass alle Arbeiten zur eigenen Arbeit in Beziehung gesetzt sind. Gut ist immer eine Produkt/Feature-

Matrix, die ganz deutlich macht, wo die existierenden Lösungen noch ihre Lücken haben und wo die Unterschiede liegen. Idealerweise zieht man also Schlussfolgerungen aus den verwandten Arbeiten für die eigene.

- Technische Evaluation: Auch wenn es schwer fällt, sollte man versuchen hier “harte Fakten” zu präsentieren, am besten also quantitativ evaluieren. Das ist bei Software Engineering oft nicht einfach, weil man Richtung Software-Metriken und Test-Metriken gehen muss. Bei einer System-Konstruktion sind natürlich auch Performance-Ergebnisse wichtig; damit kann ein Leser dann einschätzen, ob das System einsetzbar ist, auch wenn die Entwicklungsmethodik vielleicht im Vordergrund stand.
- (Konzeptionelle) Evaluation: Das ist die eigentliche Evaluation i.S.v. IMRE. Eigentlich sind hier Methoden aus der Wirtschaftsinformatik angemessen, bspw. der “Three Cycle View of Design Science Research” von Hevner in 2007 (Scandinavian Journal of Information Systems, Vol.19 Issue 2, Article 4). Bei Bachelorarbeiten mit einer prototypischen Entwicklungsleistung sprengt ein wissenschaftlich korrektes Vorgehen für eine solche Evaluation oft den Rahmen. Machen Sie sich trotzdem mit dem Anspruch einer ernsthaften Evaluation vertraut und agieren sie im Kleinen entsprechend.
- Zusammenfassung: Wirklich versuchen die eigene Arbeit mit etwas Abstand zu reflektieren. Auch die kritischen Punkte ansprechen.

## Spezielle Hinweise

- Für Recherche-lastige Arbeiten (u.a. auch Konzept-lastige oder Vergleich-lastige)
  - Bitte unter “Vorkorrektur” den Punkt “Wichtig IV” lesen!
  - Spezifische Methoden (für Vergleiche und Funktionsumfangsdarstellungen):
    - \* **Product-Feature-Matrix** (vgl. Stiftung Warentest!) (Bsp. auf Google Images: Competitive Matrix, Feature Comparison Cart)
    - \* **Morphologischer Kasten** (= Zwicky-Box)
    - \* **Nutzwertanalyse**
    - \* **Netzdiagramme/Kiviat-Diagramme** (engl. “Radar Chart”/“Polar Chart”) (techn. Bsp. (S.60ff); polit. Bsp.) (Wichtig: re-ordering!)
    - \* Portfolioanalyse/Produkt-Markt-Matrix (BCG, McKinsey, Ansoff-Matrix)
    - \* Multi-Level Pie Chart (Erweiterung von Kuchendiagrammen zur hierarchischen Darstellung)
    - \* Entscheidungsbäume
    - \* Ursache-Wirkungs-Diagramm (auch Ishikawa-Diagramm, engl. “fishbone diagram”)
    - \* Sankey-Diagramme
    - \* Harvey Balls (Beispiel)
    - \* ...
    - \* (Selber Impulse setzen! Geeignete Diagrammformen suchen/wählen!)
    - \* ...
    - \* Auch: Visual Complexity (meist zu “fancy” für Abschlussarbeiten, aber zwecks Inspiration anschauen!)
    - \* Kommerz. Software: **Excel** (inkl. Netzdiagramme), SmartDraw (z.B. Matrix Charts; jedoch **keine** Netzdiagramme), ConceptDraw Project (inkl. Netzdiagramme)