

ACE

A Collaborative Editor

Projektarbeit im Modul Projektführung

Projektantrag

Ausgabe vom:	11.03.2005
Version:	1.0
Status:	Freigabe für Review
Autor:	Mark Bigler Simon Räss Lukas Zbinden
Empfänger:	Jean-Paul Dubois Claude Fuhrer Frank Helbling
Ablage:	Subversion Repository

Allgemeines

Änderungskontrolle

Version	Datum	Wer	Bemerkungen
0.1	02.03.2005	Projektteam	Erster Entwurf
0.2	04.03.2005	Simon Räss	Verbesserungen
0.3	09.03.2005	Projektteam	Überarbeitung nach PM Coaching
1.0	11.03.2005	Lukas Zbinden	Abschluss

Inhaltsverzeichnis

1. Zweck des Dokumentes.....	5
2. Kurzbeschreibung	5
3. Ausgangslage.....	5
3.1. Problemstellung	5
3.2. Anlass und Begründung des Projektes	5
3.3. Projektrandbedingungen.....	6
3.4. Situationsanalyse.....	6
3.5. Erbrachte Vorleistungen.....	7
4. Ziele und Lösungen	7
4.1. Zielvorstellungen.....	7
4.2. Mögliche Lösungen.....	7
4.3. Bewertung der Sicherheits- und Datenschutzaspekte.....	8
5. Mittelbedarf.....	8
5.1. Sachmittel	8
5.2. Personal	8
5.3. Ausbildung	8
5.4. Dienstleistungen	9
6. Planung und Organisation	9
6.1. Projektorganisation.....	9
6.2. Termine	10
6.3. Prioritäten	10
7. Wirtschaftlichkeit	10
7.1. Kosten	10
7.2. Nutzen	11
8. Konsequenzen.....	12
8.1. Auswirkungen	12
8.2. Risikobeurteilung	12
8.3. Ausweichmöglichkeiten	12



9. Antrag.....	13
9.1. Bisherige Entscheide.....	13
9.2. Antrag zur Projektfreigabe	13

1. Zweck des Dokumentes

In diesem Dokument wird die Ausgangslage für das Projekt *ACE* definiert, um über das weitere Vorgehen zu entscheiden.

Das Projekt nimmt Bezug auf diverse Forschungsarbeiten im Bereich CSCW (Computer-Supported Cooperative Work).

2. Kurzbeschreibung

Das Ziel dieses Projektes ist es, einen kollaborativen Editor zu entwickeln. Unter Kollaboration versteht man die Zusammenarbeit mehrerer Einzelpersonen oder einer Gruppe. Dieser Editor soll es mehreren Personen ermöglichen, 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.

3. Ausgangslage

3.1. Problemstellung

Wenn heute ein Dokument von mehreren Personen bearbeitet werden soll, ist ein aufwendiges Vorgehen notwendig. Als Beispiel folgendes Szenario: Jede Person erhält eine Kopie des zu bearbeitenden Dokumentes. Anschliessend macht jeder Beteiligte seine Änderungen und schickt das Dokument zusammen mit einer Beschreibung der Änderungen zurück. Die zuständige Person muss schlussendlich alle Änderungen der Beteiligten in einem Dokument korrekt zusammenführen. Die Aktion ist zeitintensiv und fehleranfällig.

Das gemeinsame Entwerfen eines elektronischen Dokumentes (z.B. Textdatei), wobei jeder Beteiligte mit seinem eigenen Computer arbeitet, ist noch praktisch unbekannt. Der kollaborative Editor soll komplett neuartige und effiziente Editier-Möglichkeiten bieten. Neue Wege der Teamarbeit werden eröffnet.

3.2. Anlass und Begründung des Projektes

Die Idee eines kollaborativen Editors ist sehr innovativ und verspricht breite Anwendungsmöglichkeiten. Gemäss Recherchen existiert bis heute keine in breiten Kreisen verwendete, plattformunabhängige Applikation dieser Art. Die Ausnahme ist der

Editor *SubEthaEdit* (<http://www.codingmonkeys.de/subethaedit/>) welcher aber ausschliesslich für das Betriebssystem Mac OS X entwickelt worden ist. Die Entwickler von *SubEthaEdit* zeigen kein Interesse daran, den Editor für andere Plattformen zu portieren (siehe <http://www.codingmonkeys.de/subethaedit/faq.html>). Da ergibt sich die Chance, einen plattformunabhängigen Editor dieser Art auf den Markt zu bringen. Zudem gibt es Stimmen im Internet, die das Potenzial dieser Applikation erkannt haben. Diese bedauern aber die Beschränkung von *SubEthaEdit* auf Mac OS X und wünschen sich einen solchen Editor für die Betriebssysteme Windows oder Unix:

I'm really excited about this technology at the moment, not only because it's very cool but also because of the immediate, real-world uses it can be put to, especially for businesses. The only drawback, however, is a major one. It's only available for OS X at the moment, which makes it next to useless in a business context. This product can be big – but it's going to have to run on Windows first. (28.05.04)

Quelle: <http://www.allinthehead.com/retro/205/collaborative-document-editing>

Ziel des Projektes ist es, einen kollaborativen Editor für diese Marktlücke zu entwickeln.

Folgend sind einige konkrete Anwendungsgebiete aufgelistet:

- 5 Lektoren eines Buchverlages korrigieren zusammen einen neu geschriebenen Roman.
- Die 8-köpfige Geschäftsleitung eines Technologieunternehmens macht ein Brainstorming über die strategische Ausrichtung im neuen Geschäftsjahr. Alle schreiben ihre Ideen ins gleiche Dokument, jeder von seinem Computer aus.
- Extreme Programming (XP): Es bestehen Teams von 2 Entwicklern. Der eine programmiert, der andere beobachtet den geschriebenen Code von seinem Computer aus.
- Eine Praktikantin bereitet den Schulunterricht vor. Sie erarbeitet dazu einen Ablauf mit Hilfe des kollaborativen Editors. Die Betreuerin, zur gleichen Zeit in ihrem Büro, unterstützt sie dabei.
- Ein Dozent führt mit Hilfe des kollaborativen Editors seinen Studenten Programmiertricks vor. Jeder Student verfolgt diese auf seinem eigenen Computer.
- Gemeinsames Erstellen eines Berichtes, wobei die beteiligten Personen an verschiedenen Abschnitten arbeiten.

3.3. Projektrandbedingungen

- Anwendungen/Prototypen werden mit Java erstellt.
- Das Projekt muss nach Hermes 03 abgewickelt werden.
- Die Semesterarbeit muss am 24. Juni 2005 abgeschlossen sein.

3.4. Situationsanalyse

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.

3.5. Erbrachte Vorleistungen

Es gibt im Gebiet CSCW (Computer Supported Collaborative Work) bereits einige Forschungsarbeiten und Lösungsansätze. Seit den frühen neunziger Jahren wurden Algorithmen zum synchronen Bearbeiten von Dokumenten entwickelt. Das zugrunde liegende Konzept dieser Algorithmen ist als „Operational Transformation“ bekannt. Unter diesem Begriff findet man bei Google eine Fülle von Artikeln. Bis heute ist aus diesen Kenntnissen nur ein kommerzielles Produkt (*SubEhtaEdit*) entstanden.

4. Ziele und Lösungen

4.1. Zielvorstellungen

Es soll ein kollaborativer Editor entwickelt werden, der sich durch Qualität, Benutzerfreundlichkeit und Performanz auszeichnet.

Kurzfristig sinnvolle Ziele sind:

- Synchronisationsalgorithmus für Textdokumente
- Prototyp eines kollaborativen Texteditors

Längerfristige Ziele sind:

- Kollaborativer Texteditor
- Konfiguration und Installationsaufwand des Texteditors auf ein Minimum reduzieren
- Erweitern des Synchronisationsalgorithmus für andere Dokumenttypen
- Framework für kollaboratives Editieren

4.2. Mögliche Lösungen

Der erste Lösungsansatz ist eine Client-Server Applikation. Alle zu bearbeitenden Dokumente werden auf einem zentralen Server abgelegt. Client Anwendungen verbinden sich anschliessend mit dem Server um ein Dokument zu editieren.

Der zweite Lösungsansatz ist eine Peer-to-Peer Applikation. Ein zentraler Server entfällt. Und damit dessen Installation und Wartung. Die Inbetriebnahme der Applikation wird wesentlich vereinfacht. Client Anwendungen können gewünschte Dokumente zur Bearbeitung durch Dritte freigeben. Die Clients verbinden sich dann direkt miteinander.

Zu favorisieren ist die Peer-to-Peer Lösung, weil sie anwenderfreundlicher ist, d.h. die aufwendige Installation eines separaten Servers entfällt. Das macht die Applikation schlanker, unabhängiger und attraktiver für den spontanen Gebrauch. Kollaboratives Arbeiten soll überall dort möglich sein, wo mehrere Anwender im gleichen Netzwerk den Editor in Betrieb haben. Mit Hilfe bestehender Technologien lassen sich dann dynamisch alle laufenden Applikationen in einem Netzwerk erkennen. Die dadurch gewonnene Flexibilität verringert auch die Konfiguration. Die Peer-to-Peer Architektur erscheint für den kollaborativen Editor pragmatischer und wird deshalb bevorzugt.

4.3. Bewertung der Sicherheits- und Datenschutzaspekte

Beim gemeinsamen Editieren eines Dokumentes wird der Inhalt des Dokumentes über ein Netzwerk übertragen. Dabei ist es einer Drittperson möglich, den Inhalt des Dokumentes einzusehen und zu verändern (Man-in-the-middle-attack). Je nach Inhalt des Dokumentes müssen die Daten geschützt sein. Die zu entwickelnde Lösung soll es ermöglichen, Daten vertraulich zu übermitteln. Es wird hier erwähnt, dass Sicherheitsaspekte nicht zu den Kernaspekten der zu entwickelnden Anwendung gehören.

In Folgeprojekten könnte eine Benutzer Authentifizierung in die Applikation integriert werden.

5. Mittelbedarf

5.1. Sachmittel

Die Hochschule für Technik und Informatik stellt für das Projekt die folgenden Sachmittel zur Verfügung.

- 3 PC Arbeitsplätze
- 1 Continuous Integration Server

Zudem werden diverse Open Source Bibliotheken und Werkzeuge eingesetzt.

5.2. Personal

- 3 Studenten
- 2 Projektbetreuer
- 1 Betreuer Projektführung

5.3. Ausbildung

Das Projekt setzt eine gründliche Analyse der auf diesem Gebiet existierenden Kenntnisse (Algorithmen, Technologien, usw.) voraus. Aus diesem Grund wird ein grosser Teil der Zeit in Forschung investiert werden. Ein fundiertes Verständnis der

Synchronisationsalgorithmen ist unerlässlich für ein erfolgreiches Gelingen dieses Projektes.

5.4. Dienstleistungen

Es werden keine kostenpflichtigen Dienstleistungen in Anspruch genommen.

6. Planung und Organisation

6.1. Projektorganisation

6.1.1 Projektteam

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

6.1.2 Projektbetreuer

- Jean-Paul Dubois: doj@hta-bi.bfh.ch
- Claude Fuhrer: frc@hta-bi.bfh.ch

6.1.3 Betreuer Projektführung

- Frank Helbling: frank.helbling@helbling-consulting.ch

6.1.4 Organigramm

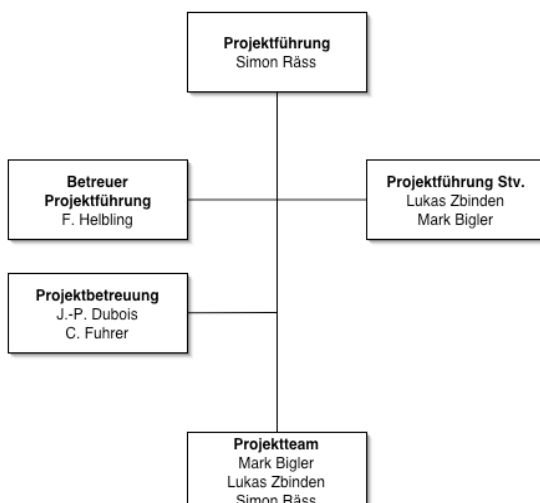


Abbildung 1: Organigramm

6.2. Termine

Folgende Termine sind von der HTI Biel vorgegeben:

Projektstart	21. Februar 2005
Projektantrag	11. März 2005
Pflichtenheft	5. April 2005
Projekthandbuch	15. April 2005
Projektplan	15. April 2005
Erfahrungsbericht	17. Juni 2005
Projektabschluss	24. Juni 2005

6.3. Prioritäten

1. Grundlegend für den Erfolg des Projektes ist das Evaluieren, Implementieren und Austesten eines funktionierenden Synchronisationsalgorithmus.
2. Entwickeln eines Prototyp Texteditors.

7. Wirtschaftlichkeit

7.1. Kosten

Bei den folgenden Kosten handelt es sich um theoretische Kosten, da der Aufwand nicht wirklich verrechnet wird. Der Stundenansatz für Studenten wurde auf CHF 150.- und für Dozenten und Betreuer auf CHF 200.- festgelegt.

7.1.1. Personalkosten:

Person	Anzahl Stunden	Stundenansatz	Total (CHF)
Bigler Mark	300h	150	45000.-
Räss Simon	300h	150	45000.-
Lukas Zbinden	300h	150	45000.-
J.-P. Dubois	15h	200	3000.-
C. Fuhrer	15h	200	3000.-
F. Helbling	5h	200	1000.-
Total	935h		142000.-

7.1.2. Materialkosten

	Anzahl	Preis/Stück (CHF)	Total (CHF)
Arbeitsplatz	3	3000.-	9000.-
Server	1	2500.-	2500.-
Diverses			500.-
Total			12000.-

7.1.3. Gesamtkosten

Teilkosten	CHF
Personalkosten	142000.-
Materialkosten	12000.-
Total Realisierungskosten	164000.-
Risikozuschlag 25%	41000.-
Nettokosten	205000.-
Angestrebter Gewinn	0.-
Bruttokosten	205000.-

Den Risikozuschlag von 25% wird damit begründet, dass der Aufwand für die Auswahl, die Implementation und das Testen eines korrekten Synchronisationsalgorithmus schwer abzuschätzen ist.

7.2. Nutzen

Das in diesem Projekt angeeignete Wissen dient als Grundlage für eine anschliessende Diplomarbeit. Ein Nutzen des Projektes ist das Aneignen von Wissen, welches auch in späteren Arbeiten weiterverwendet werden kann.

Kollaboratives Editieren ist den meisten Anwendern unbekannt. Das Projektteam ist der Überzeugung, dass ein kollaborativer Editor ein völlig neues Zusammenarbeiten ermöglicht. Dieses Zusammenarbeiten wird in vielen Fällen auch effizienter sein. Beim gemeinsamen Erstellen eines Dokumentes tritt immer wieder das Problem auf, dass verschiedene Versionen des Dokumentes zusammengesetzt werden müssen. Dies ist nicht nur aufreibend sondern auch fehleranfällig. Bei einem kollaborativen Editor entfällt dieser Arbeitsschritt vollständig, da alle Änderungen in einem zentralen Dokument gemacht werden.

8. Konsequenzen

8.1. Auswirkungen

In der Projektarbeit soll ein Prototyp erstellt werden, welcher als Basis für die Zielsetzungen der Diplomarbeit dient.

8.1.1. Nichtrealisierung

Bei Nichtrealisierung des Projektes fehlen die Grundlagen für die Realisierung der geplanten Diplomarbeit. Dies hätte zur Folge, dass die Studenten für die Diplomarbeit ein neues Thema aussuchen und realisieren müssten.

8.1.2. Verspätete Realisierung

Der Abgabetermin ist von der HTI Biel festgelegt und kann nicht verschoben werden. Eine verspätete Realisierung ist daher grundsätzlich nicht möglich. Können einzelne Ziele nicht erreicht werden, kann nur eine Teillösung erstellt werden. Dies hat auch Auswirkungen auf die geplante Fortsetzung als Diplomarbeit.

8.1.3. Schnittstellen

Es bestehen keine expliziten Schnittstellen zu anderen Systemen.

8.2. Risikobeurteilung

Das grösste Risiko ist, dass der implementierte Synchronisationsalgorithmus nicht korrekt funktioniert. Dasselbe Dokument muss auf verschiedenen Computern immer den gleichen Inhalt haben. Zudem muss der Algorithmus Randbedingungen wie zum Beispiel Latenzzeiten und Ressourcenverbrauch erfüllen.

8.3. Ausweichmöglichkeiten

Ausweichmöglichkeiten sind kaum vorhanden, da das Projekt bis zum gegebenen Abgabetermin weitergezogen wird. Falls frühzeitig erkannt wird, dass das Projekt nicht vollständig fertiggestellt werden kann, besteht die Möglichkeit nur eine Teilentwicklung anzustreben. Sollte das gewählte Thema sich als nicht geeignet für eine Fortführung im Rahmen der Diplomarbeit erweisen, müsste ein anderes Thema für die Diplomarbeit gesucht werden.

9. Antrag

9.1. Bisherige Entscheide

Falls das Projekt den Erwartungen des Projektteams entsprechen sollte, wird es als Diplomarbeit weitergeführt.

Das Projekt wird am Ende der Diplomarbeit unter einer Open-Source Lizenz (GPL – General Public License) auf dem Internet für die Allgemeinheit zur Verfügung gestellt.

9.2. Antrag zur Projektfreigabe

Das in diesem Dokument beschriebene Projekt erscheint dem Projektteam bezüglich Wirtschaftlichkeit und Risiken als sinnvoll und realistisch. Aus diesem Grund wird hiermit die Freigabe des Projektes *ACE* beantragt.

Die Beteiligten sind durch ihre Unterschrift mit diesem Dokument vollumfänglich einverstanden.

Für das Projektcoaching

.....

Frank Helbling

Für die Projektleitung

.....

Simon Räss

Für die Projektbetreuung

.....

Jean-Paul Dubois

Für die Projektbetreuung

.....

Claude Fuhrer