

ACE

a collaborative editor

Projekthandbuch

Berner Fachhochschule
Hochschule für Technik und Informatik

Projektarbeit im Modul Projektführung

Ausgabe vom: 06.04.2005

Version: 0.7

Projektteam: Mark Bigler (biglm2@hta-bi.bfh.ch)
Simon Räss (rasss@hta-bi.bfh.ch)
Lukas Zbinden (zbinl@hta-bi.bfh.ch)

Empfänger: Frank Helbling (frank.helbling@helbling-consulting.ch)
Jean-Paul Dubois (doj@hta-bi.bfh.ch)
Claude Fuhrer (frc@hta-bi.bfh.ch)

Ablage: Subversion Repository

Inhaltsverzeichnis

1	Einleitung	4
1.1	Zweck des Dokuments	4
2	Projektbeschreibung	4
2.1	Ausgangssituation	4
2.2	Ziele	4
2.3	Nicht-Ziele	5
2.4	Vorgehensstrategie	5
3	Projektspezifisches Vorgehensmodell	5
3.1	Phase Initialisierung	6
3.2	Phase Analyse und Entwicklung	6
3.2.1	Komponente Algorithmus	6
3.2.2	Komponente Netzwerk	6
3.2.3	Komponente GUI	6
3.3	Phase Abschluss	6
4	Entscheidungspunkte und auszuliefernde Ergebnisse	7
4.1	Initialisierung	7
4.2	Synchronisationsalgorithmus	7
4.3	Analyse Netzwerk	7
4.4	Analyse GUI	8
4.5	Abschluss	8
5	Methoden und Werkzeuge	8
5.1	Dokumente	8
5.2	Source Repository	8
5.3	Projektwebsite	8
5.4	Zeiterfassung	8
5.5	Trac	9
5.6	Kalender	9
5.7	Entwicklung	9
6	Standards und Richtlinien	9
6.1	Dokumentation	9
6.2	Quellcode	9
A	Anhang: Ergänzende Projektvereinbarungen	11
A.1	Projektorganisation	11
A.2	Projektplanung	11
A.3	Qualitätssicherung	12
A.4	Konfigurationsmanagement	12

Tabellenverzeichnis

1	Versionskontrolle	3
2	Prüfung/Genehmigung	3
3	Ziele	4
4	Nicht-Ziele	5
5	Projektorganisation	11
6	Projektplanung	12
7	Versionsnummern	12

Abbildungsverzeichnis

1	Projektspezifisches Vorgehensmodell	5
2	Projektorganisation	11

Versionskontrolle

Version	Datum	Verantwortlich	Bemerkungen
0.1	15.03.2005	rasss	Erste Version
0.2	22.03.2005	rasss zbinl	Überarbeitung
0.3	30.03.2005	zbinl	Projektspezifisches Vorgehensmodell
0.4	30.03.2005	rasss zbinl	Überarbeitung Punkte 2, 3 und 4
0.5	05.04.2005	zbinl	Überarbeitung Appendix A.1
0.6	06.04.2005	zbinl	Anpassungen Punkte 2, 3 und 4
0.7	06.04.2005	Projektteam	Review

Tabelle 1: Versionskontrolle

	Stelle	Datum	Visum	Bemerkungen
Freigegeben	Projektteam			
Genehmigt				

Tabelle 2: Prüfung/Genehmigung

1 Einleitung

1.1 Zweck des Dokuments

Das Projekthandbuch dient als einheitliche Handlungsgrundlage für alle Projektbeteiligten und legt damit den allgemeingültigen technischen und organisatorischen Rahmen fest.

Dieses Dokument ist soweit wie möglich als statisches Dokument zu führen. Es ist jedoch zu Beginn und am Schluss jeder Phase zu überprüfen und an die neuen Erkenntnisse anzupassen.

2 Projektbeschreibung

Im Projekt *ACE* soll ein kollaborativer, plattformunabhängiger Editor entwickelt werden. Diese Applikation ermöglicht mehreren Personen, ein Textdokument gemeinsam zu bearbeiten. Dabei arbeitet jede Person mit dem Editor an einem eigenen Computer. Alle Teilnehmer sind über ein Netzwerk verbunden und sehen jederzeit den gleichen Dokumentinhalt. Wenn jemand der Gruppe eine Änderung im Dokument vornimmt, wird dies in Echtzeit und synchron allen anderen Benutzern angezeigt. Jeder Benutzer hat dadurch den Überblick über alle Änderungen im Dokument. Dieser Editor ermöglicht zum Beispiel ein gemeinsames Brainstorming von mehreren Personen, welche sich an verschiedenen Orten befinden.

2.1 Ausgangssituation

Heutzutage besitzt praktisch jeder Mensch in den Industrieländern einen Computer oder hat einen Zugang dazu. Verfassen von Texten, E-Mails und das Verwenden des World Wide Web gehören für die meisten Menschen zum Alltag. Doch gerade im Bereich Zusammenarbeit erleichtern uns Computer die Arbeit oft nicht. Im Gegenteil, oft verhindern Computer ein intuitives Zusammenarbeiten. Allgemein sind Experten der Überzeugung, dass Anwendungen, welche die computerunterstützte Zusammenarbeit ermöglichen, ein grosses Potential haben.

2.2 Ziele

In der Semesterarbeit soll die Basis für die Implementation eines kollaborativen Texteditors im Rahmen der Diplomarbeit gelegt werden.

Ziele	Prioritäten	Beschreibung
1	1	Aufbau von Know-How im Bereich CSCW
2	1	Evaluation bestehender Algorithmen
3	1	Implementation Algorithmus
4	1	Testframework für Algorithmus
5	2	Analyse GUI
6	2	Analyse Netzwerk/Kommunikation

Tabelle 3: Ziele

2.3 Nicht-Ziele

Beschreibung
Ausarbeiten von Sicherheitsaspekten Prototyp kollaborativer Texteditor

Tabelle 4: Nicht-Ziele

2.4 Vorgehensstrategie

In dem Semesterprojekt geht es vor allem darum, Wissen aufzubauen und dazu einen funktionierenden Synchronisationsalgorithmus zu entwickeln. Das Vorgehensmodell ist im Abschnitt 3 im Detail erklärt.

3 Projektspezifisches Vorgehensmodell

Das Vorgehen wird durch drei Hauptphasen bestimmt (siehe Abbildung 1). Die erste Phase initialisiert das Projekt. Die zweite Phase ist in drei Komponenten unterteilt, welche parallel bearbeitet werden können. Die einzelnen Komponenten stellen ein Subprojekt mit eigenen Phasen dar. Die dritte und letzte Phase beinhaltet den Abschluss des Projektes. Die Abbildung 1 stellt eine logische Unterteilung der Vorgehensweise dar.

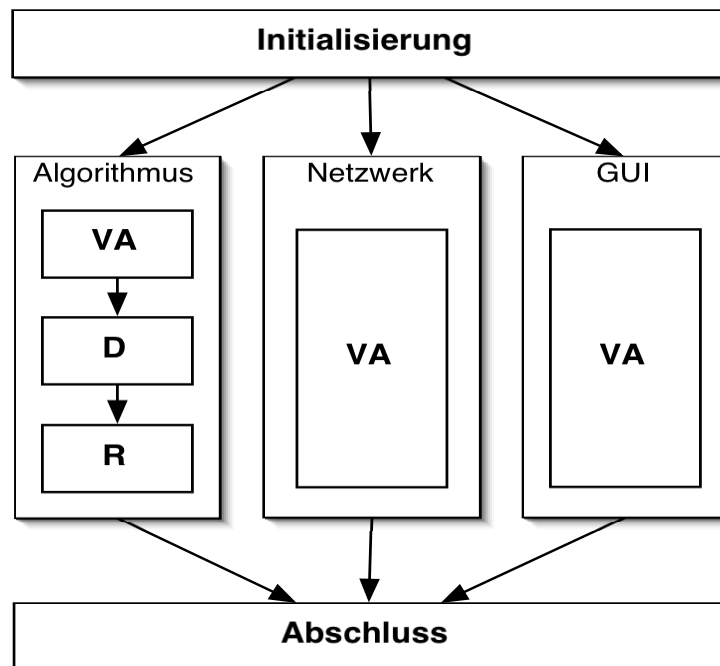


Abbildung 1: Projektspezifisches Vorgehensmodell

Nachfolgend werden die drei Hauptphasen detaillierter erklärt.

3.1 Phase Initialisierung

In der Phase Initialisierung geht es vor allem darum, einen klar definierten organisatorischen und technischen Rahmen als Voraussetzung für eine erfolgreiche Projektabwicklung zu schaffen.

3.2 Phase Analyse und Entwicklung

Die zweite Hauptphase wird in drei unabhängige Komponenten Algorithmus, Netzwerk und GUI unterteilt. Es bestehen keine Schnittstellen zwischen diesen Komponenten.

3.2.1 Komponente Algorithmus

Das ist die Hauptkomponente und der zeitintensivste Teil der Projektarbeit. Es beinhaltet die Entwicklung des Synchronisationsalgorithmus sowie das Erstellen eines Testframeworks. Die beiden Teile werden parallel bearbeitet. Die Komponente Algorithmus ist in drei Phasen aufgeteilt:

Phase Voranalyse: In der Phase Voranalyse wird ein fundiertes Basiswissen im Bereich des CSCW (*Computer Supported Cooperative Work*) aufgebaut. Mit diesem Know-How soll anschliessend eine Analyse der aktuellen Synchronisationsalgorithmen gemacht werden. Diese Studie ermöglicht die Auswahl eines geeigneten Synchronisationsalgorithmus für die Implementation.

Phase Design: In der Phase Design soll zuerst das Wissen über den gewählten Synchronisationsalgorithmus vertieft werden. Anschliessend erfolgt das Systemdesign für den Algorithmus, welches die Grundlage für eine erfolgreiche Realisationphase bildet. Das Design für das Testframework wird ebenfalls erstellt.

Phase Realisation: In der Phase Realisation werden der Synchronisationsalgorithmus und das Testframework implementiert. Sobald die beiden Teile entwickelt sind, wird der Algorithmus mit dem Testframework geprüft.

3.2.2 Komponente Netzwerk

Voranalyse: Es sollen mögliche Lösungsvorschläge für die Komponente Netzwerk erarbeitet werden. Die Lösungsvorschläge müssen untersucht und auf ihre Tauglichkeit bewertet werden. Es soll ein Lösungsvorschlag für die spätere Implementation ausgewählt werden.

3.2.3 Komponente GUI

Voranalyse: Es sollen Lösungsvarianten für die Komponente GUI evaluiert werden. Ein Lösungsvorschlag wird für die spätere Implementation ausgewählt.

3.3 Phase Abschluss

In der Phase Abschluss wird das Projekt geordnet zu Ende gebracht. Alle Projektaktivitäten werden abgeschlossen. Das Team verfasst einen Erfahrungsbericht.

4 Entscheidungspunkte und auszuliefernde Ergebnisse

4.1 Initialisierung

Ergebnisse

- Projektantrag
- Projekthandbuch
- Pflichtenheft
- Projektplan

Entscheidungspunkte

- Freigabe der nächsten Phase

4.2 Synchronisationsalgorithmus

Ergebnisse

- Bericht Evaluation und Auswahl Synchronisationsalgorithmen
- Implementation Synchronisationsalgorithmus
- Bericht Implementation Synchronisationsalgorithmus
- Sammlung von Testfällen
- Testframework Synchronisationsalgorithmen
- Bericht Testframework

Entscheidungspunkte

- Auswahl Synchronisationsalgorithmus basierend auf Evaluation
- Freigabe Komponente Algorithmus

4.3 Analyse Netzwerk

Ergebnisse

- Bericht Analyse Netzwerk (Evaluation Lösungen, Auswahl Lösung, Erfahrungen)

Entscheidungspunkte

- Auswahl Lösung

4.4 Analyse GUI

Ergebnisse

- Bericht Analyse GUI (Evaluation Lösungen, Auswahl Lösung, Erfahrungen)

Entscheidungspunkte

- Auswahl Lösung

4.5 Abschluss

Ergebnisse

- Erfahrungsbericht
- CD mit allen Ergebnissen

Entscheidungspunkte

- Projektabschluss

5 Methoden und Werkzeuge

5.1 Dokumente

Zum Erstellen aller Dokumente soll grundsätzlich \LaTeX verwendet werden. Selbstverständlich werden für Grafiken und Diagramme andere Anwendungen verwendet. Da es sich bei \LaTeX Dateien um Textdateien handelt, kann SubEthaEdit (ein kollaborativer Editor für Mac OS X) verwendet werden, um zur gleichen Zeit am gleichen Dokument zu arbeiten.

5.2 Source Repository

Subversion wird als Source Repository verwendet. Die URL für den Subversion-Zugriff ist <http://ace.iserver.ch:81/repos/ace/ace>.

5.3 Projektwebsite

Von der Projektwebsite (<http://ace.iserver.ch>) aus erreicht man alle wichtigen Werkzeuge. Sie dient auch dazu, das Produkt *ACE* zu vermarkten.

5.4 Zeiterfassung

Zur Erfassung der Arbeitszeit wird eine PHP Applikation, entwickelt von Mark Bigler, verwendet. Dies wird es ermöglichen, am Ende des Projektes Angaben zu den geleisteten Arbeitsstunden zu machen.

5.5 Trac

Trac ist ein sogenannter *issue tracker* mit einem integrierten Wiki. Das Wiki wird für den projektinternen Informationsaustausch verwendet (<http://ace.iserver.ch:81/cgi-bin/trac.cgi>). Der *issue tracker* wird für die gezielte Erfassung von Bug Reports verwendet.

5.6 Kalender

Ein online verfügbarer Projektkalender gibt Auskunft über alle anstehenden Sitzungen und andere Termine (<http://ace.iserver.ch/phpicalendar/week.php>).

5.7 Entwicklung

- Eclipse (<http://www.eclipse.org/>)
- SubEthaEdit (<http://www.codingmonkeys.de/subethaedit/>)
- ant (<http://ant.apache.org/>)
- CruiseControl (<http://cruisecontrol.sourceforge.net/>)

6 Standards und Richtlinien

6.1 Dokumentation

Alle \LaTeX Dokumente sollen die Datei `ace.tex` einbinden. Diese Datei findet man im Subversion Repository unter `/ace/trunk/doc/latex/ace.tex`. Eine Vorlage für alle \LaTeX Dokumente findet man im selben Verzeichnis (`template.tex`). Genaue Angaben zum Erstellen von Dokumenten findet man online im Wiki unter `LatexDocuments`.

6.2 Quellcode

Die Dokumentation des Quellcodes erfolgt mit JavaDoc. Die Kommentare werden einheitlich in Englisch verfasst. Quellcodedateien werden mit folgendem Header, welcher unter `/ace/trunk/doc/templates/source.header` im Subversion Repository zu finden ist, versehen.

```
/*
 * $Id$
 *
 * ace - a collaborative editor
 * Copyright (C) 2005 Mark Bigler, Simon Raess, Lukas Zbinden
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation; either version 2
 * of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
```



```
* GNU General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License  
* along with this program; if not, write to the Free Software  
* Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.  
*/
```

Konventionen

Die Quellcodedateien sollen gemäss den Sun Coding Conventions (<http://java.sun.com/docs/codeconv/>) erstellt werden. Die Einrückung erfolgt mit Tabulator Zeichen (nicht mit Spaces).

A Anhang: Ergänzende Projektvereinbarungen

A.1 Projektorganisation

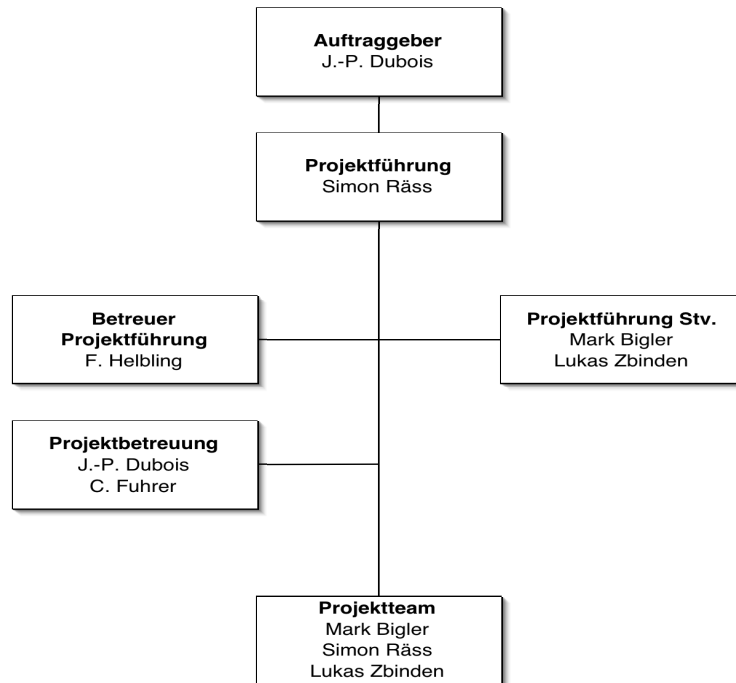


Abbildung 2: Projektorganisation

Auftraggeber	J.-P. Dubois	doj@hta-bi.bfh.ch +41 32 321 62 82
Projektleiter	S. Räss	rasss@hta-bi.bfh.ch
Projektteam	M. Bigler S. Räss L. Zbinden	biglm2@hta-bi.bfh.ch rasss@hta-bi.bfh.ch zbinl@hta-bi.bfh.ch
Projektbetreuer	J.-P. Dubois C. Fuhrer	doj@hta-bi.bfh.ch frc@hta-bi.bfh.ch
PM Coach	F. Helbling	frank.helbling@helbling-consulting.ch

Tabelle 5: Projektorganisation

A.2 Projektplanung

Die genaue Projektplanung kann dem Projektplan entnommen werden.

Anfang der Semesterarbeit	21.02.2005
Beginn der Initialisierungsphase	21.02.2005
Ende der Semesterarbeit	24.06.2005

Tabelle 6: Projektplanung

A.3 Qualitätssicherung

Für alle kritischen Stellen der Implementation müssen Unit Tests bestehen. Für das Testen des Synchronisationsalgorithmus soll ein Testframework implementiert werden. Dieses soll es ermöglichen auch komplexe zeitliche Abfolgen zu definieren und wiederholt ablaufen zu lassen. Damit können präzise Testfälle erstellt werden, die automatisiert und wiederholt ausgeführt werden können.

A.4 Konfigurationsmanagement

Es werden sämtliche Dokumente des Projektmanagements, sämtlicher Quellcode, die Inhalte der Website und die für die Erstellung des Programms benötigten externen Bibliotheken unter das Konfigurationsmanagement gestellt.

Die Projektdokumentation und Produktdokumentationen werden mit einer Versionskontrolle versehen. Die Dokumente können dabei die in der Tabelle 7 angegebenen Zustände annehmen.

Version	Bezeichnung
0.1 - 0.99	Entwurfsversionen
1.0	erste Release Version
1.1	allfällige überarbeitete Versionen

Tabelle 7: Versionsnummern

Für das Konfigurationsmanagement wird das Subversion Repository verwendet. Das entsprechende Repository ist unter <http://ace.iserver.ch:81/repos/ace/> erreichbar. Aus der *commit*-Nachricht soll erkenntlich sein, wann die Versionsnummer des Dokumentes erhöht wurde.