



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.

Sommersemester 2017

Inhaltsverzeichnis

1	Allgemeine Hinweise	2
1.1	Regeln für Studenten	2
1.2	Dos and Don'ts in Dokumenten	2
1.3	Bearbeitertabelle	4
2	Zu ändernde Dateien	5
3	Dokumentenabgabe	6
4	Ziele der Dokumente	7
5	Zwischenpräsentation	8
6	Werkzeuge	9
6.1	LaTeX	9
6.2	SVN	10
6.3	Diagramme	10

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 Regeln für Studenten

1. Jedes Gruppenmitglied muss zu allen Projektphasen inhaltlich beitragen.
2. Jedes Gruppenmitglied muss sowohl zum Code als auch zur Dokumentation beitragen.
3. Jedes Dokument muss **vollständig** bearbeitet abgegeben werden.
4. Jedes Gruppenmitglied ist für die Vollständigkeit der Dokumente verantwortlich.
5. Jedes Dokument, das nach der Deadline eingereicht wird, gilt als nicht eingereicht.
6. Die Teilnahme an allen offiziellen Veranstaltungen ist verpflichtend.
7. Alle SEP relevanten Dokumente werden ausschließlich im ISF Redmine/SVN versioniert.

1.2 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.

- Die Vorlagen sind, wie der Name sagt, Vorlagen. Das bedeutet, dass in die Abgabe keine Vorlagentexte oder Hilfstexte gehören.

-
- Die Beispieldokumente sind nicht immer fehlerfrei, jedoch alles Abgaben, die vom ISF als sehr gut oder gut eingestuft wurden. Die Beispiele sind als Hilfestellung gedacht und nicht zum direkten Kopieren. Es fällt beim korrigieren auf wenn Textpassagen einfach übernommen wurden.
 - 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).
 - Jedes Kapitel bzw. Unterkapitel beinhaltet mindestens einen Satz und führt kurz in den Inhalt, bzw. den Zweck dieses Abschnittes ein.
 - Kapitelnamen sollten immer aussagekräftig sein und sich nicht wiederholen.
 - Die Abbildungen sollten eine genaue Bezeichnung haben und diese sollten sich nicht wiederholen.
 - 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!
 - Diagramme und Abbildungen müssen 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.
 - 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. Sollte euer gewähltes Werkzeug ein Wasserzeichen auf die Diagramme setzen, benutzt ein anderes Werkzeug oder entfernt das Wasserzeichen.
 - Die Annotationen dienen dazu, Fehler oder Verbesserungen zu markieren. Es wird erwartet, dass vorhandene Annotationen in neuen Abgaben berücksichtigt bzw. behoben werden.
 - Damit die Referenzen in den Dokumenten stimmen, sollten die LaTeX Dokumente vorsichtshalber immer mehrmals übersetzt werden.
 - 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.
 - Wie schon erwähnt, handelt es sich um technische Dokumente, daher sollte ein entsprechender Sprachstil genutzt werden. Es sollten immer ganze Sätze geschrieben werden.

-
- Ist euer Dokument mehr als 100 Seiten lang, solltet ihr dieses von eurem Betreuer absegnen lassen, unnötige oder redundante Teile sollten nicht im Dokument vorhanden sein.
 - Es sieht schöner (einheitlicher) aus, wenn alle Teilnehmer die TU Braunschweig Email-adressen auf dem Deckblatt verwenden.

1.3 Bearbeitertabelle

Am Anfang jedes Dokuments gibt es eine Bearbeitertabelle, in die zu jedem Kapitel bzw. Unterkapitel die verantwortliche Person eingetragen wird. Diese Tabelle ist wichtig um sichergehen zu können, dass sich alle an dem Dokument beteiligt haben und niemand 'durchgeschliffen' wird, ohne an allen Teilen des Projekts mitgearbeitet zu haben. Für euch ist dies auch eine Sicherheit, sollte es zu Streit um die Bearbeitung der Dokumente kommen.

Haben zwei Personen an einem Kapitel gearbeitet, darf dies natürlich eingetragen werden, es soll aber nicht 'alle' oder unangemessen viele Namen eingetragen werden, nur damit jeder Name einmal in dem Dokument vorkommen. Das ist nicht Sinn dieser Tabelle und fällt bei der Korrektur auf.

WICHTIG

Alle Dokumente müssen mindestens bestanden sein. Unzureichend bearbeitet Dokumente werden als mangelhaft zurückgeschickt. 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 als PDF 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_Gruppennummer_Dokumententitel_Versionsnummer

Um dies besser nachvollziehen zu können, hier einmal ein Beispiel. Das Institut ISF hat 2 Projekte. Es gibt zwei Gruppen 8g1 und g2) die das Projekt Veranstaltungsverwaltung bearbeiten. Das Dokument für das Angebot in der ersten Version sollte wie folgt benannt werden:

isf_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.

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. In jeder Dokumentenvorlage ist nochmal ausführlicher erklärt, was erwartet wird.

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:** Zweck des Pflichtenheftes ist es die Ziele und Funktionen des Projektes verfeinert festzuhalten. Dabei sollen die Fragen WIE und WOMIT das System entwickelt werden soll beantwortet werden. Die Anforderungen an das zu entwickelnde System sollen dabei technisch (prägnant und klar strukturiert, keine Prosa) verfasst werden. Dafür soll der Umfang des Projektes anhand der Muss-, Soll-, Kann- und Abgrenzungskriterien klar spezifiziert werden.

Wichtigste Inhalte des Dokumentes:

- Allgemeine Produktbeschreibung / Zielbestimmung
- Leistungsumfang
 - Musskriterien: unabdingbare Leistungen der Software
 - Sollkriterien: erstrebenswerte Leistungen
 - Kannkriterien: Leistungen die enthalten sein können, denen der Auftraggeber jedoch neutral gegenüber steht
 - Abgrenzungskriterien: Leistungen die explizit nicht umgesetzt werden.
- 3. **Abnahmespezifikation:** In der Abnahmetestspezifikation wird das Vorgehen im Abnahmetest beschrieben. Dazu gehört ein Testplan und eine Beschreibung in welchem Umfang die Produktfunktionen getestet werden sollen.
- 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. Jedes Gruppenmitglied muss bei der Zwischenpräsentation anwesend sein! Anschließend soll ein Prototyp demonstriert werden.

Es geht hierbei in erster Linie darum, das Produkt möglichst gut zu verkaufen. Ihr sollt Vorfreude für euer Produkt schaffen und aus der Masse hervorstechen. Innovative Ideen und Konzepte sind gerne gesehen, lasst eurer Fantasie freien Lauf und versucht die Präsentation so interessant wie möglich zu gestalten, besonders um die Aufmerksamkeit des Publikums zu gewinnen und schon einmal einen Eindruck für den TDSE zu hinterlassen.

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. Haltet euch bitte an diesen Angaben, es tragen an diesem Tag eine Menge Gruppen vor, sodass es einen straffen Zeitplan gibt. Stellt außerdem sicher, dass alle technischen Geräte/Präsentationen funktionieren und ihr alle notwendigen Dinge dabei habt, z.B. ein für euren Laptop passendes Kabel um ihn an den Projektor anzuschließen oder Requisiten für eure Präsentation.

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=vS01a9SmqWA>

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

https://www.youtube.com/watch?v=rskC6c_5L1M

<https://www.youtube.com/watch?v=EHWRkuD1N0E>

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 bitte immer darauf, dass euer Editor die Zeichenkodierung UTF-8 benutzt. Solltet ihr einen Befehl nicht kennen, gebt bei Google einfach "Latex ..." ein. Das "..." ersetzt ihr einfach durch die gewünschte Funktionen, wie zum Beispiel "Äufzählung", "kursiv schreiben", "Grafiken einbinden" etc. Ihr solltet dort sehr schnell eine Erklärung und ein Beispiel finden.

Es gibt eine Menge Tutorials online verfügbar - "Hab ich nicht gefunden" ist keine Ausrede!

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)

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

Hilfe zur Bedienung von TeXStudio:

http://texstudio.sourceforge.net/manual/current/usermanual_en.html#SECTION1

TeXnicCenter ist ebenfalls ein Editor. (Plattform: Win)

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

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>.

TortoiseSvn (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, das heißt, dass alle Diagrammtypen unterstützt werden müssen und die Möglichkeit zum Extrahieren von **Vektorgrafiken** zwingend notwendig sein müssen.

Einigt euch dabei vor der Abgabe des ersten Dokuments in der Gruppe auf ein Tool, damit alle Diagramme homogen aussehen und die Dokumente einheitlich bleiben.

- 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 GITZ 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 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/>.