Projekt **Collaboral Damage**

Projektvorschlag

Kurzzusammenfassung:
Das Projekt soll Benutzern ermöglichen, gemeinsam an einem Dateienpool zu arbeiten und Notizen bzw. Ankündigungen zu organisieren. Der vorliegende Projektvorschlag führt in die technische Planung und Arbeitsplanung ein.

Autor:	Johannes Buchner
Review:	
Gruppe:	3950

Nr	Datum	Autor	Änderung
1	11.04.2008	Johannes Buchner	Dokument erstellt

Inhaltsverzeichnis

Kapitel 2

Projektauftrag

2.1 Projektbeschreibung

Die Weiterentwicklung von Collaboral Damage wird es Projektgruppen erlauben, über das Internet an Projektordnern zu arbeiten.
Collaboral Damage soll den Grundstein für eine Plattform legen, die es erlaubt, über ein Netzwerk (z.B. das Internet) gemeinsam an Dateien beliebigen Formats zu arbeiten. Es soll ein Fat Client entwickelt werden, der alle Funktionen der im Projekt definierten Synchronisationsschnittstelle benutzt, die Implementierung der Netzwerk- und Synchronisationsservices ist aber erst in darauf folgenden Ausbauphasen in Form eigenständiger Projekte geplant. Änderungen an Dateien sollen vom Programm erkannt werden und mit Hilfe der Synchronisations- und Netzwerkservices an andere User propagiert werden. In diesem Projekt soll nur die Phase 1 der folgenden Phaseneinteilung realisiert werden.

2.1.1 Phase 1, Fat Client

In dieser Phase wird das Programm als Fat Client erstellt, dessen grafischen Oberfläche die vollständige Nutzbarkeit der unten aufgeführten Features benutzbar macht.
Die Netzwerkkommunikation zwischen verschiedenen Clients soll mithilfe eines Mock-Service simuliert werden. Dieser Service wird in dieser Phase die Zusammenarbeit mit anderen Clients simulieren, wodurch die Funktionalität des Programmes getestet werden kann. Neben diesem Mock-Service, welches die Netzwerkservices der Anwendung kapselt, wird zusätzlich noch ein Synchronisationsinterface erstellt, welches es erlaubt, den Vorgang der Synchronisation zwischen den Clients, auf verschiedene Weisen zu implementieren.
Die Synchronisation soll in dieser Phase ebenfalls mithilfe eines Mock-Services realisiert werden. Die Netzwerk- und Synchronisations-Mock-Services können dann in möglichen späteren Projektphasen durch entsprechende Implementierungen (z.B. XMPP für das Netzwerkservice) ersetzt werden. Die für die Synchronisation notwendigen Elemente der Benutzeroberfläche sollen aber bereits in dieser Phase erstellt und an die entsprechenden Schnittstellen gebunden werden.

Aufgaben des Networkservice

- Authentifizierung der Benutzer
- Netzwerkverbindung zwischen den Clients
- Austausch von Nachrichtenpaketen zwischen den Clients
- Datenaustausch zwischen den Clients

Aufgaben des Synchronisationsservice

- Abholen von Dateiversionen, die andere Projektmitglieder erstellt haben
- Verbreiten eigener Änderungen
- Abgleich von Dateiversionen zwischen Clients
- Erkennen von Dateikonflikten

2.1.2 Phase 2, Networking / Synchronisation

Die Mock-Services werden durch konkrete Implementierungen des Networkservice und des Synchronisationsservice ersetzt. Für das Networkservice ist zur Zeit eine Lösung auf Basis des XMPP-Protokolles angedacht. Durch eine generischen Definition der Schnittstellen in Phase 1 kann dies aber auch mit beliebigen anderen Technologien erfolgen.

2.1.3 Phase 3, Service Sharing

In dieser Phase ist das zur Verfügung stellen lokaler Services (z.B. Printer Server) zwischen den Projektmitgliedern geplant.

2.2 Arbeitsstruktur

2.2.1 Auftraggeber

Rolle	Name	Mail	Telefon
betreuender Assistent	Marco Zapletal	marco@ ec.tuwien.ac.at	+43 (1) 588 01 - 18822
betreuender Tutor	Anton Matzneller	anton.matzneller@ googlemail.com	

2.2.2 Auftragnehmer

Rolle	Name	Mail	Telefon	Matr.	KZ
TK	Simon Wallner	me@ simonwallner.at	+43 699 11 55 24 51	0625104	532
TKS	Peter Steinberger	peter.steinberger@ student.tuwien.ac.at	+43 664 918 37 24	0626583	534
TA	Chris Sutter	chris@ doublesignal.com	+43 660 61 61 808	0505267	534
TAS	Philipp Knobelspies	e0547943@ student.tuwien.ac.at	+43 699 81 39 93 84	0547943	534
Test	Dominik Dorn	dominik.dorn@ gmail.com	+43 669 12 64 79 73	0626165	534
Doku	Johannes Buchner	e0625457@ student.tuwien.ac.at	+43 669 10 04 33 47	0625457	534

2.2.3 Main Stakeholder

Personen, die geringe bis mittlere Computererfahrung haben und in Projekten Dateien verschiedener Formate bis zu einer Größe von ca 5mb austauschen und zusammen bearbeiten möchten. Die Projektmitglieder sind während der Arbeit an dem Projekt die meiste Zeit online.

2.2.4 Modellszenario

Eine Projektgruppe, deren 3-12 Mitglieder auf verschiedenen Rechnern arbeiten, die vorwiegend online sind und gemeinsam 5-100 Dateien benutzen. Eine einzelne Datei wird dabei meist nur gleichzeitig von einem Benutzer bearbeitet.

2.3 Arbeitsziele

2.3.1 Betriebswirtschaftliche Ziele

- Die Zeit die für das Organisieren und Durchsuche von alten Versionen eines Dokuments aufgewendet wurde kann nun für andere Tätigkeiten verwendet werden.

2.3.2 Funktionale Ziele

- Durch den Einsatz der Applikation wird es einfacher ad-hoc neue Dokumente der Projektgruppe zur Verfügung zu stellen oder Aktualisierungen an bestehenden zu Propagieren. Da dieser Austausch nicht mehr per Mail geschieht wird die Übersicht über die Daten erhöht, und Versionskonflikte mit alten lokalen Versionen stark verringert.

- Durch den Einsatz der Applikation können Aktualisierungen an Dateien anderen Projektmitgliedern schneller zugänglich gemacht werden. Da die Projektmitglieder, sofern möglich, immer die aktuellsten Versionen zur Verfügung haben, ist eine dynamischere Arbeitsweise möglich, die stärker auf Zusammenarbeit setzt.
- Da der Dateiaustausch nicht mehr händisch per Mail geschieht müssen alte Versionen nicht mehr manuel organisiert werden, wodurch ein Versionschaos leichter vermieden werden kann.
- Treten dennoch Datei-Versionskonflikte auf, wird der Benutzer von der Applikation bei deren Lösung unterstützt, wodurch diese einfacher zu handhaben sind, und weniger Zeit in Anspruch nehmen.

2.3.3 Soziale Ziele

- Durch den einfacheren Datenaustausch wird die engere Zusammenarbeit der Projektmitglieder unterstützt.

2.3.4 Lieferkomponenten

Bei Projektabschluss werden folgende Komponenten übermittelt:

- die voll funktionale Applikation laut Anforderungsspezifikation als lauffähiges .jar Paket (benötigt JRE 1.6).
- Benutzerhandbuch
- Anforderungsspezifikation
- technische Dokumentation

Die gesamte Dokumentation wird in einer Website zur Verfügung gestellt. Das Programm sowie die Dokumentation werden in englischer Sprache verfasst.

2.3.5 Weitere Komponenten

- Projektvorschlag
- Projektauftrag
- Projekt-Wiki
- Dokumente der internen Projektorganisation
- Artefakte des laufenden Projektmanagement
- Stundenlisten

- Projekttagebuch
- Protokolle

2.4 Use Cases

- Projekte verwalten
- Projektmitglieder verwalten/einladen
- Dateien/Ordner zum Projektdatenpool hinzufügen
- Dateien/Ordner aus dem Projektdatenpool entfernen
- Datei aus dem Projektdatenpool zur Bearbeitung mit einer externen Applikation öffnen
- Notizen organisieren
- Metadaten des Projekts verwalten
- Metadaten der Dateien verwalten
- Metadaten der Projektmitglieder verwalten
- Lokale Änderungen an Projektmitglieder propagieren
- Aktualisierte Versionen von Projektmitgliedern holen
- Versionskonflikt lösen
- Nachrichten an Projektmitglieder schicken
- Nachrichten empfangen

2.5 Komponentendiagramm

Das Projekt ist in 6 Komponenten aufgeteilt.

- Core
- Graphical User Interface (GUI)
- Database Persistence (DAO)
- Synchronisation Services
- File System Services
- Network Service

2.5.1 Core

In der Core-Komponente befindet sich die Business Logic der Applikation. Hier wird entschieden, was bei bestimmten User-Eingaben (z.B. das Drücken des Buttons "Synchronisieren") gemacht werden soll. Die Core Komponente deligiert die vom User gewünschte Aktion an die einzelnen Komponenten, fügt die Ergebnisse der Aktionen zusammen und gibt diese zurück an die GUI. Der Core reagiert aber auch auf Events, die durch die einzelnen Komponenten gemeldet werden, z.B. wenn eine Datei im Dateisystem geändert wird, wird dies über den Core an die GUI gemeldet und je nach User Interaktion weiter an die Synchronisationskomponente geleitet.

2.5.2 Graphical User Interface

Das Graphical User Interface (GUI) ist die grafische Benutzeroberfläche, mit welcher der Endanwender arbeitet. Die GUI ermöglicht Zugriff auf alle von der "CoreKomponente für Endbenutzer zur Verfügung gestellten Funktionalitäten.

2.5.3 Database Persistence

Die Database Persistence Komponente abstrahiert den Zugriff auf die Datenbank und ermöglicht es, definierte Aktionen einfach und geregelt (z.B. mittels Transaktionen) auszuführen. Es wird das Konzept des Data Hiding umgesetzt, wodurch erreicht werden kann, dass der Core bzw. im schlimmsten Falle andere Clients keinen direkten Zugriff auf die Datenbank erhalten, sondern nur auf die spezifizierten und notwendigen Funktionen.

2.5.4 Synchronisation Services

Die Synchronisationskomponente arbeitet sehr eng mit dem Core zusammen. Der Core übermittelt beispielsweise eine Änderung einer Datei, welche durch den User bzw. automatisch (je nach Einstellung) bestätigt wurde.Die Synchronisationskomponente beinhaltet nun verschiedene Strategien, wie diese Änderungen an andere Projektmitglieder/Clients propagiert werden.

2.5.5 File System Services

Die File System Services sind die Schnittstelle des Programmes zum Dateisystem des Benutzers. Dieser Service kapselt den kompleten Zugriff auf Dateien im Dateisystem, sodass diese vom Programm wie jegliche andere Objekte (z.B. Notizen) verwendet werden können. Außerdem kann der Dateisystem Service durch entsprechende Strategien feststellen, ob Dateien geändert wurden oder in die Projektordnerstruktur kopiert wurden und dies dem Core mitteilen, welcher wiederum entsprechende Aktionen veranlasst.

2.5.6 Network Service

Der Network Service behandelt alle Netzwerktätigkeiten des Programmes. Der Service kapselt beispielsweise vollständig die Kommunikation der Clients auf Netzwerkebene und gibt die entsprechenden Nachrichten an den Core weiter. So ist es leicht möglich, verschiedene Netzwerkbackends (z.B. XMPP oder RMI) zu unterstützen, welche für den Core und somit für den Benutzer transparent sind. Außerdem wird die Authentifizierung der Nutzer in dieser Komponente durchgeführt.

2.6 Projektplan

Tätigkeit	**Kurzbeschreibung**	**ETA**
Anforderungsanalyse abgeschlossen	Abschluss der Anforderungsanalyse	
Schnittstellendefinition abgeschlossen	Abschluss der Schnittstellendefinition	
Entwurfsphase abgeschlossen	Abschluss des technischen Entwurfs	
Beginn Implementierung	Start der Implementierungsphase	
Schnittstellen Implementiert	Abschluss der Schnittstellenimplementierung	
“half-way-done“	50% der Implementierung abgeschlossen	
UI abgeschlossen	Abschluss der Implementierung des UI	
Feature Complete	Alle Features sind implementiert auch wenn diese noch nicht einwandfrei funktionieren	
Release	Fertigstellung und Release der Applikation	Di, davor
Projektabschluss	Ablieferung aller Artefakte, Applikation, Doku, etc. Ende der Übung	Fr, 20.6.2008

2.7 Arbeitsprogramm, Work Breakdown Structure

2.8 Projektabgrenzung

- Im Rahmen der Laborüubung SEPM wird nur die erste Phase laut Projektbeschreibung implementiert. Die konkrete Implementierung der Netzwerk- und Synchronisationsfunktionalität erfolgt erst in einer späteren Projektphase und wird in der ersten Phase durch Mock-Objekte simuliert.
- Es ist nicht möglich, in Echtzeit gleichzeitig an einem Dokument zu arbeiten (wie etwa in Gobby oder Google Apps).

- Es soll kein SCM oder Versionsmanagement implementiert werden. Alte Versionen von Dateien werden nicht behalten und sind daher auch nicht wiederherstellbar.
- Da der Fokus auf Nutzungsumgebungen liegt, die primär binäre, beziehungweise proprietäre Formate verwenden, soll kein automatisches Mergen (etwa von Textdateien) implementiert werden.
- Sämtliche Synchronisation soll nur zwischen Clients ablaufen und es soll keinen Server geben, der Projektdaten zentral verwaltet oder die Problematik von "verlorenen" Dateien beim Offlinegehen des letzten Clients mitigiert. Ein ähnliches Verhalten kann aber durch einen ständig online befindlichen Client im "passive mode" emuliert werden.
- Es soll kein Editor für Dateien direkt in die Applikation integriert sein. Das Bearbeiten von Dateien wird mittels externer Anwendungen gehandhabt.
- Es gibt keine explizite Rechteverwaltung, in der User Rechte für die von ihnen eingestellten Dateien festlegen, d.h. alle User sind gleichberechtigt.
- Es ist nicht möglich, über mehrere Ordner verteilte Dateien zu sharen, d.h. es gibt einen einzigen Projektordner, dessen Inhalt samt Unterordnern für ein Projekt behandelt wird und es ist nicht möglich im Projektordner Dateien auszuwählen, die nicht synchronisiert werden sollen.

2.9 Kostenabschätzung

2.10 Informationswesen/Dokumentation

2.10.1 Interne Kommunikation

Zur internen Kommunikation wird ein eigens eingerichtetes Wikisystem verwendet. Ankündigungen und wichtige Mitteilungen werden über eine Mailingliste verteilt. Die gesamte Projektgruppe trifft sich mindestens einmal pro Woche zu einem ein- bis eineinhalbstündigen Meeting. Die Agenda für die Meetings wird zuvor im Wiki bekanntgegeben und kann dort diskutiert werden.

2.10.2 Externe Kommunikation

Die externe Kommunikation wird während der Übung über regelmäßig stattfindende Review-Meetings sowie über Emailkommunikation mit unserem Tutor und Assistenten geführt.

2.10.3 Organisatorische Dokumentation

Die organisatorische Dokumentation des Projekts wird über das Wiki abgewickelt. Protokolle, Stundenlisten, Projekttagebuch, etc. werden während der Dauer der Übung laufend aktualisiert.

2.10.4 Technische Dokumentation

Die gesamte technische Dokumentation und Spezifikation wird über Maven abgewickelt. Dies Umfasst Dokumente der technischen Planung, Dokumente der Anforderungsspezifikation, Dokumente der Qualitätssicherung, Dokumentation auf Codeebene, Endbenutzer Dokumente, etc.
Die gesamte technische Spezifikation wird in englischer Sprache verfasst.