



Inhaltsverzeichnis

1	Einleitung	3
1.1	Zweck des Dokuments	3
2	Projektbeschreibung	3
2.1	Ausgangssituation	3
2.2	Ziele	3
2.3	Vorgehensstrategie	4
3	Projektspezifisches Vorgehensmodell	4
3.1	Phase Initialisierung	4
3.2	Phase Voranalyse	4
3.3	Phase Konzept	4
4	Entscheidungspunkte und auszuliefernde Ergebnisse	5
4.1	Initialisierung	5
4.2	Synchronisationsengine	5
4.3	Konzept GUI/Netzwerk	5
4.4	User Stories	5
5	Methoden und Werkzeuge	6
5.1	Dokumente	6
5.2	Source Repository	6
5.3	Projektwebsite	6
5.4	Zeiterfassung	6
5.5	Trac	6
5.6	Kalender	6
5.7	Entwicklung	6
6	Standards und Richtlinien	7
6.1	Dokumentation	7
6.2	Quellcode	7
6.2.1	Konventionen	7
A	Anhang: Ergänzende Projektvereinbarungen	7
A.1	Projektorganisation	7
A.2	Projektplanung	7
A.3	Qualitätssicherung	7
A.4	Konfigurationsmanagement	8
A.5	Sicherheit	9



Tabellenverzeichnis

1	Versionskontrolle	3
2	Prüfung/Genehmigung	3
3	Ziele	4
4	Projektorganisation	8
5	Projektplanung	8
6	Versionsnummern	9

Abbildungsverzeichnis

1	Projektorganisation	8
---	-------------------------------	---



Versionskontrolle

Version	Datum	Verantwortlich	Bemerkungen
0.1	15.03.2005	rasss	Erste Version

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.



Ziel	Prioritäten	Beschreibung
1	1	Aufbau von Knowhow im Bereich CSCW
2	1	Bestehende Algorithmen evaluieren (Vor-/Nachteile)
3	1	Algorithmus implementieren (Synchronisationsengine)
4	1	Testframework für Algorithmus (Testfälle)
5	2	Konzept GUI
6	2	Konzept 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 eine funktionierende Synchronisationsengine zu entwickeln. Die Ziele sind dabei zum Teil unabhängig voneinander. Grundsätzlich werden pro unabhängigem Teil die Phasen Voranalyse und Konzept durchlaufen. Beim Teilziel Synchronisationsengine werden dabei zusätzlich noch eine Implementierung und damit verbunden Tests erstellt.

3 Projektspezifisches Vorgehensmodell

3.1 Phase Initialisierung

In der Phase Initialisierung werden folgende Ergebnisse erstellt:

- Projektantrag
- Projekthandbuch
- Pflichtenheft
- Projektplan

3.2 Phase Voranalyse

In der Phase Voranalyse werden jeweils die folgenden Ergebnisse erstellt:

- Bericht (mögliche Lösungen)

3.3 Phase Konzept

In der Phase Konzept werden jeweils die folgenden Ergebnisse erstellt:

- detaillierter Bericht Konzept
- Prototyp (kritische Teilsysteme untersuchen)



4 Entscheidungspunkte und auszuliefernde Ergebnisse

4.1 Initialisierung

In der Phase Initialisierung werden folgende Ergebnisse erstellt:

- Projektantrag
- Projekthandbuch
- Pflichtenheft
- Projektplan

4.2 Synchronisationsengine

Für die Synchronisationsengine werden folgende Ergebnisse bereitgestellt:

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

Folgende Entscheidungspunkte treten dabei auf:

- Auswahl Synchronisationsalgorithmus basierend auf Bericht
- Abschluss Implementation

4.3 Konzept GUI/Netzwerk

Für die beiden Konzepte GUI und Netzwerk werden folgende Ergebnisse erstellt:

- ein Bericht (mögliche Lösungen, gewählte Lösung, Erfahrungen)
- Prototyp

Folgende Entscheidungspunkte treten dabei auf:

- Auswahl Lösung

4.4 User Stories

Für die Erfassung des Anwendungsumfangs sollen User Stories erfasst werden. Daraus resultiert folgendes Ergebnis:

- User Stories Katalog

Dazu gibt es keine spezifischen Entscheidungspunkte.



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.

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`).

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` zu finden ist, versehen.

```
/*
 * 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.
 */
```

6.2.1 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

A.2 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 laufen gelassen werden können.

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)

Tabelle 4: Projektorganisation

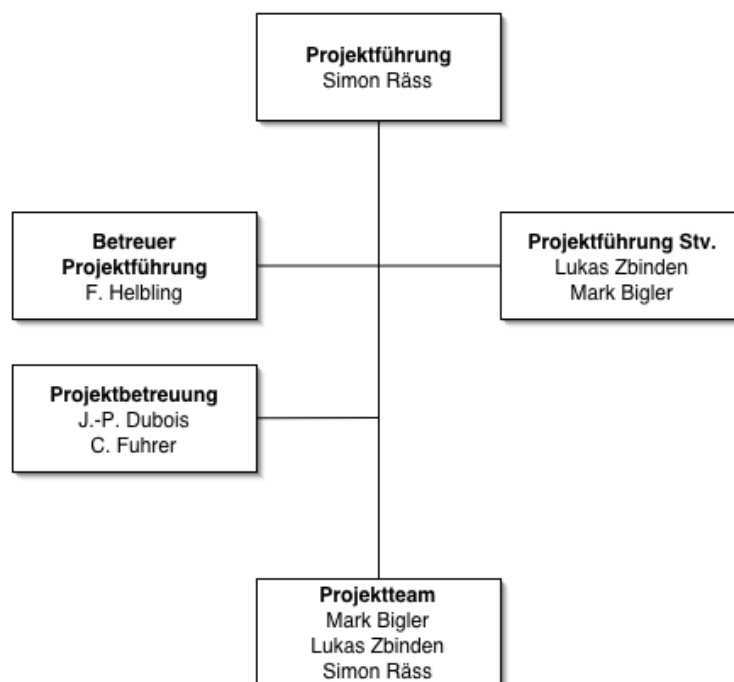


Abbildung 1: Projektorganisation

Anfang der Semesterarbeit	21.02.2005
Beginn der Initialisierungsphase	21.02.2005
Ende der Semesterarbeit	24.07.2005

Tabelle 5: Projektplanung

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.

Für das Konfigurationsmanagement wird unser eigens installiertes Subversion Repository verwendet. Das entsprechende Repository ist unter <http://ace.iserver.ch:81/repos/ace/> erreichbar. Aus



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

Tabelle 6: Versionsnummern

der commit Nachricht soll erkenntlich sein, wenn die Versionsnummer des Dokumentes erhöht wurde.

A.5 Sicherheit