

ACE

a collaborative editor

Projekthandbuch

Mark Bigler
Simon Räss
Lukas Zbinden

Hochschule für Technik und Informatik
Abteilung Informatik



Inhaltsverzeichnis

1	Einleitung	4
1.1	Zweck des Dokuments	4
2	Projektbeschreibung	4
2.1	Ausgangssituation	4
2.2	Ziele	4
2.3	Vorgehensstrategie	4
3	Projektspezifisches Vorgehensmodell	5
3.1	Phase Initialisierung	5
3.2	Phase Analyse und Entwicklung	5
3.2.1	Komponente Algorithmus	6
3.2.2	Komponente Netzwerk	6
3.2.3	Komponente GUI	6
3.2.4	Komponente User Stories	6
3.3	Phase Abschluss	6
4	Entscheidungspunkte und auszuliefernde Ergebnisse	6
4.1	Initialisierung	6
4.2	Synchronisationsalgorithmus	7
4.3	Analyse Netzwerk	7
4.4	Analyse GUI	7
4.5	User Stories	7
4.6	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	8
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	10
A.1	Projektorganisation	10
A.2	Projektplanung	10
A.3	Qualitätssicherung	11
A.4	Konfigurationsmanagement	11



Tabellenverzeichnis

1	Versionskontrolle	3
2	Prüfung/Genehmigung	3
3	Ziele	4
4	Projektorganisation	10
5	Projektplanung	11
6	Versionsnummern	11

Abbildungsverzeichnis

1	Projektspezifisches Vorgehensmodell	5
2	Projektorganisation	10



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

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

Das gemeinsame Editieren von Dokumenten kann eine grosse Herausforderung sein. Versionierungssysteme wie Subversion und CVS können einer Gruppe helfen, ein Dokument gemeinsam zu bearbeiten. Diese Systeme ermöglichen aber nicht ein gemeinsames Bearbeiten in Echtzeit. Genau an dieser Stelle setzt das Projekt *ACE* an. Es soll ein Texteditor implementiert werden, der das gleichzeitige Bearbeiten des gleichen Dokumentes in Echtzeit ermöglicht.

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 sie ein intuitives Zusammenarbeiten. Allgemein ist man der Überzeugung, dass Anwendungen, welche die Zusammenarbeit fördern und erleichtern, ein grosses Potential haben. Insbesondere in einer Zeit, wo SMS, Instant Messaging, Mobil- und Internet-Telefonie zum Alltag gehören.

2.2 Ziele

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

Ziele	Prioritäten	Beschreibung
1	1	Aufbau von Knowhow 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
7	3	User Stories für kollaborativen Texteditor

Tabelle 3: Ziele

2.3 Vorgehensstrategie

In dem Semesterprojekt geht es vor allem darum, Wissen aufzubauen und dazu einen funktionierenden Synchronisationsalgorithmus zu entwickeln. Die Ziele sind teilweise unabhängig

voneinander. 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 vier parallel zu bearbeitende Komponenten unterteilt. Die einzelnen Komponenten stellen wiederum ein Subprojekt mit eigenen Phasen dar. Die dritte und letzte Phase beinhaltet den Abschluss des Projektes.

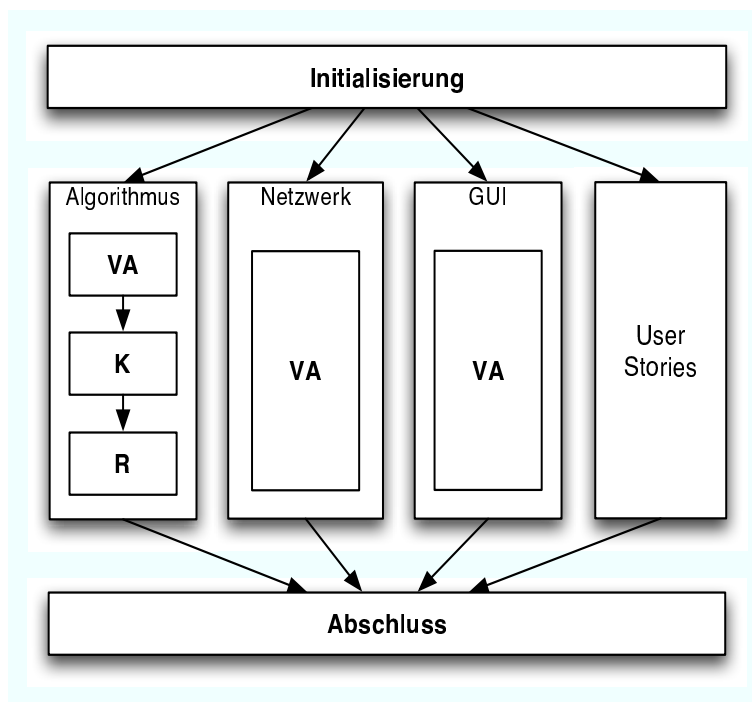


Abbildung 1: Projektspezifisches Vorgehensmodell

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 Phase wird in vier Komponenten unterteilt. Die Komponenten sind in der Bearbeitung unabhängig voneinander und können parallel bearbeitet werden.



3.2.1 Komponente Algorithmus

Dies ist die Hauptkomponente und der zeitintensivste Teil der Projektarbeit. Sie 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 Synchronisationsalgorithmus für die Implementation.

Phase Konzept: In der Phase Konzept soll ein fundiertes Verständnis des gewählten Synchronisationsalgorithmus erarbeitet werden. Das Verständnis ist die Grundlage für eine erfolgreiche Phase Realisation.

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 ausgewählt werden für die spätere Implementation.

3.2.3 Komponente GUI

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

3.2.4 Komponente User Stories

Es soll ein Katalog mit User Stories erstellt werden. Dieser Katalog dient bei der späteren Implementation als Grundlage für die zu implementierende Funktionalität.

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 nächste Phase

4.2 Synchronisationsalgorithmus

Ergebnisse

- Bericht Synchronisationsalgorithmen
- Implementation Synchronisationsalgorithmus
- Erfassen und Erstellen von Testfällen
- Testframework Synchronisationsalgorithmen

Entscheidungspunkte

- Auswahl Synchronisationsalgorithmus basierend auf Bericht

4.3 Analyse Netzwerk

Ergebnisse

- Bericht (mögliche Lösungen, gewählte Lösung, Erfahrungen)

Entscheidungspunkte

- Auswahl Lösung

4.4 Analyse GUI

Ergebnisse

- Bericht (mögliche Lösungen, gewählte Lösung, Erfahrungen)

Entscheidungspunkte

- Auswahl Lösung

4.5 User Stories

Ergebnisse

- User Stories Katalog



Entscheidungspunkte

- keine

4.6 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, können wir SubEthaEdit (ein kollaborativer Editor für Mac) verwenden, 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 aus erreicht man alle wichtigen Werkzeuge. Die Projektwebsite ist erreichbar unter <http://ace.iserver.ch>. Die Projektwebsite dient auch dazu, das Produkt *ACE* zu vermarkten.

5.4 Zeiterfassung

Zur Erfassung der Arbeitszeit wird eine PHP Applikation, entwickelt von Herrn Bigler, verwendet. Dies wird es ermöglichen, am Ende des Projektes einige 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. 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/>)

6 Standards und Richtlinien

6.1 Dokumentation

Alle L^AT_EX Dokumente sollen die Package `ace.tex` verwenden. Diese Datei findet man im Subversion Repository unter `/ace/trunk/doc/latex/ace.tex`. Eine Vorlage für alle L^AT_EX 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

Projektteam	M. Bigler (biglm2@hta-bi.bfh.ch) S. Räss (rasss@hta-bi.bfh.ch) L. Zbinden (zbinl@hta-bi.bfh.ch)
Projektleiter	S. Räss (rasss@hta-bi.bfh.ch)
Projektbetreuer	J.-P. Dubois (doj@hta-bi.bfh.ch) C. Fuhrer (frc@hta-bi.bfh.ch)
PM Couch	F. Helbling (frank.helbling@helbling-consulting.ch)

Tabelle 4: Projektorganisation

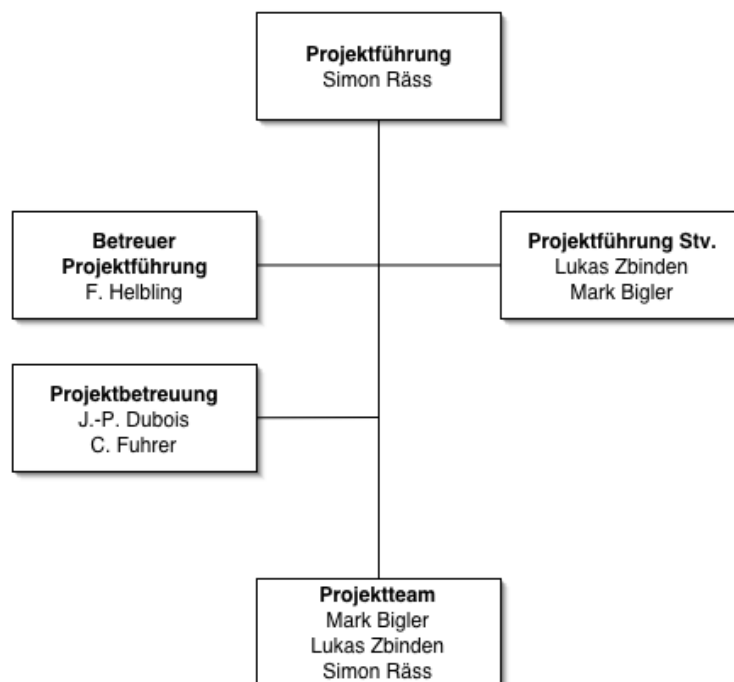


Abbildung 2: Projektorganisation

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	24.07.2005

Tabelle 5: 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 werden unter das Konfigurationsmanagement gestellt.

Die Projektdokumentation und Produktdokumentationen werden mit einer Versionskontrolle versehen. Die Dokumente können dabei die in der Tabelle 6 angegebenen Zustände annehmen.

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

Tabelle 6: Versionsnummern

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