



# Software-Entwicklungspraktikum (SEP)

## ANLEITUNG

In diesem Dokument werden ein paar allgemeine Regeln für das Bearbeiten von Dokumenten vorgestellt. Diese sollen eine Hilfestellung für das erfolgreiche Erstellen der Dokumente sein.

Außerdem gibt es eine Anleitung zur Abgabe und Benennung der Dokumente.

**Bitte lest euch dieses Dokument sorgfältig durch.**

---

# 1 Allgemeine Hinweise

Sollte es in der Gruppe oder mit dem Betreuer Probleme geben die nicht eigenständig (oder mit Hilfe des Betreuers) gelöst werden können, wendet euch bitte an das ISF. Das sollte jedoch die letzte Instanz sein.

Grundsätzlich stehen die Hiwis immer für Fragen, egal ob zu Dokumenten oder anderen SEP bezogenen Problemen, zur Verfügung.

## 1.1 Dos and Don'ts in Dokumenten

Hier findet ihr nun einen allgemeinen Guide für die Dokumente. Grundsätzlich gilt, dass es sich um technische Dokumente handelt, die entsprechend formal und einheitlich sein sollten.

1. Die Vorlagen sind, wie der Name sagt, Vorlagen, dass bedeutet das in die Abgabe keine Vorlagentexte oder Hilfstexte gehören.
2. Ihr könnt die Vorlagen anpassen, dies muss jedoch mit dem Betreuer abgesprochen und begründet werden. Es dürfen keine grundlegenden Teile entfernt werden (im Zweifel mit dem ISF absprechen).
3. Jedes Kapitel oder Unterkapitel beinhaltet mindestens einen Satz und führt kurz in den Inhalt, bzw. den Zweck dieses Abschnittes ein.
4. Kapitelnamen sollten immer aussagekräftig sein und sich nicht wiederholen.
5. Die Abbildungen sollten eine genaue Bezeichnung haben und diese sollten sich nicht wiederholen.
6. Wenn ihr Abbildungen nutzt, sollten diese im Text (am besten direkt unterhalb der Abbildung) referenziert und kurz erklärt werden. Grundsätzlich gilt, JEDE Abbildung braucht eine Beschreibung!
7. Diagramme und Abbildungen sollten immer eine Vektorgrafik sein, damit sie bei jeder Skalierung des Dokumentes gut lesbar sind. Außerdem sollten die Diagramme den Platz ausnutzen und übersichtlich gestaltet sein, so können Querformat und Seitenübergänge verhindert werden.

- 
8. Zu Beginn der Projektphase sollte sich in der Gruppe auf ein Werkzeug zur Erstellung von Diagrammen geeinigt werden. Dies ermöglicht eine Arbeitsteilung bei gleichbleibendem Layout. Diagramme mit unterschiedlichen Werkzeugen sollten nicht im Dokument vorkommen. Solltet euer gewähltes Werkzeug ein Wasserzeichen auf die Diagramme setzen, benutzt ein anderes Werkzeug.
  9. Die Annotationen dienen dazu, Fehler oder Verbesserungen zu markieren. Es wird erwartet, dass vorhandene Annotationen in neuen Abgaben berücksichtigt bzw. behoben werden.
  10. Damit die Referenzen in den Dokumenten stimmen, sollten die LaTeX Dokumente vorsichtshalber immer mehrmals übersetzt werden.
  11. Wenn in vorherigen Dokumenten Funktionen bereits beschrieben oder erwähnt wurden, solltet in den nachfolgenden Dokumenten darauf verwiesen werden. Am einfachsten ist dies durch eine einheitliche, unique(einzigartige) Bezeichnung umsetzbar.
  12. Wie schon erwähnt, handelt es sich um technische Dokumente, daher sollte auch einem entsprechenden Sprachstil genutzt werden. Es sollten immer ganze Sätze geschrieben werden.
  13. Ist euer Dokument mehr als 100 Seiten lang, solltet ihr dieses von eurem Betreuer absegnen lassen, denn unnötige oder redundante Teile sollten nicht im Dokument sein.
  14. Es sieht schöner (einheitlicher) aus, wenn alle Teilnehmer die TU Braunschweig Emailadressen auf dem Deckblatt verwenden.

WICHTIG!!! Alle Dokumente müssen mindestens bestanden sein. Ihr habt nach der ersten unzureichenden Abgabe eines Dokumentes eine weitere Möglichkeit euer nachgebessertes Dokument erneut einzureichen.

---

## 2 Zu ändernde Dateien

Hier findet ihr eine Auflistung aller Dokumente, die ihr vor Abgabe des Dokumentes verändern solltet und Dateien, die ihr nicht verändern sollt/müsst.

### Zu ändernde Dateien

- **teilnehmer.tex:** In dieser Datei müsst ihr alle Teilnehmer eures Teams eintragen. Das Beispiel zeigt euch genau wie das aussehen soll.
- **config.tex:** Hier müsst ihr alle Daten eures Instituts eintragen.
- **Dokument.tex:** Dokument ist hier durch das jeweilig bearbeitete Dokument zu ersetzen, z.B. Pflichtenheft, Fachentwurf etc. Hier braucht ihr nur etwas zu verändern, solltet ihr ein Kapitel hinzufügen oder entfernen.

Die einzelnen Kapitel der Dokumente sind natürlich von euch zu bearbeiten und sinnvoll auszufüllen.

### Nicht zu ändernde Dateien

Diese Dateien müssen nicht angepasst werden. Solltet ihr jedoch ein weiteres Paket für LaTeX benutzen wollen, so müsst ihr die **packages.tex** anpassen.

- layout.tex
- packages.tex
- titelseite.tex

---

## 3 Dokumentenabgabe

In diesem Abschnitt findet ihr, wie die Abgabe der Dokumente geschehen soll.

Ihr habt am Abgabetag bis **23:59 Uhr** Zeit die Dokumente im Redmine hochzuladen. In eurem Projekt im Redmine gibt es den Reiter Dokumente. Legt dort bitte für das abzugebene Dokument ein „Neues Dokument“ mit dem Namen der Abgabe an. Ladet dort dann das Dokument in Form eines PDF-Dokumentes hoch.

Für das Angebot wird also ein neues Dokument mit dem Namen *Angebot* angelegt und dort die Datei hochgeladen.

Bitte achtet bei der Benennung der Abgabedokumente auf folgende Namenskonvention:

Institutskürzel\_Projektkürzel\_Gruppennummer\_Dokumententitel\_Versionsnummer

Um dies besser nachvollziehen zu können, hier einmal ein Beispiel. Das Institut ISF hat 2 Projekte. Das erste ist die Veranstaltungsverwaltung (kurz Verva) und das zweite eine Bibliotheksverwaltung (kurz Bib). Für das Projekt Verva gibt es zwei Gruppen (g1 und g2) und es ist die erste Abgabe des Dokumentes *Angebot*. Das abgegebene Dokumente für Gruppe 1 sollte also wie folgt benannt werden:

isf\_verva\_g1\_angebot\_v1.pdf

Bitte haltet euch an diese Namenskonvention, da die Korrekturen so die Dokumente leicht auseinander halten können und es zu keinen Verwechslungen kommen kann.

**WICHTIG!!!** Gibt es an eurem Institut nur ein Projekt, benötigt ihr kein Projektkürzel. Außerdem ist es möglich, dass es nur eine Gruppe gibt, dann wird auch die Gruppennummer nicht benötigt. Solltet ihr euch bei der Benennung unsicher sein, wendet euch einfach an das ISF.

---

## 4 Ziele der Dokumente

In diesem Kapitel werden die Ziele der einzelnen Dokumente kurz erläutert. Ihr solltet also die Dokumente so bearbeiten, dass sie dem Ziel entsprechen.

1. **Angebot:** Im Angebot werden die Rahmenbedingungen für das Projekt festgelegt. Außerdem kann der Kunde (Betreuer) nachvollziehen, ob die Ziele und wichtigsten Funktionen des Projektes verstanden wurden.
2. **Pflichtenheft:** Das Pflichtenheft dient dazu, nach dem Angebot die Ziele und Funktionen des Projektes zu verfeinern und die Anforderungen an das Projekt festzuhalten. Dieses Dokument dient als Referenz für den späteren Abnahmetest.
3. **Abnahmespezifikation:** In der Abnahmetestspezifikation wird das Vorgehen im Abnahmetest beschrieben.
4. **Fachentwurf:** In diesem Dokument wird das Verhalten des Systems beschrieben und die anfallenden Daten. Hier wird noch nicht die konkrete Umsetzung beschrieben.
5. **Technischer Entwurf:** Im Technischen Entwurf wird der Fachliche Entwurf weiter konkretisiert und die Details beschrieben.
6. **Testspezifikation:** Die Testspezifikation ist die Erweiterung der Abnahmetestspezifikation. Hier sollt ihr die zwei weiteren Testphasen (Unit-Test und Integrationstest) beschreiben.
7. **Testprotokolle:** In der Testdokumentation werden die von euch ausgeführten Testfälle protokolliert. Dies zeigt, dass ihr euer Projekt ausreichend getestet habt und dokumentiert die Testergebnisse.

---

## 5 Zwischenpräsentation

In der Präsentation sollten kurz das Team sowie das Thema vorgestellt werden. Anschließend soll ein Prototyp demonstriert werden. Es gilt hierbei in erster Linie darum, das Produkt möglichst gut zu verkaufen. Die gezeigten Funktionen müssen nur für die Präsentation funktionieren, können also hardcoded oder gefaked sein. Jede Gruppe hat 7 Minuten Zeit für ihre Präsentation und anschließend 2 Minuten um Fragen zu beantworten.

Nachfolgend ein paar Beispiele, in welche Richtung die Präsentation gehen soll:

`https://www.youtube.com/watch?v=9-nezImUP0w`

`https://www.youtube.com/watch?v=vS0la9SmqWA`

`https://www.youtube.com/watch?v=GRnVHKp6sPo`

`https://www.youtube.com/watch?v=rskC6c\_5L1M`

`https://www.youtube.com/watch?v=EHWRkuD1NOE`

---

## 6 Werkzeuge

Damit ihr bei den vielen Werkzeugen nicht alleine suchen müsst, haben wir euch hier eine kleine Liste zusammengestellt. Es ist nur eine Hilfestellung und keine Verpflichtung diese Tools zu benutzen.

### 6.1 LaTeX

Dieser Abschnitt listet ein paar Programme zur Erstellung von LaTeX-Dokumenten auf. Achtet bittet immer darauf, dass euer Editor die Zeichenkodierung UTF-8 benutzt. Ansonsten kommt es zu unschönen Zeichen im Dokument. Solltet ihr einen Befehl nicht kennen, gebt bei Google einfach "Latex ..." ein. Das "..." ersetzt ihr einfach durch die gewünschte Funktionen. Z.B. Aufzählung, kursiv schreiben, Grafiken einbinden etc. Ihr solltet dort sehr schnell eine Erklärung und ein Beispiel finden.

LaTeX Allgemein:

<http://www.tex.ac.uk/ctan/info/lshort/german/l2kurz.pdf>

MikTeX ist eine TeX/LaTeX Distribution für Windows.

Download: <http://www.miktex.org/download>

TeXStudio ist ein Editor zum Erstellen von LaTeX Dokumenten.(Plattform: Win/Mac/Linux)

Donwload: <http://texstudio.sourceforge.net/>

Hilfe zur Bedienung von TeXStudio:

[http://texstudio.sourceforge.net/manual/current/usermanual\\_en.html#SECTION1](http://texstudio.sourceforge.net/manual/current/usermanual_en.html#SECTION1)

TeXnicCenter ist ebenfalls ein Editor.(Plattform: Win)

<http://www.texniccenter.org/>

Es ist ebenfalls möglich Eclipse bis Version 3.7 als Editor zu benutzen. (Plattform: Win/Mac/Linux)

<http://texlipse.sourceforge.net>

Hilfstools:



---

Detexify ist eine Symbolsuche mit Handschrifterkennung. Hilft einem, wenn man mal den Namen eines Symbols nicht weiß, den passenden LaTeX Code zu finden.

Download: <http://detexify.kirelabs.org/classify.html>

## 6.2 SVN

In diesem Abschnitt findet ihr ein paar Links zum Thema SVN und einen Link zu einer Anleitung. Die einfachste Methode ohne weitere Tools zu installieren ist natürlich das Redmine direkt zu nutzen. Dort findet ihr auch die URL zum SVN eures jeweiligen Projekts, solltet ihr andere Programme nutzen wollen, wie z.B.:

**Subclipse**(Plattform: Win/Mac/Linux)

Installation Subclipse in Eclipse:

Help -> Eclipse Marketplace -> Search: subclipse -> install Subclipse 1.10.x Restart Eclipse

Für weiteres: <https://www.ibm.com/developerworks/library/os-ecl-subversion/>

oder: <http://realsearchgroup.org/SEMaterials/tutorials/subclipse/>

oder: <http://ist.berkeley.edu/as-ag/tools/usage/subclipse-usage-tips.html>.

**TurtoiseSvn**(Plattform: Win)

**Subversion**(Plattform: Mac/Linux)

SVN Anleitung:

<http://svnbook.red-bean.com/de/1.7/>

## 6.3 Diagramme

Hier findet ihr eine Auflistung von Werkzeugen, die ihr für das Erstellen von UML-Diagrammen benutzen könnt. Dies ist nur eine Auswahl und von euch können auch andere Werkzeuge verwendet werden, solange sie die Voraussetzungen erfüllen (alle Diagrammtypen und Möglichkeit zum Extrahieren von Vektorgrafiken).

- Dia - ein einfaches Zeichenprogramm (Plattformübergreifend) <http://dia-installer.de/download/index.html>
- Visio - eine Diagrammsoftware von Microsoft (Windows) <http://office.microsoft.com/de-de/visio/>. Kann über das GIZ bezogen werden: <https://www.tu-braunschweig.de/it/service-interaktiv/software/doku/msdn-aa/elms/login>

- 
- Visual Paradigm - ein gutes Zeichenprogramm (Windows) VORSICHT: In der freien Version sind Wasserzeichen auf dem exportierten PDF-Dokument. <http://www.visual-paradigm.com/download/vpuml.jsp>
  - TikZ und PGF - Pakete um in LaTeX Grafiken zu erstellen (Plattformübergreifend) <http://sourceforge.net/projects/pgf/>
  - Gliffy - ein Browsertool, welches jedoch nicht kostenlos ist, aber sehr gute Diagramme erstellt. <http://www.gliffy.com/>
  - Eclipse (Papyrus) - <http://www.eclipse.org/papyrus> (Plattformübergreifend)
  - UMLet - ist ein kostenfreies Zeichentool, welches alleine oder als Plugin in Eclipse genutzt werden kann. <http://www.umlet.com/> (Plattformübergreifend)
  - PlantUML - ein kostenfreier Editor indem in einer intuitiven Sprache die Diagramme definiert werden. <http://plantuml.sourceforge.net/index.html> (Plattformübergreifend)
  - draw.io - ist ein Browsereditor für Diagramme, <https://www.draw.io/> (Plattformübergreifend)

Braucht ihr noch einmal eine gute Erklärung der Diagramme empfehlen wir euch die Seite <http://www.uml-diagrams.org/>.