

TeamFound
Infrastrukturen zur Open Source
Softwareentwicklung
Technische Universität Berlin

A. Bachmann, J. Heese, J. Kechel, M. Klink

SS 2006

Copyright ©2006 Jan Kechel, Martin Klink, Jonas Heese, Andreas Bachmann

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found online at <http://www.gnu.org/copyleft/fdl.html>

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 2 | Neue Features | 2 |
| 2.1 | Neuerungen am Server | 2 |
| 2.1.1 | Web-Crawler | 2 |
| 2.1.2 | Index-Updates | 3 |
| 2.1.3 | Logging | 3 |
| 2.1.4 | Usermanagement | 3 |
| 2.1.5 | Server Einstellungen | 4 |
| 2.2 | Neuerungen am Web-Interface | 5 |
| 2.2.1 | Komponenten | 5 |
| 2.3 | Firefox/Flock-Toolbar | 7 |
| 2.4 | Projekt-Webseite | 8 |
| 2.5 | Neuerungen an der Internet-Explorer-Toolbar | 9 |
| 2.5.1 | Anpassungen | 9 |
| 2.5.2 | Erweiterungen | 9 |
| 3 | Projektverlauf | 10 |
| 3.1 | Organisation und Kommunikation by Jonas | 10 |
| 3.1.1 | Fazit: Probleme wachsender Projekte | 11 |
| 3.2 | Projektverlauf by Jan | 11 |
| 3.3 | Projektverlauf von Martin | 14 |
| 3.4 | Projektverlauf by Andreas | 15 |
| 3.4.1 | Fazit: Begeistert von Qualität | 16 |

Kapitel 1

Einleitung

Das Teamfound-Projekt geht in die zweite Runde, im Wintersemester 2005/2006 wurde im Projekt 'Infrastrukturen zur Open Source Softwareentwicklung' der Grundstein gelegt, in diesem Semester (Sommersemester 2006) geht es nun um die Erweiterung des System sowie dem ausbügeln einiger Ecken und Kanten die bereits im Code vorhanden sind.

Die größten Neuerungen für den normalen Benutzer sind sicherlich die Rechteverwaltung die auch Einzug in das XML-Interface des Servers sowie die beiden Toolbars gehalten hat, sowie die Möglichkeit einen Teamfound-Server auch mit einem vollwertigen Web-Interface zu benutzen.

Unter der Haube sind einige Fehler aufgetaucht, wie zB. Probleme mit dem Encoding von Web-Seiten die nicht in Latin-1 geschrieben sind oder der fehlenden Integration von Updates des Index oder ordentlicher Behandlung von Framesets.

Kapitel 2

Neue Features

2.1 Neuerungen am Server

2.1.1 Web-Crawler

Eine wichtige Komponente in einer Suchmaschine ist der Crawler, welcher die zu indexierenden Dokumente herunterlädt. Die bisherige recht einfache Implementierung hat diesen Job nur teilweise gut erledigt. So war es bisher nicht möglich die Codierung der heruntergeladenen Daten korrekt zu bestimmen. Außerdem konnte Teamfound nicht mit Framesets umgehen, bzw. hat versucht die Datei mit dem Frameset zu indexieren, nicht den Inhalt der eigentlichen Frames.

Um diese Probleme zu lösen wurde der Teamfound-Crawler komplett neu implementiert. Eine dreistufige Suche nach einer Codierungsangabe, nämlich in der Verbindung an sich, im HTTP-Header sowie im HTML-Header. Letzterer Ort ist der wahrscheinlichste Fundort für ein solches Encoding. Es ist dafür also notwendig den kompletten HTML-Code herunterzuladen und dann nach einer entsprechenden Angabe zu suchen. Die Suche nach Angaben zum Encoding sieht die Möglichkeit vor, das Teamfound in der Zukunft mehrere Arten von Dokumenten indexieren kann. Da die Suche nach einem Encoding in PDF-Dateien zB. ganz anders ablaufen wird, muss diese vom Inhaltstyp abhängen. Der neue Crawler lässt sich bequem um solche erweitern.

Die Behandlung von Framesets ist eine zentrale Anforderung, die auf verschiedene Arten gelöst werden kann. So ist die Frage, ob einfach alle Frames in der Reihenfolge ihrer Definition in der Quelldatei heruntergeladen und zu einem langen Dokument zusammengefügt werden, oder ob mehr Aufwand betrieben werden soll, um zB. den Hauptinhaltsframe zu finden und an den Anfang des Ergebnisdokumentes zu stellen. Die neue Implementierung des Crawler verfolgt erstes Schema, alle Frames werden in einfacher Definiti-

onsreihenfolge hintereinander kopiert und dann an den Indexer übergeben, welcher das HTML entfernt und den Text indexiert. So steht aber natürlich nicht immer der Anfang des Inhaltes in der Teamfound-Ergebnisanzeige. Da es aber nicht einfach ist, den Hauptinhaltsframe innerhalb eines Framesets herauszufinden haben wir uns entschieden dieses auch nicht zu versuchen.

2.1.2 Index-Updates

Wurden bisher Internetseiten in einen Teamfound-Server aufgenommen so wurden sie während dieses Vorganges vom Crawler heruntergeladen und in den Index gespeichert. Änderungen an der Seite würden später nicht in den Index aufgenommen werden da bereits eingetragene Seiten nicht erneuert wurden. Um dieses Problem aus der Welt zu schaffen startet ein Teamfound-Server nun einen nebenläufigen Thread der alle Seiten in einem bestimmten Intervall erneut herunterlädt und bei Änderungen neu indexiert.

Dabei können Seiten, wenn der Crawler auf HTTP-Fehler stößt, vorrübergehend deaktiviert werden, oder wenn diese Fehler anhalten, auch komplett aus dem Index entfernt werden.

2.1.3 Logging

Da der Funktionsumfang des Teamfound-Servers stark angewachsen ist, wurde die Nutzung eines echten Logging-Systems notwendig, einerseits vereinfacht dies die Entwicklung, da Debug-Ausgaben ordentlich ausgegeben werden können, andererseits ist es für einen späteren Produktiven Einsatz evtl. wichtig Fehler bei Zugriffen auf den Teamfound-Server oder fehlgeschlagene Logins zu protokollieren. Da während eines produktiven Einsatzes keine oder nur wenige Debug-Nachrichten protokolliert werden sollen, ist ein Logging-System notwendig, das einerseits zwischen der Art von Nachrichten unterscheiden kann, andererseits aber auch Komponenten, welche Nachrichten einliefern, trennen kann. log4j aus dem Apache-Projekt erfüllt alle diese Aufgaben und wurde daher für die Integration in Teamfound ausgewählt. Einzelne Komponenten benutzen unterschiedliche sogenannte 'Logger' und trennen ihre Nachrichten in mehrere Kategorien zwischen Debug, Info sowie Warning und Error auf. Einzelne oder alle Logger können auf die Konsole oder in eine Datei umgeleitet werden.

2.1.4 Usermanagement

Mit Einführung eines Usermanagements wurden viele Erweiterungen an unterschiedlichen Stellen im Server notwendig. Die Datenbank muss erweitert

werden, neue Aufrufe müssen im Servlet entgegengenommen und dann im Server ausgeführt werden. Ausserdem muss bei alter sowie neuer Funktionalität eine Überprüfung der Berechtigung durchgeführt werden. Um vernünftige Nutzbarkeit zu erreichen ist zusätzlich ein Sessionmanagement nötig.

Da vom ServletContainer ein Mechanismus für Sessions angeboten wird, war hier hauptsächlich zu klären welche Informationen zu jeder Session gehalten werden und wie diese organisiert sind.

Die neuen Informationen, die in der Datenbank zu speichern sind, waren ziemlich klar, deshalb konnte die Grundlage ziemlich schnell geschaffen werden ohne das später grössere Änderungen nötig wurden. Die interne Programmlogik war wohl am aufwendigsten. Der wachsende Funktionsumfang und die nötige Authorisierungprüfung ergeben deutlich komplexere Vorgänge als in der vorhergehenden Version zu Milestone 2.

Neue Funktionalität des Servers :

- 3 Benutzertypen : Projektadmin , angemeldeter Nutzer, anonymer Nutzer
- Zuordnung von Nutzern/Admins zu Projekten
- Berechtigungsvergabe auf Projektebene
- Löschen von Kategorien und indizierten Dokumenten durch berechtigte Nutzer

2.1.5 Server Einstellungen

Bei wachsender Funktionalität des Servers wird auch eine vernünftige Ausgangskonfiguration notwendig. Dafür existiert ein Konfigurationsfile, welches über den von Java angebotenen Properties Mechanismus eingelesen und verwendet wird.

Derzeit sind folgende Werte einstellbar :

- Der Pfad, in welchem die Datenbank und der Index angelegt werden sollen
- Ein initiales Projekt
- Ein User, der auch Admin zum Projekt sowie Systemadmin ist
- Die Tiefe in der Framesets aufgelöst werden sollen
- Das Maximalalter, welches eine Dokument haben darf, bevor es vom Updater erneuert wird (in Tagen)

TeamFound Testserver

Browse projects | Login

- **Ossi-Teamfound** (#0) - Ein Projekt der TU-Berlin. Ziel ist eine Team-Suchmaschine.
- **Jans public project** (#1) - mit guestread und guesturladd
- **Jan's public read-only project** (#2) - nur lesen erlaubt

TeamFound - share your search results

Copyright (c) 2006 - Jan Kechel, Martin Klink, Jonas Heese, Andreas Bachmann
Using Server-Version: 0.3 / Interface-Version: 3

Abbildung 2.1: Web-Interface Eingangsseite

- Die Zeit in Minuten, die der Updatethread schlafen soll bevor er die nächste Aktion ausführt

2.2 Neuerungen am Web-Interface

Neuerungen ist gut, bisher gab es eigentlich noch kein Web-Interface :)

Die Version 0.2 des TeamFound Servers hatte noch die Option `want=xml|html`, über die man vom Server direkt eine im Browser darstellbare HTML-Seite anfordern konnte. Jedoch war dieser Teil des Servers zu grossen Teilen nicht implementiert.

In der aktuellen Version 0.3 setzen wir nur auf XML-Antworten, die dafür aber allesamt ein XSL-Transformations Stylesheet enthalten:

```
<?xml-stylesheet href="transform.xml" type="text/xsl"?>
```

Über diese XSLT Datei können alle gängigen Browser das vom Server gelieferte XML selber in HTML umwandeln und anzeigen.

2.2.1 Komponenten

Das gesamte Web-Interface besteht aus 3 Dateien:

transform.xml Transformiert unseren XML-Baum in HTML.

stylesheet.css Legt das Layout des erstellten HTMLs fest.

ui.js Enthält eine einzige JavaScript Funktion (`showhide(layer_ref)`) zum Anzeigen/Verstecken einzelner Bereiche der Webseite.

TeamFound Testserver

[Browse projects](#) | [jan's projects](#)

[logout](#)

I'm admin for:

- [Jans public project](#) - mit `guestread` und `guesturladd` ([view/edit permissions](#) / [manage users](#) / [manage categories & URLs](#))
- [Jan's public read-only project](#) - nur lesen erlaubt ([view/edit permissions](#) / [manage users](#) / [manage categories & URLs](#))
- [Jan's private project](#) - nur ausgewählte user dürfen hier lesen und urls adden ([view/edit permissions](#) / [manage users](#) / [manage categories & URLs](#))

Abbildung 2.2: Web-Interface Übersicht der eigenen Projekte

Die `transform.xsl` ist die einzige dieser Dateien die ich hier genauer beschreiben will. Sie besteht, wie jede XSLT Datei aus verschiedenen Templates, die jeweils bestimmte XML Unterbäume in HTML umwandeln. Als erstes kommt das `template match=/response` zum Tragen, da unsere XML-Bäume alle von dem tag `response` umschlossen werden. Es erstellt zuerst die allgemeinen HTML-Header, ruft dann der Reihe nach die 3 templates `menu`, `showcommandresults` und `pageselector` auf, und beendet seine Arbeit mit dem Ende der HTML-Seite incl. einem Link zur teamfound Webseite und einem Copyright Hinweis.

menu Erzeugt nur das Menü der Seite entsprechend dem Parameter `xsltpassthrough`.

showcommandresults Durchsucht den XML-Baum nach den verschiedenen Antworten der möglichen Befehle und zeigt die Ergebnisse entsprechend an.

pageselector Ruft entsprechend den `xsltpassthrough`-Parametern spezialisierte templates auf, die jeweils Seiten für die Bedienung der verschiedenen Server-Befehle mittels HTML-Formularen erzeugen.

Unterstützt werden alle im Interface Milestone 3 festgelegten Befehle. Den genauen Aufbau der Webseite schaut sich am besten jeder selber auf unserem Testserver <http://teamfound.dyndns.org:8080/tf/tf> an.

TeamFound Testserver

[Browse projects](#) | [jan's projects](#)

Project (#1): [Search](#) | [Add page](#) | [Browse categories](#)

3 of 3 results shown:

[Interface-Spezifikation - JonasSeinWiki](#)

Interface-Spezifikation Aus JonasSeinWiki Wechseln zu: Navigation, Suche << TeamFound
Inhaltsverzeichnis 1 Aufgabe 2 Protokoll 2.1 Für Milestone 1 - final 2.2 Für Milestone 2 - final 2.3
Milestone 3

<http://wiki.jonasheese.de/index.php/Interface-Spezifikation>

[TeamFound - share your search results](#)

[BerliOS Logo] - share your search results Overview - Status - Download & Installation - Web-Interface
- Screenshots - License - Architecture - Documentation - Interfaces & Clones - Development - Ch
<http://teamfound.berlios.de/>

Abbildung 2.3: Web-Interface Suchergebnisse und Projekt-Menü

2.3 Firefox/Flock-Toolbar

Nichts besonderes. Die Toolbar wurde nur an das Interface Milestone 3 angepasst. Die Projektverwaltung und Administration wird nicht unterstützt, sondern hier verweise ich freundlichst auf das Web-Interface :-)

2.4 Projekt-Webseite

Aktualisierte Version, angereichert mit Werbe-Text:

<http://teamfound.berlios.de>

TeamFound



- share your search results

[Overview](#) - [Status](#) - [Download & Installation](#) - [Web-Interface](#) - [Screenshots](#) - [License](#) -
[Architecture](#) - [Documentation](#) - [Interfaces & Clones](#) - [Development](#) - [Changelog](#)

Overview

TeamFound gives a team the capability to share search results without any usage-overhead. The toolbar (firefox and ie) can be used to mark interesting pages and full-text-search those while also showing normal search-engine results for the same keywords.

Status

Try our **new Testserver Milestone 3** at <http://teamfound.dyndns.org:8080/tf/tf>.

- register your own project
- create categories and subcategories
- add other users to your project
- no email required for registration
- It's completely free
- Install [teamfound-0.9-fx+fl.xpl](#) Firefox Extension (use 'Settings' for your own Project-ID)
- have fun

Abbildung 2.4: Projekt Webseite

2.5 Neuerungen an der Internet-Explorer-Toolbar

Die Arbeiten an der Toolbar teilen sich in zwei Bereiche Anpassungen und Erweiterungen. Die Toolbar besteht aus einer Sammlung von Views, der Controller Klasse und einer Menge von Datenklassen (Modell). Durch diese Trennung ist die Erweiterung recht gut möglich. Neue Objekte wie Projekte, Benutzer und Berechtigungen wurden zuerst als Datenobjekte implementiert und die dazugehörige Logik der Controller Klasse hinzugefügt. Wo notwendig wurden Views zur Visualisierung erstellt. Es ist abzusehen dass mit falls weitere Funktionalität hinzugefügt wird, ein Teil der Logik aus der Controller Klasse, in Service Klassen ausgelagert werden muss.

Ansonsten habe ich mit der Verwendung eines Testprogramms begonnen. Der Ansatz verdeutlicht dass es möglich ist die Funktionen der Toolbar auch ausserhalb des Internet Explorers zu nutzen. Somit könnte auch eine Integration in Entwicklungsumgebungen stattfinden (z.B. Visual Studio).

2.5.1 Anpassungen

Es war vollen allem notwendig die Suche so anzupassen dass Sie mit den Änderungen aus Milestone 3 klar kommt. Als erstes wären da die neu eingeführten Projekte zu nennen. Da ein Projekt eine Kategorie ist, ändert sich an der Suche direkt nichts. Es ergibt sich jedoch die Notwendigkeit eine Möglichkeit für die Projektauswahl und die Kategorienauswahl in Abhängigkeit davon zu gestalten.

Weiterhin musste die neue Dokumentstruktur unterstützt werden. Da die Toolbar die Suchergebnisse intern als Objekte verwaltet war der Aufwand in der Anpassung nicht so groß.

2.5.2 Erweiterungen

Wie bereits im Ablauf beschrieben, wurde der bisherige Einstellungsdialog ersetzt. Statt nur eine einzelne Serveradresse eingeben zu können, hat der User nun die Möglichkeit diverse Server zu verwalten. Zu jedem Server können jetzt Adresse, Benutzer und Passwort, die Projekte, deren Benutzer und die jeweiligen Kategorien und Berechtigungen verwaltet werden. Im wesentlichen entspricht die Funktionalität der des Web Interface.

Kapitel 3

Projektverlauf

3.1 Organisation und Kommunikation by Jonas

Das Teamfound-Projekt hat gleich mit zwei sehr starken Belastungen zu kämpfen, einerseits ist es ein OpenSource-Projekt und kann zwar auf ordentliche Werkzeuge wie Revisionsverwaltung und Mailinglisten zurückgreifen, lebt jedoch von lockerer Organisation und der weitgehend freiwilligen Mitarbeit aller Beteiligten, andererseits ist es primär immernoch ein Projekt an der Universität und hat damit mit den ganz normalen Problemen solcher Projekte zu kämpfen. Dies schliesst insbesondere den ganz eigenen Arbeitsablauf in derartigen Projekten ein, also mehr Aktivität, je näher der Abgabetermin für diesen Bericht rückt.

Anders als im letzten Semester gab es dieses mal keine regelmäßigen Treffen. Abgesehen von einem Starttreffen, bei dem vorallem Ideen für Erweiterungen und Bugs ausgetauscht wurden und einem Treffen kurz vor Schluss, in welchem die noch zahlreichen offenen Baustellen besprochen wurden, gab es kein Treffen von allen beteiligten. Benutzt wurde daher, wie auch im letzten Semester schon, die Mailingliste des Projektes (teamfound-development@lists.berlios.de), um sich auszutauschen und Ideen und vorgehensweisen zu besprechen. Weitere Werkzeuge, wie sie Berlios anbietet, darunter Feature- und Bugtracker, wurden wohl auch aufgrund des geringen nutzens für ein derart kleines Projekt noch nicht benutzt. Gegen Ende des Projektes sind auch die ungeplanten treffen im inoffiziellen Projekt-IRC-Channel (#teamfound-quakenet) häufiger und bekannt für deren Produktivität geworden.

3.1.1 Fazit: Probleme wachsender Projekte

Teamfound besteht mittlerweile aus einer Menge Code, so sind alleine im Server mittlerweile an die einhundert Klassen mit vielen tausend Zeilen Code implementiert. In diesem Semester sind ausserdem viele Funktionen hinzugekommen, so das sich die Aktionsimplementierungen im Server fast verdoppelt haben. Es zeigt sich nun das die bisherige Architektur kaum mehr geeignet ist um mit den Anforderungen klarzukommen, so sind die zentralen Punkte im Server, nämlich das Servlet, welches Requests entgegennimmt, validiert und an den Controller weiterleitet, sowie dieser Controller, der die eigentliche Nutzlast für die einzelnen Aktionen implementiert, völlig überlaufen. In Zukunft wird es also immer wichtiger effektive Strukturen für diese wachsenden Anforderungen zu schaffen und zu dokumentieren. Klare Datenschnittstellen sind notwendig und bisher kaum umgesetzt, die Kapselung von Funktionalität ist eine Aufgabe, welche die Architektur des Teamfound-Servers noch einmal deutlich verändern wird. Die Auslagerung von Code in einzelne Bibliotheken-gleiche Implementierungen, wie zB. mit dem Crawler geschehen, können eine Möglichkeit sein, besser wart- und testbaren Code zu schreiben und auch für eine sauberere Implementierung sorgen.

3.2 Projektverlauf by Jan

Während des Semesters hat sich eigentlich garnichts zu diesem Projekt getan. Selbst Steffen habe ich irgendwann im Juli 2006 gefragt ob wir uns nicht mal wieder Treffen sollten. Am 23. August 2006 war es dann soweit, und wir haben gemeinsam über mögliche Erweiterungen und neue Features diskutiert. Prinzipiell kam dabei aber nichts rum, was nicht schon lange (seit 15. Mai 2006) im wiki unter Milestone 3 eingetragen war. Die dort genannten Features für den Server entsprechen lustigerweise auch genau den jetzt implementierten:

<http://wiki.jonasheese.de/index.php/Milestones>

Server 17:05, 15. Mai 2006 Kechel (->Milestone 3 (SS06) - Planung)

- * XSL-Stylesheet anstelle von HTML-Antwort von Server
-> es wird immer in XML geantwortet
- * Web-Admin-Interface
- * Framebasierte Seiten unterstützen
- * User-Management
- * Zeichensätze beachten

* Automatisches Updated der eingetragenen Seiten (alle x tage oder so)

Inwieweit das mit den Zeichensätzen tatsächlich funktioniert weiss ich persönlich nicht, ich habe aber im Source-Code gesehen das Jonas irgendwas mit "encoding" und so gemacht hat.

Nach diesem Treffen passierte wiederum erstmal eine Woche lang gar nichts.

Ich begann dann am 2. September in unserem wiki das Interface für Milestone 3 (http://wiki.jonasheese.de/index.php/Interface.Milestone_3) zu spezifizieren. Dazu überlegten Martin und ich welche Funktionen denn für die obigen Features benötigt würden, welche Parameter sinnvoll seien und wie die XML-Antwort aussehen könnte. Leider hat Jonas's Server immernoch als Datum den 1. Januar 1970, daher ist die History dieses Dokuments nicht so eindeutig zurückzuverfolgen. Für sehr verlässlich halte ich aber die Angaben in unserem Logbuch (<http://wiki.jonasheese.de/index.php/Logbuch>). Jedenfalls ist diese *Spezifikation*, ausgedruckt auf DIN A4 Papier, 12 Seiten lang, und wurde im weiteren Projektverlauf von allen Projektseiten wohl am häufigsten benutzt und erweitert. Die einzig andere, für mich nützliche Seite im Wiki, war **Server Milestone 3**, da Martin hier die Datenbank Tabellen incl. der Spaltennamen und Fremdschlüssel aufgeführt hat.

Am 6. September war wieder ein gemeinsames Treffen, jedoch waren nur Steffen, Andreas und Ich anwesend. Nach etwa zwei Stunden palaber war auch das geschafft. Wir hatten zwar interessante Themen, jedoch nicht wirklich spezifisch über das TeamFound Projekt diskutiert.

Nach diesen Vorbereitungen war ich eine Woche im Ausland, mit dem Bewusstsein daß nur nach meiner Rückkehr nur noch eine einzige Woche Zeit verbleiben würde um überhaupt etwas zu implementieren. Dementsprechend vermisste ich auch sehr die nicht vorhandene Aktivität auf unserer Mailingliste. Nicht eine einzige Mail bzgl. TeamFound erreichte mich in dieser Zeit.

Zum Glück hatte sich doch etwas getan. Jedenfalls hatte Martin viel am DBLayer gearbeitet, und somit die Infrastruktur für unsere neuen Features stark vorangebracht. Auch Jonas arbeitete schon am Index und Crawler. Was genau da alles verändert wurde oder werden sollte weiss ich nicht, aber bis jetzt hatte ich auch noch keine einzige Zeile am Sever selber geschrieben und wusste dementsprechend wenig darüber.

Hand in Hand mit Martin begann ich dann am Server zu arbeiten, und ein Feature nach dem anderen zu Implementieren. Da wir Zimmernachbarn sind, war die Kommunikation sehr sehr einfach und produktiv. Ich musste einfach nur *rübergehen* und sagen "*Hey, schalt mal deinen Chat-Client an, ich hab da ne Frage.*" Über die folgenden Tage lernte ich so sehr schnell mich im Server-Code zurechtzufinden. Nebenbei erstellte ich noch das Web-

Interface und passte die Firefox-Toolbar an das neue Interface an (ca. eine Stunde FF-Toolbar). Das Web-Interface entwickelte sich genauso wie Martin und ich Funktionen im Server fertigstellten, und nutzten dieses auch immer sofort um die neuen Funktionen zu testen. Jonas hatte derweil den neuen Indexer und Crawler fertiggestellt. Nach ein paar Stunden Bugfixing von Martin war diese Komponente auch schon voll funktionsfähig (Jonas hatte anscheinend Probleme mit seinem Internetzugang, und konnte daher bis jetzt seine Änderungen nicht comitten. Es ergaben sich auch viele Merge-Konflikte durch diese Verzögerung..).

Gestern, zwei Tage vor Abgabetermin, hatten wir endlich alle Features, bis auf das Löschen von Kategorien und das automatische Updaten der indizierten Seiten, fertiggestellt, und ich konnte einen öffentlichen Testserver aufsetzen (<http://teamfound.dyndns.org:8080/tf/tf>). Im Chat klärten wir mit Andreas auch die letzten Interface-Fragen für seine IE Toolbar implementation. Bis dahin hatte ich null Ahnung ob Andreas überhaupt an seinem Teil des Projekts arbeitet. Wie sich herausstellte ist seine Toolbar jedoch inzwischen wesentlich weiter als *meine*. Immerhin half ich ihm einen Fehler beim Einloggen zu lösen, eine Funktion die in der FF-Toolbar noch völlig fehlt.

Danach erstellte ich ein neues Release v0.9 der FF-Toolbar auf berlios.de und passte unsere Projekt Web-Seite <http://teamfound.berlios.de> entsprechend an.

Heute, einen Tag vor Abgabe, hörte ich soeben von Martin, daß die Update-Funktionalität nun auch funktioniert (während ich selber diesen Text hier schreibe). Es sieht also so aus als würde alles *just in time* fertig werden :)

Als Fazit würde ich sagen, daß besonders die Interface Spezifikation sehr sehr Hilfreich war, um die zu implementierenden Features genau festzulegen. Auch die völlig von mir getrennt abgelaufene Entwicklung der Internet Explorer Toolbar wurde sicherlich nur dadurch möglich.

Innerhalb des Servers fehlten mir eben diese gemeinsamen Strukturen. Wie ich in der letzten Woche gelernt habe gibt es zwar auch dort viele feste Vereinbarungen, jedoch habe ich erst jetzt, nach einer Woche intensiver Beschäftigung damit, den groben Überblick darüber erhalten. Ich bin sicher daß Martin und Jonas diesen Überblick schon immer hatten, aber ich vermisste eine kurze Beschreibung z.B. im wiki dazu doch sehr.

Es fällt auch auf das keinerlei Coding-Styles vereinbart wurden. Es war mir nicht möglich Funktionsnamen aufgrund eines festgelegten Schemas zu erraten oder *schnell* zu finden. Meist hab ich einfach Martin gefragt, und er hat mir dann gesagt ob es die Funktion, die xyz tut, schon gibt, oder ob ich diese neu Implementieren muss.

Zum Schluss noch die Anmerkung, daß unser manuelles Logfile im wiki praktisch nur von mir alleine geupdated wurde. Ich hab zwar Martin immer wieder mal dazu überreden können, auch einzutragen was er denn so gemacht hat, aber das Interesse daran war wohl bei niemandem sehr hoch. Mir persönlich hat das Log besonders für diesen Projektbericht sehr geholfen, aber für die eigentliche Projektentwicklung war es nicht notwendig. Dennoch finde ich es selber sehr schön den Projektverlauf so zurückverfolgen zu können.

So, jetzt bin ich selber gespannt ob wir heute noch den Server v0.3 released bekommen.

Jan

3.3 Projektverlauf von Martin

Der Start des Projekts war einerseits gut, da wir alle 4 weiter an dem Projekt arbeiten wollten. Was bedeutet, dass es wohl vorher für keinen von uns schlecht gelaufen ist.

Leider begannen wir auch mit der halboffiziellen Übereinkunft, dass wir während der Vorlesungszeit alle keine Zeit auf das Projekt verwenden können. Andere Universitäre Leistungen und Prüfungen bzw. Arbeit oder ähnliches waren dringender. Ich war selbst damit sehr Einverstanden, da ich auch nicht gewusst hätte wann ich noch Zeit erübrigen kann.

Das bringt natürlich wenn es dann losgehen soll wieder Einarbeitungszeit mit sich. Hinzu kommt für mich, dass vieles was ich schon am ende des letzten Semester als TODO auf der Liste hatte immer noch unverändert ist. Eher ist die Liste länger geworden.

Trotzdem sehe ich den Verlauf des Projekts positiv. Bei unserem ersten gemeinsamen Treffen haben wir festgelegt welche Erweiterungen wir noch in Teamfound einbauen wollen. Jan hat ziemlich schnell ein Grundgerüst fürs Interface im Wiki angelegt. Die Interfacespezifikation war in diesem wie auch im vorigen Semester eines der wichtigsten Hilfsmittel. Ich konnte immer gut vergleichen was schon funktioniert und was noch fehlt. Ausserdem sind nötige Änderungen und Erweiterungen, die einer erarbeitet sofort für alle nachlesbar. Einen grossteil der nötigen Request/Response Kommunikation hatten wir beim ersten Treffen besprochen, allerdings war die Spezifikation am Anfang noch sehr rudimentär. Wir hatten erstmal nur die Aufrufparameter festgelegt und Jan hatte ein leicht verändertes XML-Grundgerüst für die Serverantwort entworfen.

Deshalb habe ich zu Beginn immer gleich für jede Funktion, die ich implementiert habe auch die Spezifikation um die Serverantwort erweitert. Das ist später leider aufgrund von Zeitdruck unter den Tisch gefallen.

Bis zum letzten Treffen vor Abgabe verlief die Arbeit ziemlich zäh. Danach dafür wirklich gut. Alle waren wieder oft im IRC-Channel. Die schnellen Absprachen auf diese Art boten neben dem Wiki aus meiner Sicht die beste Kommunikationsmöglichkeit während der Implementation. Ich habe gut davon profitiert, weil die anderen mir in vielen Belangen doch ein Stück voraus sind und helfen konnten. Die Mailingliste lief da eher schlecht, weil meist einfach keine direkte Antwort da ist und man schlechter diskutieren kann.

Um ein Fazit zu ziehen, würde ich sagen es ist ausser Startschwierigkeiten gut gelaufen und wir haben alles umgesetzt was wir uns vorgenommen hatten. Leider sind wir noch kein richtiges OpenSource Projekt, da wir den Abgabedruck brauchen um weiterzukommen anstatt von alleine den Antrieb zu entwickeln unser Projekt voranzutreiben.

3.4 Projektverlauf by Andreas

Ich habe mich erst spät in die Arbeit zum Release 3 von Teamfound eingebracht. Auch bei mir standen diverse Dinge an welche es verhinderten sich vorher mit der Thematik zu beschäftigen.

Ich habe versucht wie beim ersten Teil vorzugehen und mich aus der Serverimplementierung strikt rauszuhalten. Dies hat verschiedene Gründe. Zum einen finde ich sehr spannend gegen eine definierte Schnittstelle zu entwickeln. Insbesondere im Falle von Teamfound hat das hervorragend geklappt. Wie auch schon bei Milestone 1 und 2, waren die Schnittstellen sehr gut beschrieben und vor allem lieferten sie die erwarteten Resultate. Ich konnte also direkt die Erweiterungen an Hand der Beschreibungen vornehmen. Dies ist nichtmal bei kommerziellen API's selbstverständlich. Das einzige aufgetretene Problem, ließ sich durch ein Gespräch im IRC lösen, und war letztlich auf ein Fehler im Client zurückzuführen.

Zum anderen wird die Funktionalität auch auf der Clientseite umfangreicher. Hier war es mir wichtiger ein konsistentes Release zu erstellen. Insbesondere die Gui Logik ist deutlich komplexer geworden.

Problematisch war für mich zunächst die Frage wo man die neuen Funktionen aufhängt. Viele der Aspekte, welche in Milestone 3 definiert worden sind, haben erstmal nichts direkt mit der Suche zu tun. Hier waren zum die Verwaltungen von Benutzern und Projekten, welche jetzt in die Toolbar integriert werden mussten.

Ich habe dabei für einen Admin Dialogentschieden mit denen diese Funk-

tionen aufrufen werden können. Dieser ersetzt den bisherigen Einstellungsdialog.

3.4.1 Fazit: Begeistert von Qualität

Ich persönlich bin von der Qualität der geleisteten Arbeit begeistert. Es wurden in kürzester Zeit eine Menge Features realisiert. Dabei haben sich die Leute selten persönlich gesehen. Die gewählten Kommunikationsmittel wurden entsprechend der Projektsituation und dem Zweck gewählt. Das Wiki diente hierbei zum Festschreiben von Agreements während IRC für die Spontankommunikation genutzt wurde.

Allerdings bin ich der Meinung das die Qualität nicht nur auf die Art der Projektabwicklung der zurückzuführen ist. Alle Beteiligten verfügen bereits über Erfahrung in Projektarbeit, welche hier deutlich wurde. So waren die bestimmte Aspekte wie Aufgabenverteilung, korrekte Schnittstellendefinition und Dokumentation von vornherein selbstverständlich.