

Projekthandbuch

Mark Bigler

Simon Räss

Lukas Zbinden

30. März 2005



Inhaltsverzeichnis

1	Einleitung	5
1.1	Zweck des Dokuments	5
2	Projektbeschreibung	5
2.1	Ausgangssituation	5
2.2	Ziele	5
2.3	Vorgehensstrategie	5
3	Projektspezifisches Vorgehensmodell	6
3.1	Phase Initialisierung	6
3.2	Phase Entwicklung kollaborativer Editor	7
3.2.1	Komponente Algorithmus	7
3.2.2	Komponente Netzwerk	7
3.2.3	Komponente GUI	7
3.2.4	Komponente User Stories	8
3.3	Phase Abschluss	8
4	Entscheidungspunkte und auszuliefernde Ergebnisse	8
4.1	Initialisierung	8
4.2	Synchronisationsalgorithmus	8
4.3	Konzept GUI/Netzwerk	8
4.4	User Stories	9
5	Methoden und Werkzeuge	9
5.1	Dokumente	9
5.2	Source Repository	9
5.3	Projektwebsite	9
5.4	Zeiterfassung	9
5.5	Trac	9
5.6	Kalender	10
5.7	Entwicklung	10
6	Standards und Richtlinien	10
6.1	Dokumentation	10
6.2	Quellcode	10
A	Anhang: Ergänzende Projektvereinbarungen	11
A.1	Projektorganisation	11
A.2	Projektplanung	11
A.3	Qualitätssicherung	12
A.4	Konfigurationsmanagement	12
A.5	Sicherheit	12



Tabellenverzeichnis

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

Abbildungsverzeichnis

1	Projektspezifisches Vorgehensmodell	6
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

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	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 einen funktionierenden Synchronisationsalgorithmus zu entwickeln. Die Ziele sind teilweise unabhän-

gig voneinander. Grundsätzlich werden pro unabhängigem Teil die Phasen Voranalyse und Konzept durchlaufen. Beim Teilziel Synchronisationsalgorithmus werden zusätzlich noch eine Implementierung und damit verbundene Tests erstellt.

3 Projektspezifisches Vorgehensmodell

Das Vorgehen wird durch drei Hauptphasen bestimmt (siehe Abbildung 1). Die erste Phase initialisiert das Projekt. Die zweite Phase definiert das tatsächlich projektspezifische Vorgehensmodell und wird 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.

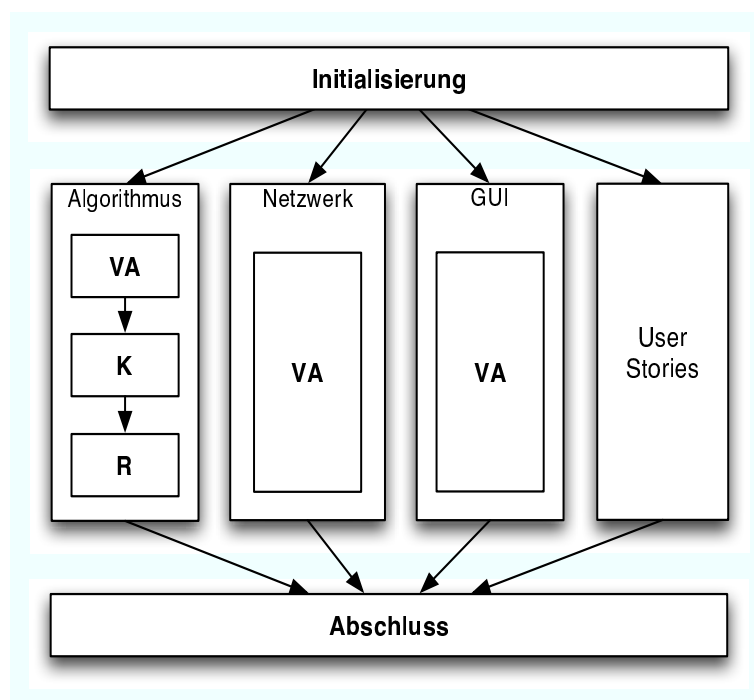


Abbildung 1: Projektspezifisches Vorgehensmodell

Nachfolgend werden die drei Hauptphasen detaillierter erklärt.

3.1 Phase Initialisierung

Die Phase Initialisierung beinhaltet folgende Arbeiten:

- Einrichten der Projektumgebung (Arbeitsplatz, Rechner,...)
- Installation von Projektmanagement-Tools
- Installation von Entwicklungs-Tools
- Aufbau der Projektwebsite



- Fertigstellen der Zeiterfassungs-Software

Zusätzlich werden Projektmanagement spezifische Dokumente erstellt:

- Projektantrag
- Pflichtenheft
- Projekthandbuch
- Projektplan

3.2 Phase Entwicklung kollaborativer Editor

Die Phase wird in 4 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. 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 aufgebaut. Mit diesem Know-how soll anschliessend eine Marktstudie über die aktuellen Synchronisationstechniken erarbeitet werden. Diese Studie ermöglicht die Auswahl einer oder eventuell mehrerer Synchronisationstechniken für die Implementation. Dies führt zur Freigabe der Phase Konzept.

Phase Konzept In der Phase Konzept soll eine fundierte Beurteilung der in der Phase Voranalyse gemachten Auswahl von Synchronisationstechniken erfolgen. Sobald eine bestimmte Synchronisationstechnik bestimmt worden ist, beginnt die Entwicklung des Testframeworks parallel zum Algorithmus. Das zu erarbeitende Konzept über die Implementation des Synchronisationsalgorithmus legt den Rahmen für die nachfolgende Realisierung fest.

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 soll eine Marktstudie über die in diesem Gebiet vorhandenen Technologien erstellt werden. Diese Studie soll die Auswahl einer bestimmten Technologie für das Projekt ermöglichen.

3.2.3 Komponente GUI

Voranalyse Es soll eine Evaluation der GUI Möglichkeiten erfolgen, welche die gewählte Programmiersprache anbietet.



3.2.4 Komponente User Stories

Ein Bericht mit sogenannten User Stories wird erstellt.

3.3 Phase Abschluss

In der Phase Abschluss wird das Projekt geordnet zu Ende gebracht. Alle Projektaktivitäten werden abgeschlossen. Dazu wird ein Abschlussbericht über das Projekt erstellt. Jedes Teammitglied verfasst zusätzlich einen Erfahrungsbericht.

4 Entscheidungspunkte und auszuliefernde Ergebnisse

4.1 Initialisierung

Ergebnisse

- Projektantrag
- Projekthandbuch
- Pflichtenheft
- Projektplan

Entscheidungspunkte

- keine

4.2 Synchronisationsalgorithmus

Ergebnisse

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

Entscheidungspunkte

- Auswahl Synchronisationsalgorithmus basierend auf Bericht
- Abschluss Implementation

4.3 Konzept GUI/Netzwerk

Ergebnisse

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



Entscheidungspunkte

- Auswahl Lösung

4.4 User Stories

Für die Erfassung des Anwendungsumfangs sollen User Stories erfasst werden.

Ergebnisse

- User Stories Katalog

Entscheidungspunkte

- keine

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

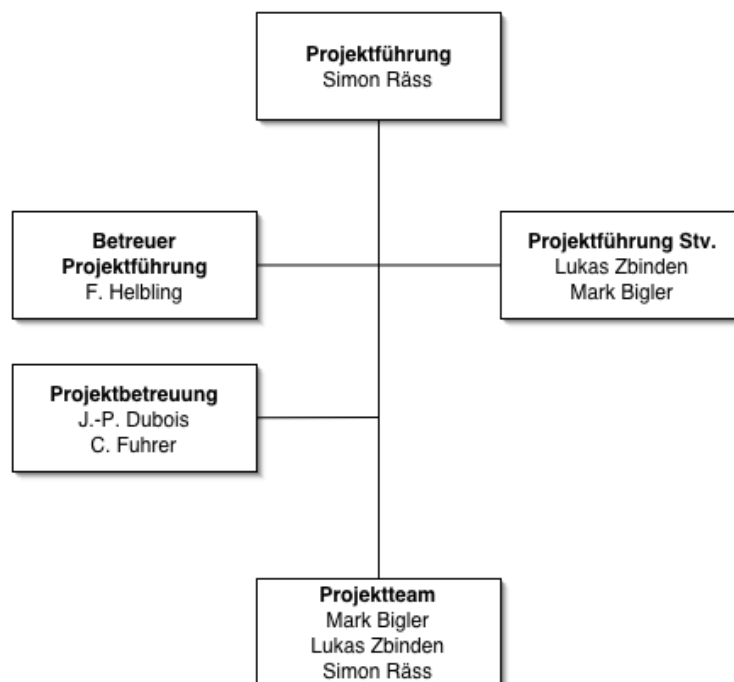


Abbildung 2: Projektorganisation

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.

A.5 Sicherheit

- Subversion Backup - ??