

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 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 (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

2.4 Funktionale Anforderungen

2.5 Komponentendiagramm

2.6 Projektplan

2.7 Arbeitsprogramm, Work Breakdown Structure

2.8 Projektabgrenzung

2.9 Kostenabschätzung

2.10 Informationswesen/Dokumentation