

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

2	Projektauftrag	2
2.1	Projektbeschreibung	2
2.1.1	Phase 1, Fat Client	2
2.1.2	Phase 2, Networking / Synchronisation	3
2.1.3	Phase 3, Service Sharing	3
2.1.4	Projektabgrenzung	3
2.2	Arbeitsstruktur	3
2.2.1	Auftraggeber	3
2.2.2	Auftragnehmer	4
2.2.3	Main Stakeholder	4
2.2.4	Modellszenario	4
2.3	Arbeitsziele	4
2.3.1	Betriebswirtschaftliche Ziele	4
2.3.2	Funktionale Ziele	5
2.3.3	Soziale Ziele	5
2.3.4	Lieferkomponenten	5
2.3.5	Weitere Komponenten	5
2.4	Funktionale Anforderungen	5
2.5	Komponentendiagramm	6
2.6	Projektplan	6
2.7	Arbeitsprogramm, Work Breakdown Structure	6
2.8	Projektabgrenzung	6
2.9	Kostenabschätzung	6
2.10	Informationswesen/Dokumentation	6

Kapitel 2

Projektauftrag

2.1 Projektbeschreibung

Die Weiterentwicklung von Collaboral Damagewird es Projektgruppen erlauben, über das Internet an Projektordnern zu arbeiten.

Collaboral Damagesoll 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 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 Netzwerk-kommunikation 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.

2.1.4 Projektabgrenzung

Es wird nur die erste Phase implementiert, was das Programm nur lokal benutzbar macht, und andere Systeme durch Mock-Services simuliert. Es ist nicht möglich, "live" gleichzeitig an einem Dokument zu arbeiten (wie etwa Gobby oder Google Apps). Es werden keine alten Versionen/Revisionen behalten, die wiederhergestellt werden könnten. Es wird kein automatisches Mergen (etwa von Textdateien) implementiert, da auf Benutzungsumgebungen fokussiert wird, die binäre bzw. proprietäre Formate verwenden.

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@ doublesi- gnal.com	+43 660 61 61 808	0505267	534
TAS	Philipp Knobelspies	e0547943@ stu- dent.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@ stu- dent.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

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.

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 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.

Die Benutzer werden bei der Lösung von Datei-Versionskonflikten von der Applikation unterstützt, wodurch diese einfacher zu handhaben sind.

Aktualisierungen können mit einem Klick an die Projektmitglieder propagiert werden, ohne dafür extra ein Mail zu schicken.

2.3.3 Soziale Ziele

Das Verteilen und Freigeben von Dateien wird wesentlich vereinfacht. Wo früher per Mail Dateien ausgetauscht wurden, wird nun der Projektfolder ganz einfach an alle Projektteilnehmer verteilt. Die Gruppenmitglieder müssen sich keine Gedanken mehr über Dateigrößen oder dem Packen von Ordnern machen. Konflikte und Synchronisationsprobleme werden minimiert, da sich die Dateien einfach als "wird derzeit bearbeitet" markieren lassen.

Der Arbeitsalltag der Projektmitglieder vereinfacht sich, die kurzfristigere Zusammenarbeit der Projektmitglieder unterstützt.

2.3.4 Lieferkomponenten

Die Software wird als ausführbare jar-Datei geliefert. Diese ist mit einer aktuellen Java-Runtime Machine (ab 1.5) lauffähig.

2.3.5 Weitere Komponenten

- Projektdokumentation

2.4 Funktionale Anforderungen

- Authentifizierung bei Programmstart mittels User/Passwort (Login gegen das Netzwerkinterface)
- Erstellen eines Projektes mit Auswahl eines Projektordners.
- Bearbeiten von Benutzern eines Projektes.(mittels Netzwerkservice)
- Teilnahme an einem Projekt bestätigen/ablehnen.
- Erstellen/Verwalten von Labels/Tags für Datenobjekte. (Dateien, Notizen)

- Zuweisen/Entfernen von Labels zu einer/mehreren Dateien.
- Dateiverwaltung (hinzufügen, entfernen) von Dateien/Ordner.
- Hinzufügen von Metadaten zu Datenobjekten.
- Erstellen von Notizen.
- Suchen von Dateien mittels Name, Label/Tag
- Verbindungsmanagement zum Netzwerk. (verbinden, trennen)
- Daten/Notizen können gepusht/gepullt werden.
- Feature Auto-pull: Änderungen werden automatisch heruntergeladen.
- Feature Auto-push: Änderungen werden automatisch verteilt.

* * * * Teilnahme an einem Projekt bestätigen/ablehnen * Spezifikation eines Ordners auf der lokalen Festplatte für den Dateipool * Hinzufügen von Dateien zu einem Dateipool * * * Suchen von Dateien mittels Name * Suchen von Dateien mittels Label/Tag * Erstellen von Notizen * Hinzufügen von Metadaten zu Dateien/Notizen * Verbinden zum Projekt (sofern zuerst offline gearbeitet wurde) * Verbindung zum Projekt trennen (sofern zuerst online gearbeitet wurde) * Explizites Synchronisieren des Projektes (Änderungen herunterladen / veröffentlichen)

2.5 Komponentendiagramm

2.6 Projektplan

Tätigkeit	Kurzbeschreibung	ETA
Anforderungsanalyse abgeschlossen	Abschluss der Anforderungsanalyse	
Beginn Implementierung	Start der Implementierungsphase	
Entwurfsphase abgeschlossen	Abschluss des technischen Entwurfs	
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, Do-ku, etc. Ende der Übung	Fr, 20.6.2008

2.7 Arbeitsprogramm, Work Breakdown Structure

2.8 Projektabgrenzung

2.9 Kostenabschätzung

2.10 Informationswesen/Dokumentation