Hochschule Bremerhaven

Fachbereich II Management und Informationssysteme Wirtschaftsinformatik B.Sc.

Modul Qualitätsmanagement

Semesteraufgabe

Entwicklung einer Hausverwaltung

Vorgelegt von: Junior Lesage Ekane Njoh MatNr. 40128

Steve Aguiwo II MatNr. 40088

Franck Majeste Dogmo Silatsa MatNr. 00000

Vorgelegt am: 19. Februar 2025

Dozent:in: Prof. Dr. Karin Vosseberg

Inhaltsverzeichnis

1	Einle	eitung
2	Anfo 2.1 2.2	rderungsanalyse4Review der Anforderungen4Verbesserung der Anforderungen4
3	3.1 3.2 3.3	Auswahl von Testverfahren
4	Entw 4.1 4.2	Ableitung konkreter Testfälle 4 Erstellung einer Testfall-Dokumentation 4
5	Proto 5.1 5.2 5.3	Software-Architektur und Technologien
6	Qual 6.1 6.2	itätsmanagement-Methoden in der Softwareentwicklung
7	Einle	eitung
Lit	teratu	rverzeichnis
Lis	stingv	erzeichnis
An	Ihang I II III IV V	Review der Anforderungen
Se	lbststä	indigkeitserklärung

1 Einleitung

- 2 Anforderungsanalyse
- 2.1 Review der Anforderungen
- 2.2 Verbesserung der Anforderungen
- 3 Testkonzept
- 3.1 Auswahl von Testverfahren
- 3.2 Teststufen und Testarten
- 3.3 Testumgebung und Testdaten
- 4 Entwicklung der Testfälle
- 4.1 Ableitung konkreter Testfälle
- 4.2 Erstellung einer Testfall-Dokumentation
- 5 Prototypische Umsetzung der Hausverwaltung
- 5.1 Software-Architektur und Technologien
- 5.2 Implentierung
- 5.3 Anwendung des Testkonzepts
- 6 Qualitätsmanagement-Methoden in der Softwareentwicklung
- 6.1 Relevanz der Qualitätssicherung
- 6.2 Anwendung von QS-Methoden im Projekt

7 Einleitung

Listingverzeichnis

II.1	linbinden von subfiles	8
II.2	rstellen von subfiles	9
IV.1	inbinden von subfiles	2
IV.2	rstellen von subfiles	2

Anhang

I Review der Anforderungen

Review der Anforderungen

Eine unnötig komplizierte und sehr sehr lange Abhandlung

Untersuchung einer Sache von wirklichem Wert

Maxi Mustermensch

Einleitung

ausgaben. Wer diesen Text Heist, ist sehlt schalde Der Text gibt lediglich den Grauwert der Schrift un. Ist das wirklich so 7 hie en gelebehglitz, den Der Text gibt lediglich den Grauwert der Schrift un. Ist das wirklich so 7 hie en gelebehglitz, den Grauwert der Schrift un. Ist das wirklich so 7 hie en gelebehglitz, der des geharte Schrift un der Schrift un. Bei Buldext bletet mit wichtige Informationen. An him messe ich die Lesbartet in ere Schrift, him et Annutung, wie harmonisch die Figuren zuestander stehe Buchstaben enthalten und in der Originalsprache Buchstaben enthalten und in der Originalsprache Buchstaben enthalten und in der Originalsprache Des bletes in ein Bindext und restehen in Fremdupenschige Texte wie Ausgaben. Wer diesen in Kritt im Texte vom Cratta Lexen in pauri Genera in kein dem eigentlichen Lexen in pauri Genera in kein dem eigentlichen Des bletes ist ein Bindext unm Testen vom Texte unspache. Wer diesen in Kritt im Testen vom Texte und der Verlagen der V

Zielstellung

Dies hier ist ein Blindtext zum Testen von Tex aussgaben. Wer diesen Text liest, ist selbst schul Der Text gibt lediglich den Grawwert der Schri an, 1st das wirklich soo? Ist es gleichgültig, ob ic schreibe: "Dies ist ein Blindtext" oder "Huardei gefbun"? Kjitt – mitnichten! Ein Blindtext bier mir wichtige Informationen. An ihm messe ic die Lesbarkeit einer Schrift, link Annutune, wi harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie "Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Tabelle 1: LiTipX Schriftgrößen		
Befehl	Ergebnis	
\tiny	winig	
\scriptsize	klein	
\footnotesize	etwas größer	
\small	noch etwas größer	
\normalsize	normal	
\large	groß	
\Large	größer	
\LARGE	noch größer	
Vhuge	riesig	
\Huge	noch riesiger	

Dies hier ist ein Blindetex zum Testen von Test ungehen. Wer diesen Erxt ließt, sie sehns schuld Der Text gibt lediglich den Grauwert der Schrift, an Ist das wirklich zoh's it es gleichgültig, ob ich schreiber. Dies ist ein Blindeter oder, Hausselse gebrun"? Kliff- unteinteiert Ein Blindete bietet mit wichtige Informationen. An ihm messe ich ein wichtige Informationen. An ihm messe ich mit wichtige Informationen in ihm wich ihm wir schalt in ihm wir ihm wir ihm wir ihm wir ihm wir schalt in ihm wir schalt in ihm wir ihm

Materialien und noch etwas, damit es zweizeilig

Dies hier ist ein Blindeax zum Testen vom Testungglein. Wer dieser fast lies, ist eines schuld ausgehen. Wer dieser fast lies, ist eines schuld im 1st den werklich vo? It er spielspällig, eicht schriebe. Dies ist ein Blindeart worft. Hausdest gefburn" Rjiff – minnichten Ein Blindeart worft. Hausdest gefburn" Rjiff – minnichten Ein Blindeart bei ein zu der die Leubsteder der Sjörft, in er Jammung, wer um wecktige Informationen. An ihm messe ich die Leubsteder der Sjörft, in er Jammung, wer prifte, wei ber ich er den sich in er Jammung, wer prifte, wei ber ich er der Sjörft, in Ein Blind text sollte möglicht viele verschieden Buchats sein. Er mus keinen Sinn ergeben, sollte aber sein. Er mus keinen Sinn ergeben, sollte aber jumm" denen nicht dem eigentlichen Zweck, da

Hochschule Bremerhaven

sie eine falsche Anmutung vermitteln.

or i in (1..10); do
echo "\$i"
one

Listing 1: Ein Listing

Schluss

Dies hier ist ein Billedoten zum Teisten von Exasungehen. Wer diesen Erst leist, ist oder beschuld. Der Ten gibt lediglich dem Grausvert ers Schrift und. Der Ten gibt lediglich dem Grausvert ers Schrift und sich sein wirde. Dies ist eine Billedurer oder "Hundeste geführun" Right—minischent Ein Billedurer oder "Hundeste mir wichtige Informationer. An ihm messes ich die Leibzskeit einer Schrift, ihre Ammunung wie harmonisch die Figurerz zuseinunder stehen und die Leibzskeit einer Schrift, ihre Ammunung wie harmonisch die Figurerz zuseinunder stehen und seine Schrift wirden der Schrift und der Schrift und der Schrift und der Schrift und seine Schrift und seine Schrift und der Originalsprache gester sein. Er muss keinen Sim ergeben, sollte Sim der Schrift und seine Schrift und

Abbildung 1: Bildunterschrift [1, S.14]

Dies hier ist ein Blindetext zum Testen von Testusagnehn. We diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift, an f. das wirklich oder Sit es gleicheiglie, ob ein geführer. Vijft — mindelune Ein ober Alle und geführer. Vijft — mindelune Ein ober der die Lesbarkeit einer Schrift, ihrer Annattung, wir mir wehötig in fürgern zueinunder stehen und prüfe, wie berit oder schmid sie liste. Ein Blindnatung der schwarzeit werden gestellt der Schrift werden gestellt der Schrift und der Ortginalspruche gestellt sie Ern uns keinen Sinn ergeben, sollest leis eine Truss keinen Sinn ergeben, sollest eine Sinn ergeben, solle eine Sinn ergeben, sollest eine Sinn ergeben ergeben ergeben ergeben ergeben ergeben ergebe

Referenzen

 M. M. Musterfrau Maxi. "MS Window NT Kernel Description." de. (Dez. 2012 Adresse: http://www.808multimedi. com/winnt/kernel.html (besucht ar 30.09.2012).

Abbildung I.1: Poster

II Verbesserte Review der Anforderungen

Wird ein Projekt mit der Zeit relativ groß, verfügt also über eine Vielzahl von Listings, Tabellen, Abbildungen und Text, dann wird für das Kompilieren eine nicht unerhebliche Zeit benötigt. Um das ganze Dokument, inkl. aller Referenzen und Abkürzungen etc. zu bauen, ist dies auch nötig und lässt sich nicht vermeiden. Oft benötigt man jedoch nur einen kurzen Blick auf Layout oder den Umfang des bisher geschriebenen und möchte nicht das Dokument in seiner Gesamtheit kompilieren.

Für diese Anforderungen gibt es sog. subfiles, diese ermöglichen es, nur einen kleinen Teil

des Dokumentes zu kompilieren und als PDF zu betrachten. Dabei kommt es zu gewissen Einschränkungen, so werden Seitenzahlen, Referenzen und Abkürzungen nicht korrekt dargestellt, dennoch erhält man einen guten Blick auf das Layout. Um mit subfiles zu arbeiten, muss einiges beachtet werden. So muss statt des \input{} das Kommando \subfile{} genutzt werden. Darüber hinaus müssen alle .tex-Dateien mit einer eigenen documentclass ausgestattet werden. In der hauptdatei.tex sieht dies dann folgendermaßen aus.

```
\defverbatim[colored]\CodeOne{
1
      \begin{minted}{python}
2
        import numpy as np
        import pylab as pl
        def f_x(x):
          return np.exp(x)+x**2-5*x
        def approx_f(x):
          return 1 -4*x +3./2*x**2
10
11
        xvals = np.arange(-4,4,0.1)
12
        fx_vals = [f_x(x) for x in xvals]
13
        approx_vals = [approx_f(x) for x in xvals]
14
15
        pl.plot(xvals,fx_vals)
        pl.plot(xvals,approx_vals)
18
        pl.show()
19
    \end{minted}
20
    \captionof{listing}{Example Code}
21
22
23
    \section{Code Example}
24
    \begin{frame}{Code Example}
25
      Some Text on first Frame~\autocite{gregg2013systems}
      \CodeOne
27
      \vspace{0.5cm}
28
    \end{frame}
29
```

Listing II.1: Einbinden von subfiles

Jedes Subfile muss dann mit der passenden documentclass beginnen.

```
\defverbatim[colored] \CodeOne{
1
      \begin{minted}{python}
2
         import numpy as np
3
         import pylab as pl
         def f_x(x):
           return np.exp(x)+x**2-5*x
         def approx_f(x):
           return 1 -4*x +3./2*x**2
10
11
         xvals = np.arange(-4,4,0.1)
12
         fx_vals = [f_x(x) \text{ for } x \text{ in } xvals]
13
         approx_vals = [approx_f(x) for x in xvals]
14
15
         pl.plot(xvals,fx_vals)
16
         pl.plot(xvals,approx_vals)
17
18
         pl.show()
19
    \end{minted}
20
    \captionof{listing}{Example Code}
21
    }
22
23
    \section{Code Example}
24
    \begin{frame}{Code Example}
25
      Some Text on first Frame~\autocite{gregg2013systems}
26
      \CodeOne
27
      \vspace{0.5cm}
28
    \end{frame}
29
```

Listing II.2: Erstellen von subfiles

Durch die eigene documentclass, welche wiederum auf die hauptdatei.tex verweist und die Angabe von \begin bzw. \end{document} kann die entsprechende tex-Datei separat kompiliert werden. Dafür kann das Skript quickbild.sh genutzt werden, welches als Argument den Namen der zu kompilierenden Datei erwartet und ein PDF erzeugt.

Abbildungen, Listings, Tabellen können wie gewohnt genutzt werden. Wird buildlualatex.sh ausgeführt, baut nach wie vor das gesamte Dokument inkl. aller Subfiles und Referenzen.

III Testfälle

Wie im Listing gezeigt, wird das Listing außerhalb des Frames definiert und in Zeile 27 eingefügt. CodeOne ist dabei der zugewiesene Name unter dem das Listing eingefügt werden kann. Die Definition erfolgt in Zeile 1, der Name ist frei wählbar. Das Listing so einzubinden ist nicht optimal, jedoch bleibt aktuell keine andere Möglichkeit, wenn mit minted gearbeitet wird.

Das Inhaltsverzeichnis und die Referenzen werden automatisch erstellt, alle Einstellungen können natürlich innerhalb der Vorlage angepasst werden.

Obiges Listing würde mit der Vorlage folgendermaßen aussehen. Wie immer finden sich alle Beispiele auch in den jeweiligen Vorlagen.

```
Code Example
            Some Text on first Frame [1]
             import numpy as np
import pylab as pl
             def f_x(x):
              return np.exp(x)+x**2-5*x
            def approx_f(x):
              return 1 -4*x +3./2*x**2
            xvals = np.arange(-4,4,0.1)
fx_vals = [f_x(x) for x in xvals]
       10
             approx_vals = [approx_f(x) for x in xvals]
             pl.plot(xvals,fx_vals)
             pl.plot(xvals,approx_vals)
             pl.show()
                                                                                                                        Hochschule
Bremerhaven
                                                  Listing: Example Code
Maxi Musterfrau (12345), Maxi Mustermann (54321)
```

Abbildung III.1: Frame mit Listing

IV Quickbuild

Wird ein Projekt mit der Zeit relativ groß, verfügt also über eine Vielzahl von Listings, Tabellen, Abbildungen und Text, dann wird für das Kompilieren eine nicht unerhebliche Zeit benötigt. Um das ganze Dokument, inkl. aller Referenzen und Abkürzungen etc. zu bauen, ist dies auch nötig und lässt sich nicht vermeiden. Oft benötigt man jedoch nur einen kurzen Blick auf Layout oder den Umfang des bisher geschriebenen und möchte nicht das Dokument in seiner Gesamtheit kompilieren.

Für diese Anforderungen gibt es sog. subfiles, diese ermöglichen es, nur einen kleinen Teil des Dokumentes zu kompilieren und als PDF zu betrachten. Dabei kommt es zu gewissen Einschränkungen, so werden Seitenzahlen, Referenzen und Abkürzungen nicht korrekt dargestellt, dennoch erhält man einen guten Blick auf das Layout. Um mit subfiles zu arbeiten, muss einiges beachtet werden. So muss statt des \input{} das Kommando \subfile{} genutzt werden. Darüber hinaus müssen alle .tex-Dateien mit einer eigenen documentclass ausgestattet werden. In der hauptdatei.tex sieht dies dann folgendermaßen aus.

```
\defverbatim[colored]\CodeOne{
1
      \begin{minted}{python}
2
        import numpy as np
3
        import pylab as pl
4
        def f_x(x):
          return np.exp(x)+x**2-5*x
        def approx_f(x):
          return 1 -4*x +3./2*x**2
10
11
        xvals = np.arange(-4,4,0.1)
12
        fx_vals = [f_x(x) for x in xvals]
13
        approx_vals = [approx_f(x) for x in xvals]
14
15
        pl.plot(xvals,fx_vals)
16
        pl.plot(xvals,approx_vals)
17
18
        pl.show()
19
    \end{minted}
20
    \captionof{listing}{Example Code}
21
22
23
    \section{Code Example}
24
    \begin{frame}{Code Example}
```

```
Some Text on first Frame~\autocite{gregg2013systems}

CodeOne
vspace{0.5cm}

end{frame}
```

Listing IV.1: Einbinden von subfiles

Jedes Subfile muss dann mit der passenden documentclass beginnen.

```
\defverbatim[colored]\CodeOne{
      \begin{minted}{python}
        import numpy as np
        import pylab as pl
        def f_x(x):
6
          return np.exp(x)+x**2-5*x
        def approx_f(x):
          return 1 -4*x +3./2*x**2
10
        xvals = np.arange(-4,4,0.1)
12
        fx_vals = [f_x(x) for x in xvals]
13
        approx_vals = [approx_f(x) for x in xvals]
14
15
        pl.plot(xvals,fx_vals)
16
        pl.plot(xvals,approx_vals)
17
18
        pl.show()
19
    \end{minted}
20
    \captionof{listing}{Example Code}
21
22
23
    \section{Code Example}
24
    \begin{frame}{Code Example}
25
      Some Text on first Frame~\autocite{gregg2013systems}
26
      \CodeOne
27
      \vspace{0.5cm}
28
    \end{frame}
```

Listing IV.2: Erstellen von subfiles

Durch die eigene documentclass, welche wiederum auf die hauptdatei.tex verweist und die Angabe von \begin bzw. \end{document} kann die entsprechende tex-Datei separat kompiliert werden. Dafür kann das Skript quickbild.sh genutzt werden, welches als Argument den Namen der zu kompilierenden Datei erwartet und ein PDF erzeugt.

Abbildungen, Listings, Tabellen können wie gewohnt genutzt werden. Wird buildlualatex.sh ausgeführt, baut nach wie vor das gesamte Dokument inkl. aller Subfiles und Referenzen.

V Prototyp-Dokumentation

Passend zu diesem Template existieren noch zwei weitere Vorlagen für Poster. Diese unterscheiden sich lediglich durch die Ausrichtung der Seite und sind ansonsten identisch. Der Aufbau der Dateien und Projektstruktur ist weitestgehend identisch zu dieser Vorlage, es gilt lediglich auf einige kleine Besonderheiten zu verweisen.

In der sog. multicols Umgebung ist es nicht möglich, mit Floatumgebungen zu arbeiten, dies ist für fast alle Beispiele aus dieser Anleitung irrelevant, lediglich Grafiken müssen anders eingebunden werden. Dazu finden sich jedoch Beispiele in den Vorlagen. Ebenso ist ein Teil der Präambel in eine separate Datei ausgelagert, in dieser sollten auch nicht ohne weiteres Änderungen vorgenommen werden. Wie in diesem Template, müssen innerhalb der Präambel Zitierstil, sowie Autor:innen und Titel gesetzt werden.

Die Vorlagen können auf meiner Webseite sowohl angeschaut, als auch heruntergeladen werden, alle Hinweise zum Kompilieren gelten für die Poster ebenso. Ergebnis dieser Vorlage ist z. B. folgendes Poster:

Selbstständigkeitserklärung

Wir versichern, die von uns vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die wir für die Arbeit benutzt haben, sind angegeben. Die Arbeit haben wir mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegt.

Bremerhaven, den 19. Februar 2025 Unterschrift: Junior Leage EKane Njoh, Franck Majeste Silatsa Dogmo, Steve Aguiwo II