Arbeitsgruppe Software Engeneering

Verteilte Paarprogrammierung in Open Source Software Projekten

Eike Starkmann

starkman@inf.fu-berlin.de

Matrikelnummer: 3787331

Erstgutachter: Prof. Dr. Lutz Prechelt

Zweitgutachter:Prof. Dr. Carsten Schulte

30.03.2010

Selbstständigkeitserklärung

Ich versichere, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben. Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 30. März 2010

Eike Starkmann

Danksagung

Mein besonderer Dank gilt meinen Eltern, die mir mein Studium ermöglicht haben und meiner Freundin Heike Kanter die das alles mit macht.

Inhaltsverzeichnis

Abbildungsverzeichnis i

Tabellenverzeichnis ii

Glossar iii

Glossar . . . iii

Abkürzungsverzeichnis . . . v

1 Einleitung 1

2 Bezugsrahmen 3

2.1 Softwarekategorien . . . 3

2.1.1 Freie Software . . . 3

2.1.2 Open Source Software . . . 4

2.1.3 Software unter Copyleft . . . 5

2.1.4 Kommerzielle Software . . . 5

2.1.5 Proprietäre Software . . . 5

2.1.6 Lizenzen . . . 5

2.2 Open Source Software Projekte . . . 6

2.2.1 Die Kathedrale und der Basar . . . 6

2.2.2 Das soziale Umfeld von Open Source Projekten . . . 7

2.2.3 Motivation der Teilnehmer . . . 8

2.2.4 Erfolgsfaktoren von Open Source-Projekten . . . 9

2.2.5 Die Infrastruktur Open Source Projekten . . . 10

2.2.6 Entstehung von Open Source Projekten . . . 12

2.2.6.1 Gemeindebasierte Open Source Projekte 12
2.2.6.2 Gesponserte Open Source Projekte 12
2.3 Saros 12
2.3.1 Paarprogrammierung 13
2.3.2 Verteilte Paarprogrammierung 14
2.3.3 Verteilte Paarprogrammierung mit Saros 15

3 Fragestellung 19

4 Methodik 20
4.1 Die Forschungsmethode 20
4.1.1 Handlungsforschung 20
4.1.2 Qualitatives Experiment 21
4.1.3 Vorgehensweise 21
4.2 Innovationseinführung 24
4.2.1 Allgemeines 24
4.2.2 Auswahl der Projekte 26
4.2.2.1 Kriterien 26
4.2.2.2 Auswahl 27
4.2.3 Die Kontaktaufnahme 28
4.2.4 Kommunikationsmittel 28
4.3 Datenauswertung 31
4.4 Das Kategoriensystem 31
4.4.1 Technische Ablehnung 32
4.4.1.1 Die Kategorie *„Es gibt schon so etwas“* (ege) . . . 32
4.4.1.2 Die Kategorie *„Defekt“* (def) 33
4.4.1.3 Die Kategorie *„Abhängigkeit von einer Entwicklungsumgebung“*(abe) 33
4.4.2 Konzeptionelle Ablehnung 33
4.4.2.1 Die Kategorie *„Abneigung gegen Pair Programming“* (app) 34
4.4.3 Umfeld bezogene Ablehnung 34

4.4.3.1 Die Kategorie *„Fehlende Funktionalität“* (fft) . . . 34
4.4.3.2 Die Kategorie *„Muss jeder selber entscheiden“* (mje) 34
4.4.3.3 Die Kategorie *„Wenige Entwickler würden das benutzen“* (web) 35
4.4.3.4 Die Kategorie *„Zu große Einarbeitungszeit“* (zge) . 35
4.4.3.5 Die Kategorie *„Es wird schon so etwas benutzt“* (ewb) 35
4.4.3.6 Die Kategorien *„Produkt könnte nützlich sein, aber nicht für uns“ und „Konzept könnte nützlich sein, aber nicht für uns“* (pnu) (knu) 35
4.4.3.7 Die Kategorie *„Entwickler arbeiten zu asynchron“* (eaa) 35
4.4.3.8 Die Kategorie *„Entwickler arbeiten schon in einem Büro zusammen“* (eab) 35
4.4.4 Technische Akzeptanz 36
4.4.4.1 Die Kategorie *„Interesse/sinnvoller Vorschlag“* (isv) 36
4.4.4.2 Die Kategorie *„Anwendungsvorschlag“* (aws) 36
4.4.4.3 Die Kategorie *„Schon nach so etwas gesucht“* (sng) 36
4.4.4.4 Die Kategorie *„Gute Lösung (Saros getestet.)“* (gul) 36
4.4.5 Konzeptionelle Akzeptanz 37
4.4.5.1 Die Kategorie *„Produktivitätssteigerung“* (pts) . . . 37
4.4.6 Umfeld bezogene Akzeptanz 37
4.4.6.1 Die Kategorie *„Generell von Nutzen für das Projekt“* (gnp) 37
4.4.6.2 Die Kategorie *„Anwendungsvorschlag für das Projekt“* (afp) 37
4.4.7 Technische Nachfragen 38
4.4.8 Konzeptionelle Nachfragen 38

5 Projekte 39
5.1 TYPO3 40
5.1.1 Das Projekt 40
5.1.2 Geschichte 40

5.1.3 Das Team 41
5.1.4 Projektauswahl 43
5.1.4.1 Begründung 43
5.1.4.2 Kontaktaufnahme 44
5.1.5 Erste Iteration 45
5.1.5.1 Resultate 45
5.1.5.2 Fazit der ersten Iteration 54
5.1.5.3 Weiteres Vorgehen 57
5.1.6 Zweite Iteration 57
5.1.6.1 TYPO3 Security Team 58
5.1.6.2 Auswahl 59
5.1.6.3 Fazit der zweiten Iteration 60
5.2 Vuze 61
5.2.1 Das Projekt 61
5.2.2 Geschichte 61
5.2.3 Das Team 62
5.2.4 Projektauswahl 62
5.2.4.1 Begründung 62
5.2.4.2 Kontaktaufnahme 63
5.2.5 Resultate 63
5.2.6 Fazit 64
5.2.7 Weiteres Vorgehen 64
5.3 Freenet 64
5.3.1 Das Projekt 64
5.3.2 Geschichte 65
5.3.3 Das Team 65
5.3.4 Projektauswahl 65
5.3.4.1 Begründung 65
5.3.4.2 Kontaktaufnahme 66
5.3.5 Resultate 66
5.3.6 Fazit 69
5.3.7 Weiteres Vorgehen 70

5.4 Drupal 70
5.4.1 Das Projekt 70
5.4.2 Geschichte 70
5.4.3 Das Team 71
5.4.4 Projektauswahl 71
5.4.4.1 Begründung 71
5.4.4.2 Kontaktaufnahme 72
5.4.5 Erste Iteration 72
5.4.5.1 Resultate 72
5.4.5.2 Fazit der ersten Iteration 81
5.4.5.3 Weiteres Vorgehen 84
5.5 WordPress 84
5.5.1 Das Projekt 84
5.5.2 Geschichte 85
5.5.3 Das Team 85
5.5.4 Projektauswahl 86
5.5.4.1 Begründung 86
5.5.4.2 Kontaktaufnahme 86
5.5.5 Erste Iteration 86
5.5.5.1 Resultate 86
5.5.5.2 Fazit der ersten Iteration 88
5.6 JUnit 89
5.6.1 Das Projekt 89
5.6.2 Das Team 89
5.6.3 Projektauswahl 89
5.6.3.1 Begründung 89
5.6.3.2 Kontaktaufnahme 90
5.6.4 Resultate 90
5.6.5 Fazit 91
5.7 Eclipse-cs 92
5.7.1 Das Projekt 92
5.7.2 Das Team 92

5.7.3 Projektauswahl 92
5.7.3.1 Begründung 92
5.7.3.2 Kontaktaufnahme 93
5.7.4 Resultate 93
5.7.5 Fazit 93
5.8 ADempiere 94
5.8.1 Das Projekt 94
5.8.2 Geschichte 94
5.8.3 Das Team 94
5.8.4 Projektauswahl 95
5.8.4.1 Begründung 95
5.8.4.2 Kontaktaufnahme 95
5.8.5 Erste Iteration 97
5.8.5.1 Resultate 97
5.8.5.2 Fazit der ersten Iteration 104
5.8.5.3 Weiteres Vorgehen 107
5.8.6 Zweite Iteration 107
5.8.6.1 Resultate 108
5.8.6.2 Fazit der zweiten Iteration 108
5.8.6.3 Weiteres Vorgehen 109
5.8.7 Dritte Iteration 109
5.8.7.1 Resultate 109
5.8.7.2 Fazit der dritten Iteration 110
5.8.7.3 Weiteres Vorgehen 111
5.9 Openbravo 111
5.9.1 Das Projekt 111
5.9.2 Geschichte 111
5.9.3 Das Team 112
5.9.4 Projektauswahl 112
5.9.4.1 Begründung 112
5.9.4.2 Kontaktaufnahme 112
5.9.5 Resultate 113

5.9.6 Fazit . . . 113
5.10 OpenProj . . . 113
5.10.1 Das Projekt . . . 113
5.10.2 Geschichte . . . 114
5.10.3 Das Team . . . 114
5.10.4 Projektauswahl . . . 114
5.10.4.1 Begründung . . . 114
5.10.4.2 Kontaktaufnahme . . . 114
5.10.5 Resultate . . . 115
5.10.6 Fazit . . . 115
5.11 Tomcat . . . 115
5.11.1 Das Projekt . . . 115
5.11.2 Geschichte . . . 116
5.11.3 Das Team . . . 116
5.11.4 Projektauswahl . . . 117
5.11.4.1 Begründung . . . 117
5.11.4.2 Kontaktaufnahme . . . 117
5.11.5 Resultate . . . 117
5.11.6 Fazit . . . 117
5.12 Hibernate . . . 118
5.12.1 Das Projekt . . . 118
5.12.2 Geschichte . . . 118
5.12.3 Das Team . . . 119
5.12.4 Projektauswahl . . . 119
5.12.4.1 Begründung . . . 119
5.12.4.2 Kontaktaufnahme . . . 119
5.12.5 Resultate . . . 119
5.12.6 Fazit . . . 121
5.13 Spring . . . 121
5.13.1 Das Projekt . . . 121
5.13.2 Geschichte . . . 122
5.13.3 Das Team . . . 122

5.13.4 Projektauswahl 123
5.13.4.1 Begründung 123
5.13.4.2 Kontaktaufnahme 123
5.13.5 Resultate 123
5.13.6 Fazit 124

6 Ergebnisse 125
6.1 Forschungsfragen 125
6.2 Kontaktaufnahme 126
6.3 Projekte 128
6.3.1 Organisation 129
6.3.2 Eclipse Nutzung 130
6.3.3 Größe 130
6.3.4 Arbeitsverteilung 130
6.4 Saros 133
6.4.1 Defekte 135
6.4.2 Features 135
6.4.3 Unklarheiten 135
6.5 Bewertung des Untersuchungsplanes 136

7 Ausblick 138

A Anhang 140
A.1 Tagebuch 140
A.1.1 Samstag 15.08.09 140
A.1.2 Sonntag 16.08.09 144
A.1.3 Dienstag 18.08.09 145
A.1.4 Mittwoch 19.08.09 145
A.1.5 Samstag 20.08.09 147
A.1.6 Freitag 26.08.09 147
A.1.7 Montag 31.08.09 147
A.1.8 Dienstag 03.09.09 148
A.1.9 Mittwoch 04.09.09 149

A.1.10 Donnerstag 05.09.09 152
A.1.11 Dienstag 08.09.09 152
A.1.12 Mittwoch 23.09.09 155
A.1.13 Donnerstag 24.09.09 155
A.1.14 Dienstag 29.09.09 156
A.1.15 Mittwoch 30.09.09 158
A.1.16 Montag 05.10.09 159
A.1.17 Freitag 09.10.09 160
A.1.18 Mittwoch 14.10.09 161
A.1.19 Donnerstag 15.10.09 171
A.1.20 Samstag 17.10.09 176
A.1.21 Sonntag 18.10.09 176
A.1.22 Montag 19.10.09 177
A.1.23 Donnerstag 21.10.09 180
A.1.24 Mittwoch 03.11.09 181
A.1.25 Donnerstag 04.11.09 181
A.1.26 Freitag 05.11.09 183
A.1.27 Donnerstag 19.11.09 184
A.1.28 Montag 23.11.09 211
A.1.29 Mittwoch 25.11.09 211
A.1.30 Donnerstag 26.11.09 212
A.1.31 Sonntag 29.11.09 213
A.1.32 Montag 30.11.09 214
A.1.33 Dienstag 08.12.09 219
A.1.34 Montag 14.12.09 220
A.1.35 Dienstag 12.01.10 221
A.1.36 Mittwoch 12.01.10 223
A.2 Kategoriensystem 224

Literaturverzeichnis 234

Abbildungsverzeichnis

2.1 Verschiedene Softwarekategorien 4
2.2 Die Saros Roster Ansicht 15
2.3 Die Saros Shared Project Ansicht 16
2.4 Saros Annotationen 17

4.1 Methodisches Vorgehen 23

5.1 Die kontaktierten Projekte 39
5.2 Exemplarischer Diskussionsverlauf auf einer Mailingliste 40
5.3 Emailverkehr auf der Liste typo3-project-5_0-general: TY1, TY2, TY3, TY4, TY5 45
5.4 TYPO3v5: Verteilung der Commits über die Wochentage im Jahr 2009 48
5.5 TYPO3v5: Verteilung der Commits über die Uhrzeiten im Jahr 2009 49
5.6 Die im Jahr 2009 durchgeführten Commits der Typo3v5 Entwickler 50
5.7 Emailverkehr auf der Liste typo3-project-4-3: TY6 und TY7 50
5.8 Emailverkehr auf der Freenet-Mailingliste: FR1, FR2, FR3 66
5.9 Freenet: Die zehn häufigsten Commits 2009 68
5.10 Freenet: Verteilung der Commits über Uhrzeit und Wochentage . . 68
5.11 Freenet: Verteilung der Commits auf die Zeitzonen 69
5.12 Emailverkehr auf der Drupal Mailingiste: DR1, DR2, DR3, DR4, DR5, DR6, DR7, DR8 72
5.13 Die im Jahr 2009 durchgeführten Commits der Drupal Entwickler . 75
5.14 Drupal: Verteilung der Commits über die Uhrzeiten im Jahr 2009 . 76
5.15 Drupal: Verteilung der Commits über die Wochentage im Jahre 2009 77
5.16 Codezeilen pro Drupal Entwicker seit Mai 2000 84

5.17 Emailverkehr auf der WordPress Mailingiste: WP1, WP2 86
5.18 Emailverkehr auf der JUnit Mailingliste: DR1 90

Tabellenverzeichnis

5.1 Arbeitsverteilung TYPO3v5 48
5.2 Arbeitsverteilung Drupal 75

6.1 Überblick über die durchgeführten Kontaktmethodene 127
6.2 Akzeptanz und Ablehnung von gesponserten und gemeindebasierten Projekte 129
6.3 Übersicht der Ablehnungen in den jeweiligen Kategorien 131
6.4 Überblick über die kontaktierten Projekte 132
6.5 Übersicht über positive Aussagen und ihren Katagorien 134
6.6 Gefundene Defekte 135
6.7 Feature Anfragen 136
6.8 Übersicht über die Nachfragen 137

Glossar

Glossar

Committer

Als Committer werden nur die Entwickler bezeichnet, die Schreibrechte (Commitrechte) auf der verwendeten Versionsverwaltung haben.

Content Management System

Software zum Verwalten von Inhalten, meist mit Redaktionssystem und Rechteverwaltung.

Derived

Wird innerhalb von Eclipse als Flag für Dateien benutzt, welche nicht direkt editierbar sein sollen.

Fetchmail

Fetchmail ist eine Software zum Empfangen und Weiterleiten von Emails.

Framework

Ein Framework, engl. für 'Rahmenstruktur', ist ein Programmiergerüst, das in der Softwareentwicklung verwendet wird.

Gantt-Diagramm

Ein Gantt-Diagramm oder Balkendiagramm dient der Darstellung einer zeitlichen Abfolge von Ereignissen oder Aktivitäten. Erfunden wurde es von Henry L. Gantt und wird häufig im Projektmanagement eingesetzt.

Java EE

Java Platform, Enterprise Edition ist ein Industriestandard für Java Programme, welcher hauptsächlich im Web-Bereich Anwendung findet.

Linux Kernel

Linux ist ein Betriebssystemkern der als Open Source Projekt der im Jahre 1991 von Linus Torvalds initiiert wurde. Der Kern ist unter der freien GNU General Public License (GPL) veröffentlicht.

MVC

Model View Controller (MVC) ist ein Architekturmuster, bei dem die Software in die drei Einheiten *Datenmodell (engl. model)*, *Präsentation (engl. view)* und *Programmsteuerung (engl. controller)* unterteilt wird. Dadurch können die einzelnen Komponenten wiederverwendet und spätere Änderungen einfach durchgeführt werden.

Peer-Review

Peer-Review (dt. Begutachtung durch Kollegen) ist ein Verfahren zur Beurteilung von Projekten oder Software durch unabhängige Gutachter, 'Peers' (engl. für Kollegen, Fachleute), mit dem Ziel der Qualitätssicherung.

Persistenz

Persistenz bezeichnet die Möglichkeit, im Arbeitsspeicher vorhandene Objekte in Datenbanken oder Dateisystemen abzulegen.

PERT-Diagramm

Ein PERT-Diagramm stammt aus der Netzplantechnik und dient der Darstellung von Abhängigkeiten im zeitlichen Ablauf von Projekten.

Proxy

Ein Proxy ist eine Kommunikationsschnittstelle in einem Netzwerk und arbeitet als Vermittler, der Anfragen stellvertretend entgegennimmt und dann an den Anfragenden weiterreicht.

Trunk
Der Hauptentwicklungszweig einer Versionsverwaltung wird als Trunk bezeichnet.

Workaround
Unter einem Workaround versteht man die Umgehung eines Problems durch eine provisorische Hilfskonstruktion.

Abkürzungsverzeichnis

ASF	Apache Software Foundation.
CMS	Content Management System .
CPALv1	Common Public Attribution License 1.0.
CVS	Concurrent Versions System.
DPP	Verteilte Paarprogramierung, engl. distributed pair programming.
ECF	Eclipse Comunication Framework.
ERP	Planung der Unternehmensressourcen, engl. Enterprise Resource Planning.
GNU	**G**NU is **n**ot **U**nix.
GPL	GNU General Public License.
GPLv2	GNU General Public License Version 2.
GUI	Grafische Benutzeroberfläche, engl. Graphical User Interface.
HTTP	Hypertext Transfer Protocol.

IDE	Integrierte Entwicklungsumgebung, engl. integrated development environment.
IRC	Internet Relay Chat.
JDT	Java development Tools.
LGPL	GNU Lesser General Public License.
ORM	Objektrelationale Abbildung, engl. englisch object-relational mapping.
OSSP	Open Source Software Projekt.
PoS	Point of Sales.
PP	Paarprogrammierung, oder engl. pair programming.
VNC	Virtual Network Computing.

1 Einleitung

Paarprogrammierung ist eine Arbeitstechnik, die ein zentraler Bestandteil des *eXtreme Programming* ist und zu den agilen Entwicklungsmethoden für Software gehört. Dabei bearbeiten zwei Entwickler eine Aufgabe zusammen an einem Computer, wobei der eine die Tastatur und die Maus bedient (Driver) und der andere die produzierten Ergebnisse überwacht (Observer). Diese Rollen können nach Bedarf getauscht werden.
Die verteilte Paarprogrammierung ist eine besondere Form des Paarprogrammierens, bei der die Entwickler räumlich getrennt voneinander arbeiten. Um dies zu ermöglichen werden Werkzeuge benötigt, die eine virtuelle Arbeitsumgebung schaffen. So sollen die Entwickler, obwohl räumlich getrennt, dieselben Möglichkeiten haben, als wenn sie nebeneinander an einem Computer sitzen würden. Das Eclipse-Plugin Saros stellt ein solches Werkzeug dar.

Open Source Software Projekte sind Projekte, die quelloffene Software herstellen. Dadurch ist es möglich, den Quelltext der Software zu begutachten und gegebenenfalls zu verändern. Projekte dieser Art zeichnen sich meist durch einen besonderen Softwareentwicklungsprozess aus. Da die Mitglieder des Projektes moftmals räumlich weit verteilt arbeiten, kommt die Nutzung agiler Methoden wie beispielsweise die Paarprogrammierung nicht in Frage.

Die Frage, ob sich verteilte Paarprogramierung mit Hilfe von Saros für Open Source Projekte eignet, soll Gegenstand dieser Arbeit sein. So soll herausgefunden werden, ob und wie Saros in Open Source-Projekten einsetzbar ist. Dafür wird Saros in ausgesuchten Open Source Projekten bekannt gemacht und validiert. Zudem soll es durch diese Arbeit möglich sein, sowohl neue Features als auch

noch vorhandene Defekte zu entdecken und eine aktive Nutzergemeinde aufzubauen.

Im folgenden erläutere ich, was Open Source Software ist und wie Open Source Software-Projekte arbeiten und strukturiert sind. Danach werde ich das Werkzeug Saros einführen und seine Funktionalitäten erläutern. Anschließend stelle ich die gewählte Vorgehensweise vor, wie ich einerseits Saros in Projekte einzuführen gedenke, und andererseits, wie sich Erkenntnisse aus diesem Prozess ziehen lassen. Dazu gehört zunächst die Vorstellung meiner zugrunde liegenden Forschungsfragen sowie anschließend die Darlegung der Methodik, um auf diese Fragen eine Antwort zu finden.
Nach diesem theoretischen Teil werden die kontaktierten Projekte beschrieben und die Resultate dargeboten. Dies umfasst den größten Teil der Arbeit. Zum Abschluss werden die gesammelten Erkenntnisse zusammengefasst und ein Ausblick für nachfolgende Forschungsarbeiten formuliert.

2 Bezugsrahmen

Dieser Abschnitt stellt verschiedene Softwarekategorien vor und ordnet Open Source Software in dieses Umfeld ein. Desweiteren wird die Software *Saros* vorgestellt und in ihren Funktionalitäten erläutert. Dies stellt dann den Rahmen dar für eine mögliche Zusammenschau von Open Source Software und Saros, die im Fokus dieser Arbeit steht.

2.1 Softwarekategorien

Dieser Abschnitt soll einen kurzen Überblick der verschiedenen Softwarekategorien geben und ihre Zusammenhänge darstellen, wie dies in Abbildung 2.1 gezeigt wird. Ich gehe dabei im folgenden nur auf die für diese Arbeit relevanten Punkte ein.

2.1.1 Freie Software

Der Begriff Freie Software wurde von Richard Stallman geprägt. Nach seiner Definition ist Software dann frei, wenn der Benutzer die Freiheit hat,

- das Programm für jeden Zweck zu benutzen (Freiheit 0).
- zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann (Freiheit 1). Der Zugang zum Quellcode ist dafür Voraussetzung.
- Kopien weiterzuverbreiten, so dass man seinem Nächsten helfen kann (Freiheit 2).

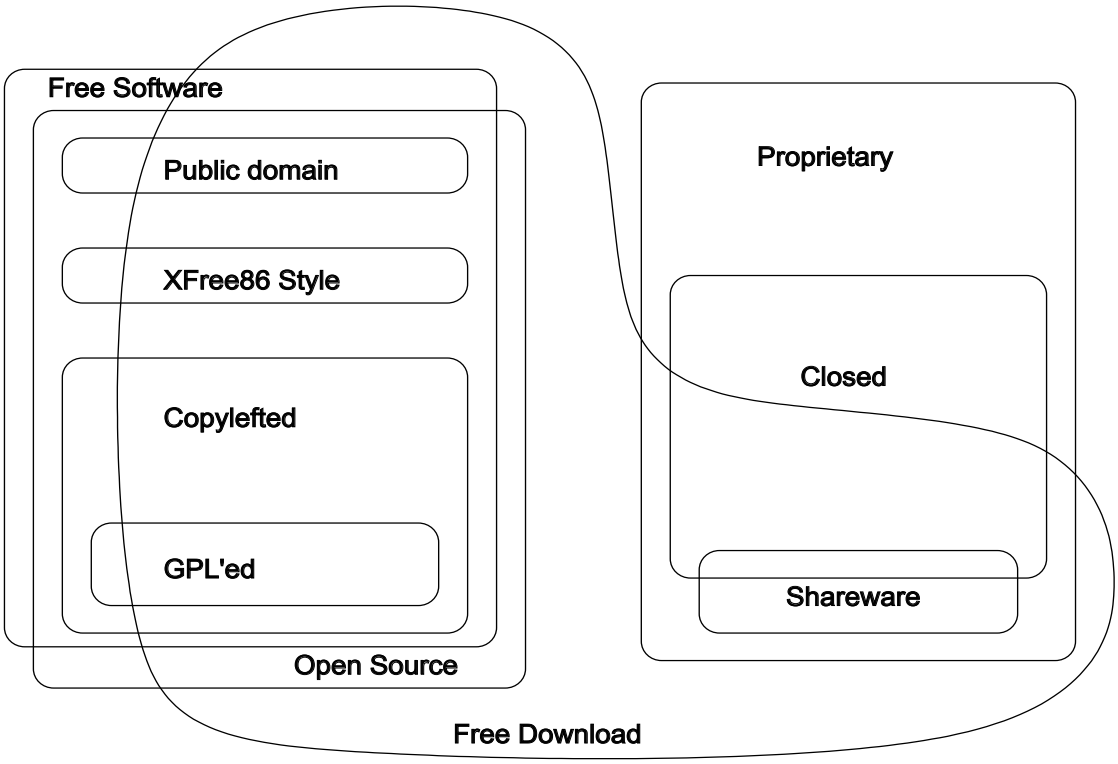

Abbildung 2.1: Verschiedene Softwarekategorien [Sta10a]

- das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen, damit die ganze Gemeinschaft davon profitieren kann (Freiheit 3). Bedingung dafür ist ebenfalls der Zugriff auf den Quellcode. [Sta10a]

2.1.2 Open Source Software

Im Jahre 1998 schlug Eric Raymond vor, den Begriff *Freie Software* durch *Open Source Software* zu ersetzen. Seiner Meinung nach brachte der Begriff „Frei“ zwei Probleme mit sich. Erstens ist der Begriff „frei“ zweideutig, d.h. er beschreibt, das etwas umsonst zu haben ist oder er kann so verstanden werden, wie Richard Stallman ihn verwendet. Das zweite Problem ist, dass Firmen sich von von dem Begriff „frei“ abschrecken lassen, oder wie Raymond es ausdrückt „the term makes a lot of corporate types nervous“ [Ray10]. Jedoch ist es für ihn wünschenswert, dass

auch Firmen Freie Software benutzen und evtl. an deren Entwicklung teilhaben. Heute ist „Open Source Software“ der gebräuchlichere Begriff.

2.1.3 Software unter Copyleft

Software unter Copyleft ist Software bei der jede erstellte Kopie, auch wenn diese verändert oder erweitert wurde wiederum Copyleft Software darstellt.

2.1.4 Kommerzielle Software

Üblicherweise entwickelt eine Firma kommerzielle Software, um damit Geld zu verdienen. Ein Großteil der produzierten kommerziellen Software ist proprietär, da die Implementierung einen Teil des Wertes der Software darstellt und so geschützt werden muss, um z.B. konkurrenzfähig zu bleiben. Es gibt aber auch einige Firmen die Freie Software herstellen. Auf diesen Punkt wird im Abschnitt 2.2.6.2 näher eingegangen.

2.1.5 Proprietäre Software

Proprietäre Software oder auch Closed Source Software genannt, ist das genaue Gegenteil von Freier Software. Einer Veränderung der Software, sofern überhaupt möglich, sowie dessen Weiterverbreitung ist verboten oder an sehr strenge Regeln geknüpft. Software dieser Art wird ohne Quelltext und meistens in Binärform ausgeliefert.

2.1.6 Lizenzen

Im Rahmen dieser Arbeit werden verschiedene Open Source Projekte untersucht. Diese Projekte sind zwar alle Open Source Software nach der obengenannten Definition, aber teilweise unter verschiedenen Lizenzen verfügbar. Auf jede dieser Lizenz im einzelnen einzugehen würde den Rahmen dieser Arbeit sprengen. So werden die verschieden Lizenzen der kontaktierten Projekte zwar genannt, aber nicht näher erläutert.

2.2 Open Source Software Projekte

Im weiteren Verlauf wird dargestellt wie Open Source Projekte organisiert sind und was ihren Entwicklungsprozess von klassischen unterscheidet. Ferner wird erläutert, wie und warum Open Source Projekte funktionieren.

2.2.1 Die Kathedrale und der Basar

Die Kathedrale und der Basar ist ein von Eric Raymond geschriebenes Essay über zwei verschiedene Stile der Softwareentwicklung. Er vergleicht dabei die klassische, kommerzielle Softwareentwicklung mit dem Bau einer Kathedrale und den von Linus Torvalds (Linux Kernel) verfolgten Open Source-Entwicklungsstil mit einem Basar. Aus dem Linux Projekt heraus entwickelt er Thesen, die er anschließend anhand einer Fallstudie zu seinem Projekt *Fetchmail*[1] belegt [Ray99].

Der Kathedralenstil

Spezialisten ihres Faches planen unter Ausschluss der Öffentlichkeit und auf dem Papier den Bau bzw. die Architektur der Kathedrale. Dabei wird so lange und möglichst genau geplant, dass keine Fehler beim eigentlichen Bau auftreten sollten. Denn Fehler sind um jeden Preis zu vermeiden. Dies führt zu einer langen Planungsphase und einer strengen, unflexiblen Hierarchie [Ray99].

Der Basar

Der Basar lebt von Selbstorganisation, es gibt kaum zentrale Planung außer einer Koordination der einzelnen Stände. Es ist nicht nur möglich, sondern ausdrücklich erwünscht, als Außenstehender am Basar teilzunehmen. Durch seine Vielfalt und Größe ist es dem Basar möglich auf äußere Änderungen schnell zu reagieren. Der Basar befindet sich im ständigen Weiterentwicklungsprozess [Ray99].

1 Fetchmail ist eine Software zum Empfangen und Weiterleiten von Emails

2.2.2 Das soziale Umfeld von Open Source Projekten

Ein Open Source Software Projekt besteht aus einem Zusammenschluss von Personen, die auf Basis freiwilliger Arbeit an einem Produkt bzw. Problem arbeiten. Dieser Zusammenschluss wird auch Gemeinde oder Community genannt; die folgenden Rollen sind dabei die wichtigsten:

Maintainer

Der Maintainer (engl.: *Unterstützer, Erhalter*) ist oft der Initiator des Projektes und leitet bzw. koordiniert das Projekt. Es gibt dabei unterschiedliche Formen, wie diese Leitung umgesetzt wird. Je nach Größe und Organisation des Projektes kann es auch mehrere Maintainer geben.
Nach Eric Raymond ist es für den Maintainer nicht unbedingt notwendig über herausragende Programmierfähigkeiten zu verfügen, sofern er gute Ideen und Lösungen anderer erkennen kann.

> „I think it is not critical that the coordinator be able to originate designs of exceptional brilliance, but it is absolutely critical that the coordinator be able to recognize good design ideas from others.“ [Ray99]

Es ist also wichtig für einen Maintainer gut mit Menschen umgehen und kommunizieren zu können. Dazu gehört auch, dass er Menschen begeistern kann und ihnen die durch seinen Entwicklungsstil die Möglichkeit gibt an dem Projekt teilzunehmen.

Entwickler/ Committer[1]

Die Entwickler eines OSS-Projektes haben meist die Möglichkeit den Quelltext des Programms zu verändern und in die aktuelle Entwicklungsversion einfließen zu lassen, sie werden deshalb zum Teil auch als Committer bezeichnet. In einigen Projekten kann dies aber nur der Maintainer.

1 Als Committer werden nur die Entwickler bezeichnet, die Schreibrechte (Commitrechte) auf der verwendeten Versionsverwaltung haben

Patchers

Patcher sind Personen mit ausreichendem technischen Wissen, um ein gefundenes Problem beheben zu können. Sie können ihre Lösung aber nicht aktiv in die Entwicklung einfließen lassen, da sie keine Commitrechte besitzen, sondern müssen ihre Änderungen in Form von Patches an die Entwickler oder den Maintainer weitergeben. Dafür ist meistens eine zentrale Verwaltung vorhanden, vgl. Abschnitt 2.2.5.

Bug-Reporter

Bug-Reporter sind Personen, die einen Defekt in der Software gefunden haben und diesen in einem Bericht (Bug-Report), den sie dem Projekt zukommen lassen, genauer beschreiben. Oft ist hierfür ebenfalls eine zentrale Verwaltung vorhanden, vgl. Abschnitt 2.2.5. Liefert der Bug-Reporter ebenfalls eine Lösungen des Defekts mit, wird er zum Patcher.

Darüber hinaus sind je nach Projekt weitere, unterschiedliche Rollen vorhanden, z.B. Übersetzer oder Webadministratoren. Die einzelnen Mitglieder müssen aber nicht immer ein bestimmtes Aufgabenfeld haben, es kann Überschneidungen geben. [Bra04, S. 6]

2.2.3 Motivation der Teilnehmer

Wie bereits erwähnt ist es wichtig, dass der Maintainer Menschen animieren kann an dem Projekt teilzunehmen. Allein mit Freundlichkeit und Überzeugung kann er dies sicherlich nicht schaffen. OSS-Projekte weisen einen Führungsstil und eine Struktur auf, die Menschen motiviert an dem Projekt teilhaben zu wollen. Es ist den Entwicklern möglich, sich einzubringen und dadurch Einfluss auf das Projekt nehmen zu können. Dabei steigt die soziale Anerkennung innerhalb des Projektes mit dem Ausmaß der Beteiligung. [Bra04, S. 25-26] Der Anreiz resultiert also aus der Einflussnahme auf das Projekt und das damit einhergehende Prestige. Eric Raymond bezeichnet dies als Pflege des jeweiligen Egos und der Reputation unter

Kollegen. So ist seiner Meinung nach der Egoismus der Teilnehmer ein zentraler Motivator. [Ray99]

> „ Aus diesem Grund ist es doppelt wichtig, daß Open Source-Hacker sich selbst organisieren, um durch Selbsternennung das Maximum an Produktivität zu liefern – und das soziale Milieu wählt gnadenlos nur die höchste Kompetenz.“ [Ray99]

Es gilt also Organisationsformen zu finden, in denen der eigene Egoismus der Allgemeinheit dient.

2.2.4 Erfolgsfaktoren von Open Source-Projekten

Eric Raymond formulierte insgesamt 19 Regeln, die er an dem Linux Kernel[1] Projekt und seinem eigenen (Fetchmail) aufgestellt hat. [Ray99] Alle diese Regeln lassen sich seiner Meinung nach mit dem Basarstil realisieren. Im folgenden werden die, meiner Meinung nach, wichtigsten Regeln vorgestellt.

- Die Benutzer von Open Source Software sollten an dem Projekt teilhaben können und es sollte ihnen die Möglichkeit gegeben werden, Änderungen und Verbesserungen an die Entwickler weiterzugeben. Dabei sind Testresultate ebenso wertvoll wie Quelltext. Die Entwickler wiederum sollten Anregungen und Änderungen von den Benutzern begrüßen und sie in das Projekt integrieren.
  „6. Trating your users as co-developers is your least-hassel route to rapid code improvement and effective debugging“
  „10. If you treat your beta-testers as if they’re your most valuable resource, they will respond by becoming your most valuable resource.“
  „11. The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.“ [Ray99]

1 Linux ist ein Betriebssystemkern der als Open Source Projekt der im Jahre 1991 von Linus Torvalds initiiert wurde. Der Kern ist unter der freien GNU General Public License (GPL) veröffentlicht

- Um Anregungen und Änderungen schnell in die Software integrieren zu können, sollte diese regelmäßig und früh veröffentlicht werden. Dabei kann unstabile Software in Kauf genommen werden.
  „7.Relesae early. release often. And listen to your customers.“ [Ray99]

- Defekte in der Software müssen nicht zwangsläufig von den Entwicklern gefunden werden, die Benutzer können dies ebenso leisten. Je mehr Benutzer zur Verfügung stehen, desto wahrscheinlicher ist es, dass alle Defekte gefunden werden. Diese Regel wurde von Eric Raymond *„Linus Law“* getauft.
  „8. Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.“ [Ray99]

- Sind die passenden Rahmenbedingungen gegeben, arbeiten viele Personen besser als nur einer. Diese Faktoren betreffen den Projektkoordinator, als auch die Infrastruktur sowie das soziale Umfeld des Projektes. Auf diese Punkte wird in Abschnitt 2.2.5 genauer eingegangen.
  „19. Provided the development coordinator has a communications medium at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one.“[Ray99]

2.2.5 Die Infrastruktur Open Source Projekten

Die Mitglieder eines Open Source Projektes kommunizieren hauptsächlich über elektronische Kanäle, wie z.B. Emails oder Foren. Daher ist der Zugang zum Internet die Basis eines jeden Open Source Projektes. Personen ohne Internetzugang können kaum an einem OSS-Projekt teilnehmen [Bra04, S. 9].

Mailingliste

Die typische Form der Kommunikation ist die Mailingliste. Wird eine Email an eine Mailingliste geschickt, so wird diese an alle Listenteilnnehmer weitergeleitet. Üblicherweise ist dies nur möglich, wenn der Sender ebenfalls Mitglied der Liste ist. Bei vielen OSS-Projekten ist diese Mailingliste öffentlich über Archive einsehbar und jeder hat die Möglichkeit sich in die Liste einzutragen. Oft sind mehrere Listen

vorhanden, die dann thematisch eingegrenzt sind. Es gibt aber auch OSS-Projekte, welche die eigentliche Entwicklungsmailingliste nicht öffentlich zugänglich machen, vgl. Abschnitt 5.2.4.2.

Versionsverwaltung

Der Quelltext der Software wird normalerweise von einer Versionsverwaltung verwaltet. Diese Software macht es möglich, jede durchgeführte Veränderung rückgängig zu machen bzw. sich Änderungen anzeigen zu lassen. Außerdem erleichtert sie das Zusammenenarbeiten von mehreren Entwicklern an einem Quelltext, da Konflikte von unterschiedlichen Versionen angezeigt und gelöst werden können. Welcher Entwickler Schreibrechte hat lässt sich ebenfalls mit dieser Software regulieren. Häufig benutzte Versionsverwaltungen sind CVS[1], SVN[2] und GIT[3].

Bugtracker

Ein Bugtracker (engl. Verfolger) verwaltet und organisiert Defekt-Beschreibungen einer Software. In Open Source Projekten sind sie üblicherweise öffentlich einsehbar und es ist möglich Einträge zu verfassen. Es gibt noch andere Arten dieser Tracker, die z.B. für Feature-Anfragen benutzt werden.

Mittlerweile gibt es Webseiten, die OSS-Projekten diese Infrastruktur anbieten. Jeder kann dort ein Projekt anlegen und es stehen u.a. die oben genannten Möglichkeiten zur Verfügung. Dieser Service wird von vielen OSS-Projekten benutzt, eine bekannte Webseite dieser Art ist *www.sourceforge.net*, die für die Projektauswahl in dieser Arbeit von Bedeutung ist.

1 http://www.nongnu.org/cvs/
2 http://subversion.apache.org/
3 http://git-scm.com/

2.2.6 Entstehung von Open Source Projekten

2.2.6.1 Gemeindebasierte Open Source Projekte

Viele Open Souce Software Projekte wurden von einer Person oder einer kleinen Gruppe initiiert. Der Anfang liegt oft in einem Problem, das der Initiator zu lösen versucht, vgl. [Ray99]. Diese Lösung oder ein Teil dessen wird dann als Open Source Software zur Verfügung gestellt und andere Personen, die ein ähnliches Problem haben können diese Lösung benutzen bzw. ihren Bedürfnissen anpassen. Je nach Problem und Fortschritt der Software, die zur Verfügung gestellt wird, kann sich dann eine Gemeinde um das Projekt bilden. Es muss sich aber nicht immer um ein konkretes Problem handeln, so kann z.B. ein Grund für eine Projektgründung sein, dass es die zu entwickelnde Software noch nicht als Open Source Software gibt.
Projekte dieser Art werden von einer Gemeinde getragen und schützen sich so davor von Firmen oder anderen Organisationen übernommen zu werden [WO05, S. 1]. Ich nenne diese Projekte im folgenden *gemeindebasierte Open Source Projekte*.

2.2.6.2 Gesponserte Open Source Projekte

Bei gesponserten OSS-Projekten wird die Software nicht öffentlich von einem Sponsor entwickelt. Zur Weiterentwicklung wird diese Open Source Software zur Verfügung gestellt. Der Sponsor kann dabei z.B. eine Firma oder Regierungsstelle sein. Die Verbreitung der Software als OSS kann verschieden Gründe haben. Zum einen kann damit eine weitere Verbreitung der Software erreicht werden. Zum anderen wird so eine Software, welche zwar benötigt wird, aber nicht den Kernbereich des Sponsors darstellt, kostengünstig weiterentwickelt. [WO05, S. 3]. Der Sponsor versucht meistens die Kontrolle über das Projekt zu behalten.

2.3 Saros

Saros ist ein Plugin für die Entwicklungsumgebung *Eclipse* und wird als Open Source Projekt von der Arbeitsgruppe Software Engineering an der Freien Universität

Berlin entwickelt. Das Plugin ermöglicht verteilte, kollaborative Softwareentwicklung sowie verteilte Paarprogrammierung. Dabei wird der Editor von Eclipse mit verschiedenen Personen über das Internet verbunden und Änderungen werden in Echtzeit übertragen. Entstanden ist Saros im Rahmen einer Diplomarbeit im Jahre 2006 [Dje06] und wurde seitdem stetig weiterentwickelt.

2.3.1 Paarprogrammierung

Paar Programmierung (PP) bedeutetet, dass zwei Entwickler eine Aufgabe zusammen an einem Computer erledigen, wobei der eine die Tastatur und die Maus bedient (Driver) und der andere die produzierten Ergebnisse überwacht (Observer). Diese Rollen können nach Bedarf getauscht werden sollte dies nötig sein. Paar Programmierung ist ein Bestandteil des *Extreme Programming* und gehört so zu den agilen Entwicklungsmethoden [Bec99]

.

Dieses Verfahren liefert einige Vorteile für den Entwicklungsprozess:

- Wissensverteilung
  Wissen, z.B. über ein bestimmtes Modul, wird über mehrere Personen verteilt. Dadurch werden Wissensmonopole abgebaut und personelle Ausfälle sind weniger problematisch[CW01].

- Teambildung
  Die Projektteilnehmer lernen sich gegenseitig (schneller) kennen, wodurch die Zusammenarbeit verbessert werden kann und die Community zusammen wächst[CW01].

- Entwurfsqualität
  Verschiedene Herangehensweisen und Erfahrungen der Paare führen zu besseren Entwürfen und kürzerem Programmcode[Wil00], [CW01].

- Zufriedenheit
  Viele Entwickler finden die Zusammenarbeit angenehmer als alleine zu arbeiten[CW01].

- Durchsichten
  Kontinuierliche Durchsichten führen zu einer geringeren Defektrate sowohl im Code als auch im Design[Wil00], [CW01].

- Lernen
  Jeder hat unterschiedliches Wissen und dieses wird durch die Zusammenarbeit geteilt, so dass jeder davon profitiert. Neueinsteiger können so schnell angelernt werden. Durch die permanente Diskussion über die benutzen Lösungsstrategien, findet ebenfalls ein permanente Weiterbildung statt.[Wil00], [CW01].

- Lesbarkeit
  Die Lesbarkeit des Codes wird verbessert [Nos98].

2.3.2 Verteilte Paarprogrammierung

Verteilte Paarprogrammierung (engl. Distributed Pair Progamming) (DPP) ist eine spezielle Form des Paarprogrammierens, hierbei sind die beiden Entwickler räumlich voneinander getrennt. Mit Hilfe von Werkzeugen wird es ihnen ermöglicht eine virtuelle Arbeitsumgebung einzurichten. Diese sollten den Entwicklern ähnliche Möglichkeiten bieten, wie wenn sie nebeneinander an einem Computer sitzen würden.

1. Ein Entwickler muss die Möglichkeit haben etwas schreiben zu können, was der Andere unmittelbar lesen kann.

2. Es muss die Möglichkeit bestehen, diese Rollen tauschen zu können oder beide müssen gleichzeitig schreiben können.

3. Darüber hinaus muss es möglich sein, verbal kommunizieren zu können.

Nonverbale Kommunikation, z.B. das Zeigen auf ein bestimmte Stelle auf dem Monitor kann mit DPP aufgrund der räumlichen Trennung nur schwer erreicht werden, wodurch die verbale Kommunikation noch wichtiger wird.

2.3.3 Verteilte Paarprogrammierung mit Saros

Bis auf die verbale Kommunikation, für die Skype, Mumble oder einfach Telefon empfohlen wird, deckt Saros alle genannten Möglichkeiten des DPP ab. Wie dies geschieht wird im folgenden erläutert.

Saros benutzt *XMPP*[1] zur Kommunikation und kann so mit einem Jabber[2] oder Google Talk[3] Account benutzt werden. Der sogenannte *Roster* ist dafür zuständig, die Kontakte zu verwalten.

Roster

Wie in Abildung 2.2 zu sehen ist, sind im *Roster* alle vorhandenen Kontakte und ihr Status zu sehen. Es wird angezeigt welche Kontakte gerade online sind und ob sie gerade in Eclipse arbeiten. Die Roster Anzeige ist der Einstieg in Saros, d.h bevor eine Sitzung aufgebaut werden kann muss zumindest ein Kontakt vorhanden sein. Eine neuen Kontakt kann ebenfalls in dieser Ansicht hinzugefügt werden.

Abbildung 2.2: Die Saros Roster Ansicht

Indem der Benutzer die Anderen einlädt, eröffnet er die Sitzung und wird deswegen Host genannt. Zum Aufbau einer Sitzung wählt dieser ein Projekt in Eclipse aus an dem gearbeitet werden soll. Dabei stehen ihm zwei verschiedene Möglichkeiten

1 http://xmpp.org/
2 http://www.jabber.org/
3 http://www.google.com/talk/

zur Verfügung: *Share project* und *Share project partially*.
Die erste Möglichkeit stellt dem Eingeladenen (Client) das ausgewählte gesamte Projekt zur Verfügung und dieser kann wählen, ob er es in ein Existierendes integrieren will, z.B. von der letzten Sitzung oder ob er ein neues Projekt erstellen will. Dabei ist zu beachten, das existierende Projekte immer mit dem Host synchronisiert werden, d.h. es besteht die Möglichkeit gemachte Änderungen auf der Clientseite zu verlieren.
Die zweite Methode, das Partial Sharing - im folgenden auch Partielles Verteilen genannt - bietet dem Host die Möglichkeit dem Client nur bestimmte Teile oder Dateien des Projektes anzubieten. Diese Methode ist allerdings noch experimentell.
Wurde eine Sitzung erstellt, verfügen alle Teilnehmer über ein synchrones Projekt und das *Shared Projekt View* Fenster wird angezeigt.

Shared Project View

Im Shared Project View Fenster werden alle Sitzungsteilnehmer und ihre dazugehörigen Farben angezeigt. Die Farben werden benutzt, um im Editor die verschiedenen Aktivitäten des jeweiligen Benutzer anzuzeigen.

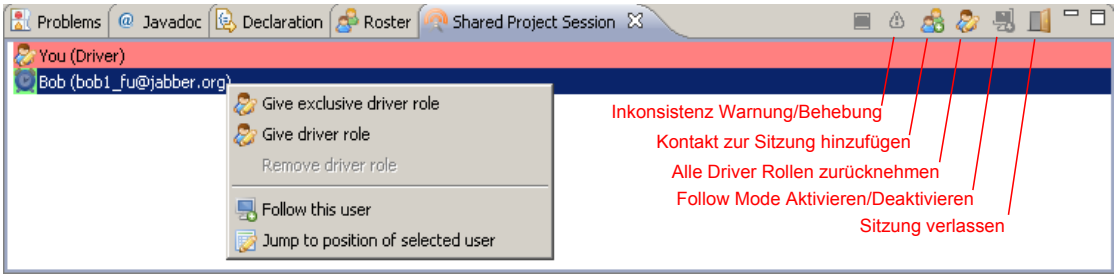

Abbildung 2.3: Die Saros Shared Project Ansicht

Die Standardeinstellung von Saros ist so gewählt, dass der Host nach einer aufgebauten Sitzung der Driver ist und alle anderen Teilnehmer Observer sind. Dies hat den Vorteil das zunächst nur der Host Änderungen durchführen kann. Dies kann jedoch in der Shared Project View vom Host geändert werden, vgl. Abbildung 2.3.

Der Host hat die Möglichkeiten einem Teilnehmer exklusive Driver-Rechte zu geben, das heißt der Host ist nun nicht länger Driver und übergibt diese Rolle an den Teilnehmer. Ferner hat er die Möglichkeit einen zweiten oder mehrere Driver hinzuzufügen, die dann ebenfalls Änderungen in den Dateien vornehmen können. Diese Option wird *Multi Dirver* genannt und kann in den Saros Einstellungen auch deaktiviert werden.

Während des Multi Driver Modus kann es zu Inkonsistenzen kommen. Diese werden in der Shared Project Ansicht angezeigt und können dort auch behoben werden. Der Host hat zu jeder Zeit die Möglichkeit, alle vergebenen Rollen zurückzunehmen und zusätzliche Benutzer, die online sind, zu der laufenden Sitzung hinzuzufügen.

Dem Client steht zur Wahl, ob er dem Driver folgen möchte (Follow Mode) oder selbst in den Dateien navigieren möchte. Bei dem Follow Mode (Verfolger Modus) werden die Aktivitäten des Drivers verfolgt, dass heißt der Client Editor spring immer an exakt die Stelle und Datei, die der Driver gerade bearbeitet.

Annotationen

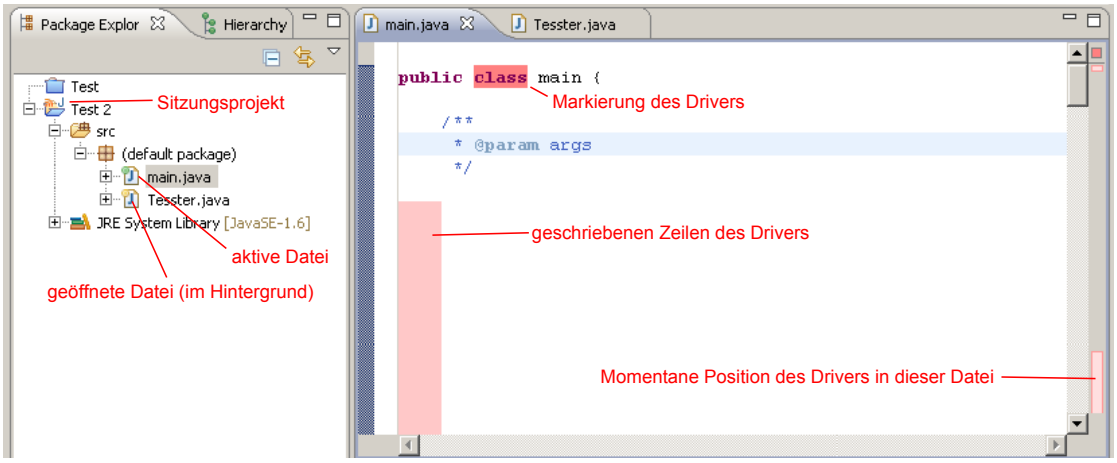

Abbildung 2.4: Saros Annotationen

Annotationen bieten dem Client die Möglichkeit nachzuvollziehen, wo sich der

Driver befindet. Dies ist besonderes nützlich, wenn der Client sich nicht im Verfolger-Modus befindet. Zu diesem Zweck werden die aktuell vom Driver geöffneten Dateien farblich markiert, die momentan Bearbeitete wird gesondert markiert.
Wo sich der Driver in der bearbeiteten Datei befindet, wird in der Farbe des Drivers an der Seite des Editors markiert, dies ist bei größeren Dateien besonders nützlich. Wie in Abbildung 2.4 zu sehen ist, werden die hinzugefügten Zeilen in der Farbe des jeweiligen Benutzer hinterlegt. Markierungen der Benutzer werden ebenfalls in der entsprechenden Farbe angezeigt.

Nach der Vorstellung der Besonderheit von Open Source Projekten im Gegensatz zur kommerziellen Software sowie dem Werkzeug Saros zur verteilten Paarprogrammierung, lege ich nun meine Forschungsfragen dar.

3 Fragestellung

Es soll herausgefunden werden, ob Saros in Open Source Projekten einsetzbar ist. Dafür wird Saros in ausgesuchten Open Source Projekten bekannt gemacht und validiert. Dadurch erhoffe ich mir sowohl neue Anwendungsmöglichkeiten als auch noch vorhandene Defekte von Saros herauszufinden und eine aktive Nutzergemeinde aufzubauen. Dafür wird es nötig sein, eine Strategie zur Einführung von Saros in OSS-Projekten zu erstellen und anschließend zu bewerten. Zudem werden die Projekte analysiert und ihre Struktur in Bezug auf eine erfolgreiche, bzw. nicht erfolgreiche Kontaktaufnahme eingestuft.
Die gewonnenen Erkenntnisse sollen helfen, Saros für den Einsatz in OSS Projekten zu verbessern und die benutzte Kontaktstrategie zu validieren. Zentrale Forschungsfragen sind also:

1. Wie ist Saros in OSS-Projekten einsetzbar?
   a) Wie wird Saros benutzt?
   b) In welchen Kontext wird Saros benutzt?
2. Wie kann Saros in OSS-Projekten eingeführt werden?
3. Welche Folgen ergeben sich aus dem Einsatz von Saros in OSS Projekten?

4 Methodik

Um den Prozess der Datengewinnung und -auswertung transparent und nachvollziehbar zu halten, beschreibe ich an dieser Stelle meine verwendete Forschungsmethodik.

4.1 Die Forschungsmethode

Die Ausgangsbasis für die Erforschung des Einsatzes von Saros im Open Source-Bereich stellt die Kontaktaufnahme zu den Projekten dar. Saros soll damit in ausgesuchten OSS-Projekten bekannt gemacht und anschließend validiert werden. Dafür kommen v.a. zwei methodische Ansätze in Frage. Ich werde nun kurz umreißen, was die einzelnen Methoden auszeichnet und im darauffolgenden Abschnitt darstellen, wie ich sie für meine Herangehensweise fruchtbar mache.

4.1.1 Handlungsforschung

Das Konzept der Handlungsforschung stammt ursprünglich aus den Erziehungswissenschaften. Die Forschung soll an konkreten Problemen ansetzen und ihre Ergebnisse bereits im Forschungsprozess in die Praxis umsetzen. [Gun86] Zu Beginn wird das Problem, das untersucht und gelöst werden soll, definiert. Hierbei geschieht „die Problemauswahl und –definition nicht vorrangig aus dem Kontext wissenschaftlicher Erkenntnis, sondern entsprechend konkreten gesellschaftlichen Bedürfnissen“ [Sta10b]. Das methodische Vorgehen ist ein iteratives, es wird sich dem Problem schrittweise angenähert. Um einen Diskurs zur Lösung des Problems mit den betroffenen Personen zu etablieren, müssen der Forscher und die Personen gleichberechtigt sein.

4.1.2 Qualitatives Experiment

Neben der Handlungsforschung erschien mir das qualitative Experiment als ein guter Ansatz für die Beantwortung meiner Fragen. „Das qualitative Experiment ist der nach wissenschaftlichen Regeln vorgenommene Eingriff in einen (sozialen) Gegenstand zur Erforschung seiner Struktur.“ [Kle82] Demnach stehen bei dem qualitativen Experiment nicht die Überprüfung vorgefertigter Hypothesen im Vordergrund, sondern das Aufdecken von Strukturen des zu untersuchenden Gegenstandes. Hierzu ist es möglich, immer wieder in veränderter Form einzugreifen und so in einen Dialog mit dem Gegenstand zu kommen. [May02, S. 59]

4.1.3 Vorgehensweise

Die Frage *wie Saros in Open Source Projekten eingeführt werden kann*, soll sich nicht damit begnügen, die Resultate der Kontaktaufnahme zu den verschiedenen Projekten aufzuzeichnen. Vielmehr soll die Kontaktaufnahme gleichzeitig evaluiert und ggf. verbessert werden, um den Erfolg bei den jeweils anschließenden Kontaktaufnahmen zu verbessern. Dies gleicht dann einer auf Saros eingeschränkten Innovationseinführung, worauf ich im späteren Abschnitt noch eingehen werde (4.2). Um die Kontaktaufnahme zu überprüfen und Rückschlüsse auf die Struktur von geeigneten bzw. nicht geeigneten OSS Projekten zu erhalten, bietet sich auf den ersten Blick das *qualitatives Experiment* an, vlg. [Roß07].
Bei den beiden weitergehenden Fragen "*Welche Folgen ergeben sich aus dem Einsatz von Saros in OSS Projekten*und "*Wie ist Saros in OSS Projekten einsetzbar?*", ist zu beachten, dass ich bereits durch die Kontaktaufnahme und meinem Anliegen Einfluss auf das Projekt nehme.
Die Frage nach den Auswirkungen des Einsatzes von Saros in Open Source Projekten beinhaltet aber auch umgekehrt die Frage nach den Konsequenzen, die sich wiederum für Saros durch den Einsatz in OSS Projekten ergeben. Um dies beantworten zu können, schlage ich hier ein iteratives Vorgehen vor. Bei diesem wird Saros vorgestellt, die Projektteilnehmer empfehlen Verbesserungen oder Änderungen, welche dann in Saros umgesetzt werden müssen, um Saros wieder von vorne an dem Projekt zu evaluieren. Dabei lassen sich natürlich auch strukturelle

Änderungen des Projektes erkennen, sollten diese durch den Einsatz von Saros auftreten. Wichtig ist vor allem der Dialog mit den Projektteilnehmern.
Die *Handlungsforschung* kommt somit diesem Vorgehen sehr nahe, wenn auch - wie schon erwähnt - diese sich v.a. darauf bezieht, ein in einer Personengruppe existierendes Problem gemeinsam zu untersuchen bzw. zu lösen. Im Falle von Saros existiert das Problem „Einsatz von Saros“ innerhalb der Open Source Projekte eigentlich nicht. So kann natürlich hypothetisch definiert werden, das dieses Problem die momentan noch fehlenden Möglichkeiten sind, die ein Einsatz von Saros bieten würde, oder die Projekte ein Bedürfnis nach kollaborativer Zusammenarbeit haben. Anlass dieser Arbeit ist aber ebenfalls zu untersuchen, ob der Einsatz von Saros für OSS Projekte überhaupt von Nutzen ist. So wird in die Projekte aktiv eingegriffen. Ein gleichberechtigter Dialog vonnöten ist, um Saros in das Projekt einzuführen und ggf. den Bedürfnissen des Projektes anzupassen, als auch neue Anwendungsmöglichkeiten von Saros zu entdecken.
Mein Vorgehen stelle ich mir wie folgt vor Abb. 4.1 .

1. Projekt auswählen
   Es hierbei soll iterativ herausgefunden werden, welche Struktur Projekte haben die sich für eine Kontaktaufnahme und Einführung von Saros eignen. Nach jeder Kontaktaufnahme müssen die ausschlaggebenden Kriterien des Projektes, vgl. Abschnitt 4.2.2, festgehalten und bei der Auswahl des nächsten Projektes berücksichtigt werden.

2. Projekt anschreiben
   Hierbei werden verschiedene Kontaktstrategien untersucht. Diese können mit der Struktur der Projekte korrelieren.

3. Diskussion
   Nun folgt der von der Handlungsforschung vorgeschlagene gleichberechtigte Dialog zwischen Forscher und Projektteilnehmer. Von diesem Dialog erhoffe ich mir Erkenntnisse, darüber, *wie Saros in OSS Projekten einsetzbar ist.* Hier soll auch klar werden, ob die Kontaktstrategie und das Projekt gut gewählt wurden. Durch die offene und gleichberechtigte Art der Diskussion erhoffe ich

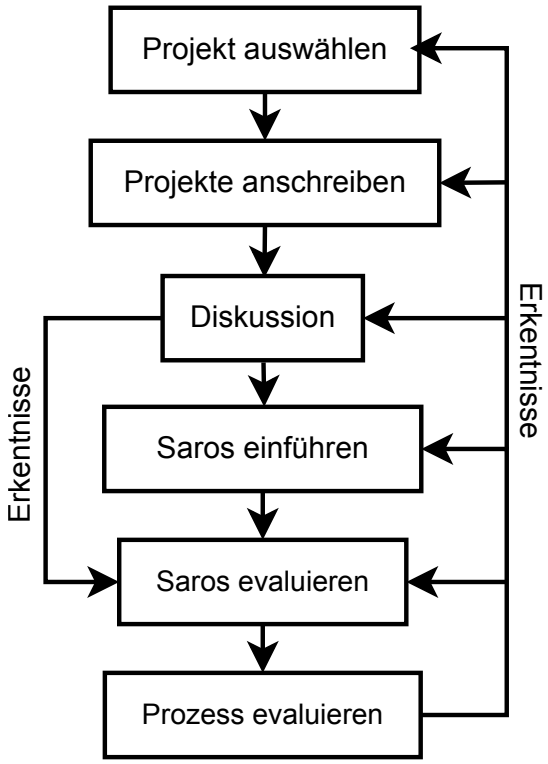

Abbildung 4.1: Methodisches Vorgehen

mir, dass die Projektteilnehmer selbstständig Anwendungsmöglichkeiten von Saros innerhalb des Projekte vorschlagen bzw. Gründe vorbringen, warum Saros nicht einsetzbar ist. Ebenso sollten natürlich innerhalb dieser Diskussion Probleme oder fehlende Funktionalitäten von Saros aufgedeckt werden.

4. Saros einführen
   Dieser Punkt ist eigentlich Teil der Diskussion, d.h. ich denke das die Projektteilnehmer Saros von sich aus testen werden und es keiner Einführung von Saros meinerseits bedarf. Ich stelle mir unter diesem Punkt eher eine aktive Begleitung und Hilfestellung vor oder eine Diskussion wo Saros innerhalb des Projektes genutzt werden kann vor.

5. Saros evaluieren
   Sollten Probleme oder fehlende Funktionalität nicht im Vorfeld aufgedeckt

worden sein, werden sie nach einer Anwendung von Saros bestimmt in Erscheinung treten. Dieser Punkt und die Diskussion sind zentral für die Verbesserung von Saros.

6. Prozess evaluieren
   An dieser Stelle soll mein gesamter Ansatz hinterfragt und evaluiert werden. Daraus soll sich das weitere Procedere innerhalb des Projektes ergeben als auch meine Vorgehensweise für das nächste Projekt angepasst werden.

Bei der eben beschriebenen Auflistung gilt es zu beachten, dass sich diese Punkt nicht so klar trennen lassen, wie ich es hier dargestellt habe. Beispielsweise wird die Diskussion immer parallel zu den anderen Punkten ablaufen.
Dies verdeutlicht Abbildung 4.1, die ebenfalls den Erkenntnisfluss der einzelnen Punkte darstellt. Zu sehen ist, wie sich z.B. durch die Evaluation meines Vorgehens, Erkenntnisse und mögliche Änderungen der einzelnen Punkte für die nächste Iteration bzw. das nächste Projekt ergeben. Gute Ansätze (in Bezug auf die Struktur des Projektes) werden so beibehalten und Vorgehensweisen die schlecht funktioniert haben identifiziert und entsprechend geändert. Dabei ist zu beachten, dass ich OSS-Projekte kontaktieren und Saros aus Innovation anbieten möchte.
Insgesamt ähnelt die Handlungsforschung meinem geplanten Vorgehen mehr, als das qualitative Experiment.

4.2 Innovationseinführung

In dieser Arbeit soll u.a. geklärt werden, wie eine erfolgreiche Kontaktaufnahme zu erreichen ist und welche Faktoren diese beeinflussen. Im folgenden wird der Begriff der *Innovationseinführung* eingeführt sowie meine gewählten Kontaktstrategien erläutert.

4.2.1 Allgemeines

Denning und Dunham [DD06] unterscheiden in ihrem Innovationsmodell folgende Stadien bei der Innovationseinführung:

1. Erfassen (Sensing)
   Zunächst muss ein Problem entdeckt werden, das ein Bedürfnisse nach einer Verbesserung hervorruft. Dieses Problem muss nicht offensichtlich sein und kann sowohl von außen (durch den Forscher) als auch von innen (durch den Gegenstand) entdeckt werden.

2. Vergegenwertigen (Envisioning)
   Hier sollten Lösungen für das Problem erdacht werden.

3. Anbieten (Offering)
   Die Lösungsvorschläge werden dann dem Projekte angeboten.

4. Ausführen (Executing)
   Wird das Angebot von dem Projekt akzeptiert, muss diese Lösung evaluiert/-ausgeführt werden.

5. Aufnehmen(Adopting)
   Sollte die Innovation eine Verbesserung darstellen, wird diese in das Projekt integriert.

6. Erhalten (Sustaining)
   Die Innovation ist endgültig aufgenommen, wenn es keinen Bedarf an Hilfe durch den Forschers gibt und die Innovation sich selbst erhält.

7. Führen (Leading)
   Dieser Punkt ist genau genommen kein Abschnitt in einem Ablauf, sondern beschreibt die Rolle des Forschers, der die Abläufe koordiniert und im Dialog mit dem Projekt/Gegenstand steht.

Im Rahmen dieser Arbeit stellt Saros nun die Innovation dar, der zu untersuchende Gegenstand sind Open Source Projekte. Überträgt man die oben genannten Punkte auf die Einführung von Saros in OSS-Projekte, lassen sich folgende Parallelen erkennen:

Zunächst gibt es kein Problem, das es in OSS-Projekten zu erfassen gibt und keine Lösung, die erdacht werden muss, da Saros schon diese Lösung darstellt. Zudem soll auch herausgefunden werden, ob es überhaupt ein Bedürfnis nach Saros gibt. So kann in einem Projekt nach diesem Bedürfnis bzw. konkreten Anwendungsmöglichkeiten, wo Saros meiner Meinung nach eine Verbesserung darstellt, gesucht werden(1)(2) und Saros als Lösung angeboten werden(3). Wird mein Vorschlag angenommen, muss Saros getestet und evtl. verbessert werden(4) bis Saros in das Projekt integriert werden kann (5). Sollte keine Hilfestellung meinerseits mehr vonnöten sein, werde ich mich aus dem Projekt zurückziehen (6). Dieses Vorgehen der Innovationseinführung spricht ebenfalls für die Nutzung der Handlungsforschung als Untersuchungsplan.

4.2.2 Auswahl der Projekte

Ob eine Kontaktaufnahme bzw. Einführung von Saros erfolgreich oder nicht erfolgreich ist, kann von der Struktur des ausgewählten Projektes abhängen. Diese Struktur gilt es zu analysieren und zu bewerten.

4.2.2.1 Kriterien

Um möglichst hohe Erfolgschance zu haben, sollten die Projekte folgende Kriterien erfüllen. Diese Kriterien werden iterativ während meines Vorgehens erweitert und verändert. Nachfolgend zeige ich die Ausgangsbasis mit der ich das erste Projekt ausgewählt habe. Die aus dieser Untersuchung resultierenden Projektkriterien, werden im Abschnitt 6.3 vorgestellt.

Größe

Die Größe eine Projektes definiert sich in diesem Fall über die Anzahl der Entwickler, da nur diese Personen für meine Anfrage interessant sind. Die minimale, bzw. maximale Anzahl der Entwickler wurde aus der Literatur entnommen. Dort wird eine Größe von 5 bis 50 Entwickler für eine Innovationseinführung empfohlen. [OP07]

Eclipse Nutzung

Da Saros ein Eclipse Plugin darstellt, ist es sinnvoll Projekte zu kontaktieren, dessen Entwickler mit Eclipse arbeiten. Eine Möglichkeit dies herauszufinden, ist die Kommunikationswege nach Diskussionen in denen der Begriff „Eclipse“ vorkommt zu durchsuchen. Eine andere, sicherere Methode ist in der Versionsverwaltung nach einer .project-Datei zu suchen. Diese wird von Eclipse automatisch in jedem Projekt angelegt und ist ein Indiz, dass das Projekt Eclipse zum Entwickeln benutzt.

Offenheit

Um dies oben genannten Kriterien überhaupt überprüfen zu können müssen diese Informationen über das Projekt auch verfügbar sein. Dies erfordert eine gewisse Offenheit. So wäre es wünschenswert wenn viele Informationen über ein Projekt verfügbar sind und die Versionsverwaltung öffentlich zugänglich ist.

4.2.2.2 Auswahl

Um Projekt dieser Art zu finden bietet sich die Webseite *www.sourceforge.net* an, die im Abschnitt 2.2.5schon erwähnt wurde. SourceForge bietet Open Source Projekten eine Vielzahl an Services an, wie z.B. Webhosting, Mailinglisten, Foren, verschiedene Tracker, Versionsverwaltung und vieles mehr. [SOU10g] Die Webseite hosted über 230.000 verschieden Projekte und verfügt über mehr als zwei Millionen registrierten Benutzern. [SOU09] Ein weiterer Grund, warum sich SourceForge für die Suche nach Projekten eignet, ist dass Sourceforge viele Informationen und Statistiken über die verschiedenen Projekte liefert.
Weitere Webseiten dieser Art sind:

- BerliOS
- Freshmeat
- Google Code
- GNU Savannah

- GNA!
- RubyForge
- GitHub

4.2.3 Die Kontaktaufnahme

Die Kontaktaufnahme zu OSS-Projekten mit dem Vorschlag Saros zu testen, kann meiner Meinung nach auf zwei verschiedene Arten geschehen.

1. Die Entwickler kontaktieren.
   Die Entwickler stellen die direkten Ansprechpartner dar, da sie Saros benutzen würden. Die Zustimmung zur Nutzung ist eine Entscheidung, die direkt ihren Arbeitsablauf betrifft. Hierbei muss jeder selber entscheiden, ob mein Vorschlag nützlich ist.

2. Den Maintainer kontaktieren.
   Der Maintainer kann aber ebenfalls Entwickler sein und durch seine Stellung in dem Projekt eventuell andere Entwickler überzeugen bzw. mit sich ziehen, wenn er Saros benutzt. Hierbei kann der Maintainer beschließen, ob er mein Angebot als sinnvoll erachtet.

4.2.4 Kommunikationsmittel

Die Kontaktaufnahme zu den Open Source Projekten wird, vermute ich, hauptsächlich per Email erfolgen. Daher ist es wichtig, eine informative und überzeugende Email zu verfassen. Diese sollte eine kurze Vorstellung meines Anliegens sowie meiner Person beinhalten. Ferner ist nach v.Krogh et al. (2003) die Wahrscheinlichkeit einer erfolgreichen Kontaktaufnahme von Open-Source-Projekten höher, wenn dem Projekt „Geschenkeïn Form von Quelltext-Beilagen angeboten werden [KSL03]. Dies ist bei meiner Anfrage natürlich nicht möglich, aber Saros selbst könnte hier als Geschenk angesehen werden, wenn ich die Vorteile aufzeige, die Saros Open Source Projekten bringen kann. Da dies ebenfalls Gegenstand dieser Arbeit ist,

kann in der ersten Email nur vermutet werden, das Saros tatsächlich die von mir formulierten Vorteile mit sich bringt. Ein weiteres Geschenk, das ich anbieten kann, ist bei der Einführung von Saros zu helfen, in dem ich z.B. Einführungssitzungen mit Interessierten veranstalte.

Bei einer zu langen Email ist die Gefahr hoch, das der Leser diese nur teilweise oder gar nicht liest. Um aber trotzdem möglichst viele Informationen unterbringen zu können habe ich Webseiten[1] angelegt, die weiterführende Informationen liefern und auf die im Emailtext verwiesen wird.
Um zu zeigen, dass ich mich über das jeweilige Projekt informiert habe, ist es sinnvoll konkrete Anwendungsmöglichkeiten von Saros innerhalb des Projektes vorzuschlagen, sofern diese gefunden werden können.
Aus diesen Überlegungen resultierte unten stehender Text, der sich im Laufe der Arbeit noch geändert hat, worauf ich in den einzelnen Projekten näher eingehen werde. Dieser Text wird denn auch im Bezug auf die Anwendungsmöglichkeiten des jeweiligen Projektes angepasst.

> Dear [Projektname] Team,
>
> My name is Eike Starkmann and I'm working working as part of the Saros Team at the Freie University in Berlin.
>
> Saros is an Eclipse plugin for collaborative text editing and distributed pair programming, i.e. it allows two or more developers to work together in real-time on the same files. It is similar to Gobby, SubEthaEdit or Google Docs but focuses on programming in Eclipse.
>
> It is my master thesis to figure out whether Saros is useful when developing Free/Open Source Software.
>
> In my opinion Drupal can benefit from Saros because I think it brings many advantages to Open Source Software development:

1 https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin, https://www.inf.fu-berlin.de/w/SE/DPPforOSS

> * Distributed Pair Programming is like a live peer review. This should help with finding good design, get rid of bugs, increase readability, etc.
>
> * Transferring knowledge should be easier to do when more than one person look at and work with the same code. This should also help to give new developers an introduction to the code.
>
> * In contrast to screen sharing, Saros only shares your actions inside of Eclipse with regards to the project you are both working on (think privacy) and you are still independent to explore the project on your own.
>
> Saros can be useful in the following contexts:
>
> * Working on complicated problems in the code * Performing code reviews * Debugging * Code presentation * Code sprints * Introducing new developers to the project * ...
>
> What do you think? Could you and [Projektname] benefit from doing pair programming using Saros?
>
> If you are interested in Saros but still curious about how it works please visit our website or feel free to contact me.
>
> I hope you will find Saros useful and give me feedback.
>
> Kind regards, Eike Starkmann
>
> Website: https://www.inf.fu-berlin.de/w/SE/DPP Update Site: http://dpp.sf.net/update Saros @ SF: http://sourceforge.net/projects/dpp/ Programming Languages Supported by Saros : https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
>
> – Eike Starkmann This message is part of my Master thesis research. Feel free to contact my advisors in case of inappropriate behavior on my side: christopher.oezbek@fu-berlin.de and stephan.salinger@fu-berlin.de

Dabei ist zu beachten, dass die Datenbasis aus Text bestehen wird. Aus diesem Grunde wird im kommenden Abschnitt die Datenauswertung mittels der qualitativen Inhaltsanalyse eingeführt.

4.3 Datenauswertung

Zur Auswertung und Interpretation der Daten verwende ich die *qualitatative Inhaltsanalyse* nach Mayring (2002) [May02]. Danach wird im Vorfeld der Datenerhebung ein Kategoriensystem erstellt. Die Kategoriedimensionen und das Niveau der Abstraktion müssen bei der qualitativen Inhaltsanalyse vor der Zusammenfassung des Textmaterials erstellt werden. Anschließend wird das Kategoriensystem anhand der Texte induktiv erweitert. Das heißt Aussagen, die in eine Kategoriedefinition eingepasst werden können, bilden eine neue Kategorie, welche in das System mitaufgenommen wird. Der Satz oder Begriff, welcher diese Kategorie bildet, wird Kategorienbezeichnung genannt. Weitere in diese Kategorie passende Begriffe oder Sätze werden ebenfalls mit aufgenommen (Subsumption). Textstellen, welche in keine vorhandenen Kategorie passen, bilden eine neue Kategorie. Mayring schlägt dabei drei Grundformen vor:

- "*Zusammenfassung:* Ziel der Analyse ist es, das Material so zu reduzieren, dass die wesentlichen Inhalte erhalten bleiben. Durch Abstraktion wird ein überschaubares Korpus geschaffen, das immer noch ein Abbild des Grundmaterials ist.

- *Explikation:* Ziel der Analyse ist es, zu einzelnen fraglichen Textteilen (Begriffen, Sätzen ...) zusätzliches Material heranzutragen, welche das Verständnis erweitert, das die Textstelle erläutert, erklärt und ausdeutet.

- *Strukturierung:* Ziel der Analyse ist es, bestimmte Aspekte aus dem Material herrauszufiltern, unter vorher festgelegten Ordnungskriterien einen querschnitt durch das Material zu legen oder das Material aufgrund bestimmter Kriterien einzuschätzen.„ [May02, S. 114-121].

4.4 Das Kategoriensystem

Um die Fragestellung nach erfolgreicher oder nicht erfolgreicher Kontaktaufnahme beantworten zu können, werden die beiden übergeordneten Kategorien *Akzeptanz*

und *Ablehnung* eingeführt. Als Akzeptanz wird jede Aussage gewertet, die beinhaltet, dass entweder Saros im allgemeinen nützlich ist, das Konzept von DPP bzw. PP ein erstrebenswerter Arbeitsprozess ist oder Saros konkret für das aktuelle Projekt anwendbar ist bzw. Anwendung findet. Es wird dabei in *technische* (Abschnitt 4.4.4), *konzeptionelle* (Abschnitt 4.4.5) und *umfeldbezogene* (Abschnitt 4.4.6) Akzeptanz unterschieden. Als Ablehnung wird jede Aussage gewertet, die beinhaltet, dass entweder Saros generell nicht nützlich ist oder Mängel hat, das Konzept von DPP bzw. PP keine sinnvolle Arbeitsmethode oder Prozess angesehen wird oder das Saros oder das Konzept DPP wegen konkreten Problemen nicht in dem aktuellen Projekt verwendet werden kann. Dabei wird ebenfalls in technische (Abschnitt 4.4.1), konzeptionelle (Abschnitt 4.4.2)und umfeldbezogene (4.4.3) Ablehnung unterscheiden.
Mit der dritten übergeordneten Kategorie *Unklarheiten* sollen Informationen über den Nutzen der Dokumentation und der Webseite von Saros, als auch über die Verständlichkeit meiner Anfrage an die Projekte, gesammelt werden. Mit Hilfe dieser Informationen soll es dann möglich sein, meinen Kontakttext als auch die Dokumentation und Webseite von Saros zu verbessern, da Unklarheiten und Missverständnisse durch diese Kategorie sichtbar werden. Die Kategorie *Unklarheiten* ist dabei wieder in zwei Unterbereiche aufgeteilt: *technische Nachfragen* (Abschnitt 4.4.7) und *konzeptionelle Nachfragen* (Abschnitt 4.4.8).

4.4.1 Technische Ablehnung

Diese Kategorien beinhalten, wie schon erwähnt, Aussagen die die Software Saros aus technischen Gründen Ablehnen.

4.4.1.1 Die Kategorie *„Es gibt schon so etwas“* (ege)

In diese Kategorie fallen ablehnende Aussagen mit der Begründung, dass es schon eine andere Software mit gleicher oder ähnlicher Funktionalität gibt. Dabei wird keine Aussage darüber getroffen, ob diese Software in dem Projekt Anwendung findet, das würde dann in die Kategorie *Es wird schon so etwas benutzt* (4.4.3.5) fallen.

4.4.1.2 Die Kategorie *„Defekt“* (def)

Die Kategorie *Defekt* beinhaltet alle Aussagen, welche Fehlverhalten von Saros beschreiben und die Software aus diesem Grund ablehnen. Dabei ist zu beachten, dass fehlende Funktionalitäten, die das Projekt benötigt, nicht dieser Kategorie zugeordnet werden, sondern der Kategorie „Fehlende Funktionalität“ (Abschnitt 4.4.3.1) Es werden in dieser Kategorie nur verbale Aussagen aufgenommenen, d.h. sollten z.B. während einer Testsession mit einem Entwickler Fehler entdeckt werden, werden diese gesondert ausgeführt, es sei denn der Entwickler äußert diese mir gegenüber.

4.4.1.3 Die Kategorie *„Abhängigkeit von einer Entwicklungsumgebung“*(abe)

Eclipse wird nicht von jedem Entwickler als IDE[1] bevorzugt. Ablehnungen von Saros mit der Begründung, das Eclipse nicht verwendet wird, bzw. auch nicht zu Eclipse gewechselt wird, um Saros zu benutzen, fallen in diese Kategorie.

4.4.2 Konzeptionelle Ablehnung

Ablehnungen, die nicht unmittelbar mit der Software Saros sondern sich gegen das Konzept des kollaborativen Zusammenarbeitens bzw. Verteilten Paarprogrammierens richten werden unter der Kategorie *Konzeptionelle Ablehnung* zusammengefasst. Es wird dabei also nicht die Software an sich abgelehnt sondern der ihr Verwendungszweck. Dabei sind sind diese Aussagen nicht auf das aktuelle Projekt bezogen, sondern allgemeiner Natur. Projekt bezogene Aussagen dieser Art fallen unter die Kategorie „Umfeld bezogene Ablehung“ (Abschnitt 4.4.3).

1 Integrierte Entwicklungsumgebung, engl. integrated development environment

4.4.2.1 Die Kategorie *„Abneigung gegen Pair Programming“* (app)

Es gibt einige Parallelen zwischen DPP[1] und PP[2]. Zwar sind mit Saros vielfältigere Anwendungsmöglichkeiten als DPP möglich, es liegt jedoch nahe, dass wenn schlechte Erfahrungen mit PP gemacht wurden, diese auch auf Saros übertragen werden. Unter diese Kategorie fallen Aussagen die Saros ablehnen mit der Begründung, dass schlechte Erfahrungen mit PP gemacht wurden.

4.4.3 Umfeld bezogene Ablehnung

Ablehnungen, die sich weder auf technische Aspekte der Software noch auf deren generelle Anwendungsmöglichkeiten beziehen, haben oft konkrete projektbezogene Gründe, warum Saros nicht benutzt werden kann.

4.4.3.1 Die Kategorie *„Fehlende Funktionalität“* (fft)

Diese Kategorie beschreibt Ablehnungen die beinhalten, dass Saros in dem Projekt nicht eingesetzt werden kann, da eine benötigte Funktionalität fehlt, z.B. die Unterstützung einer verwendeten Programmiersprache.

4.4.3.2 Die Kategorie *„Muss jeder selber entscheiden“* (mje)

Aussagen in dieser Kategorie müssen streng genommen keine klare Ablehnung beinhalten. Die Aussagen, die hier zusammengefasst werden, beinhalten, dass meine Anfrage, ob Saros in dem Projekt benutzt werden kann, nicht sinnvoll ist. Ob Saros in dem Projekt benutzt werden soll oder nicht kann nicht gemeinschaftlich entschieden werden, da diese Entscheidung dem jeweiligen Entwickler im Einzelfall obliegt. Ich sehe es dennoch als sinnvoll an, diese Kategorie den Ablehnungen zuzuordnen, da hier meine Anfrage als unangebracht erachtet wird.

1 Verteilte Paarprogramierung, engl. distributed pair programming
2 Paarprogrammierung, oder engl. pair programming

4.4.3.3 Die Kategorie *„Wenige Entwickler würden das benutzen“* (web)

Ablehnungen mit der Begründung, dass Saros nur wenig Anwendungsmöglichkeiten in dem Projekt hat, fallen in diese Kategorie.

4.4.3.4 Die Kategorie *„Zu große Einarbeitungszeit“* (zge)

Diese Kategorie umfasst Aussagen, die beinhalten, dass die Einarbeitungszeit in Saros zu groß ist bzw. die Integration in das Projekt zu viel Zeit benötigt, zusammen.

4.4.3.5 Die Kategorie *„Es wird schon so etwas benutzt“* (ewb)

In dem aktuellen Projekt wird schon eine Software für die von mir vorgeschlagenen Anwendungsmöglichkeiten benutzt. Ablehnungen dieser Art werden in dieser Kategorie zusammengefasst.

4.4.3.6 Die Kategorien *„Produkt könnte nützlich sein, aber nicht für uns“ und „Konzept könnte nützlich sein, aber nicht für uns“* (pnu) (knu)

Die positiven Aussagen dieser Kategorie sind genereller Natur, die Ablehnung ist aber projektbezogen. Das heißt, das entweder das Produkt Saros oder das Konzept hinter Saros generell von Nutzen ist, allerdings nicht für das aktuelle Projekt.

4.4.3.7 Die Kategorie *„Entwickler arbeiten zu asynchron“* (eaa)

Aussagen die beinhalten, dass Saros in dem Projekt nicht benutzt werden kann, da die Entwickler sowohl räumlich zu verteilt sind, als auch zu unterschiedliche Arbeitszeiten haben, um gleichzeitig mit Saros arbeiten zu können, werden in diese Kategorie zusammengefasst. Es handelt sich hierbei nicht um ein technisches Problem von Saros, sondern um ein konzeptionelles.

4.4.3.8 Die Kategorie *„Entwickler arbeiten schon in einem Büro zusammen“* (eab)

Die Entwickler des Projektes arbeiten in einem Büro zusammen und sehen daher keine Nutzen in Saros.

4.4.4 Technische Akzeptanz

Diese Kategorien beinhalten positive Äußerungen, die sich direkt auf die Software Saros beziehen.

4.4.4.1 Die Kategorie *„Interesse/sinnvoller Vorschlag“* (isv)

Antworten auf meine Anfrage die generelles Interesse und positive Äußerungen beinhalten fallen in diese Kategorie.

4.4.4.2 Die Kategorie *„Anwendungsvorschlag“* (aws)

Aussagen von Entwicklern, die generelle Anwendungsmöglichkeiten von Saros vorschlagen werden in dieser Kategorie zusammengefasst. Dabei beziehen sich diese Aussagen nicht auf das aktuelle Projekt, diese werden der Kategorie „Anwendungsvorschlag für das Projekt“ (4.4.6.2) zugeordnet. Ich habe diese Kategorie der technischen und nicht der konzeptionellen Akzeptanz untergeordnet, da es sich hierbei um Vorschläge handelt, zu welchen Zweck das *Programm* Saros eingesetzt werden kann.

4.4.4.3 Die Kategorie *„Schon nach so etwas gesucht“* (sng)

Antworten auf meine Anfrage, die beinhalten, das das Projekt bzw. der Entwickler schon länger nach einer Software wie Saros sucht, fallen in diese Kategorie.

4.4.4.4 Die Kategorie *„Gute Lösung (Saros getestet.)“* (gul)

Diese Kategorie fasst alle positiven Aussagen zusammen, die nach einem Test von Saros geäußert werden. Sie ähnelt dabei der Kategorie *„Interesse/sinvoller Vorschlag“*, ich wollte aber bewusst eine Unterscheidung machen zwischen Aussagen die nur auf Grund meiner Vorstellung und der Dokumentation von Saros gemacht werden und Aussagen, die auf eigenen Erfahrungen des jeweiligen Entwicklers beruhen.

4.4.5 Konzeptionelle Akzeptanz

Alle positiven, auf den Zweck bzw. die Anwendungsmöglichkeiten von Saros bezogenen Äußerungen werden in dieser Kategorie zusammengefasst.

4.4.5.1 Die Kategorie *„Produktivitätssteigerung“* (pts)

Aussagen die beinhalten, dass Anwendungsmöglichkeiten von Saros wie z.B DPP oder Peer-Reviews[1] , nützliche Arbeitsmethoden sind, werden dieser Kategorie zugeordnet. Dabei handelt es sich hier wieder nur um generelle und nicht projektbezogene Aussagen.

4.4.6 Umfeld bezogene Akzeptanz

Diese Kategorie gleicht der *Konzeptionellen Akzeptanz*, jedoch mit dem Unterschied, das die Aussagen in dieser Kategorie das Projekt direkt betreffen.

4.4.6.1 Die Kategorie *„Generell von Nutzen für das Projekt“* (gnp)

Unter dieser Kategorie werden alle Aussagen zusammen gefasst, die beinhalten, das die Software Saros für das Projekt von Nutzen sein kann. Dabei wird nicht näher auf den Kontext der Benutzung eingegangen.

4.4.6.2 Die Kategorie *„Anwendungsvorschlag für das Projekt“* (afp)

Dieser Kategorie werden Aussagen zugeordnet, welche beinhalten das Saros für das Projekt von Nutzen ist, dabei aber auch konkrete Anwendungsmöglichkeiten beinhalten. Der Kontext, in welchem Saros für das Projekt nützlich ist, ist hierbei ausschlaggebend.

1 Peer-Review (dt. Begutachtung durch Kollegen) ist ein Verfahren zur Beurteilung von Projekten oder Software durch unabhängige Gutachter, „Peers“ (engl. für Kollegen, Fachleute), mit dem Ziel der Qualitätssicherung

4.4.7 Technische Nachfragen

Diese Kategorie fasst alle Nachfragen zusammen die technischer Natur sind. Dabei werden die Unterkategorien *Installation*, *Performance*, *Kompatibilität* und *Verwendete Technologien* benutzt, um die Nachfragen besser strukturieren zu können.

4.4.8 Konzeptionelle Nachfragen

Konzeptionelle Nachfragen werden in die Unterkategorien *Verständigung*, *Effektivität* und *Sonstige* unterteilt. *Verständigung* beinhaltet dabei Nachfragen bezüglich des Ablaufes der Kommunikation während der Arbeit mit Saros, bzw. wie die Teams gebildet werden können. Die Kategorie *Effektivität* beinhaltet Nachfragen bezüglich der Produktivität der von Saros abgedeckten Anwendungsmöglichkeiten, z.B. die Nachfrage, ob DPP zu einer Produktivitätssteigerung führt.

5 Projekte

Dieser Abschnitt beschreibt alle Projekte, die ich während meiner Diplomarbeit kontaktiert habe. Abbildung 5.1 zeigt den zeitlichen Rahmen, in dem diese kon-

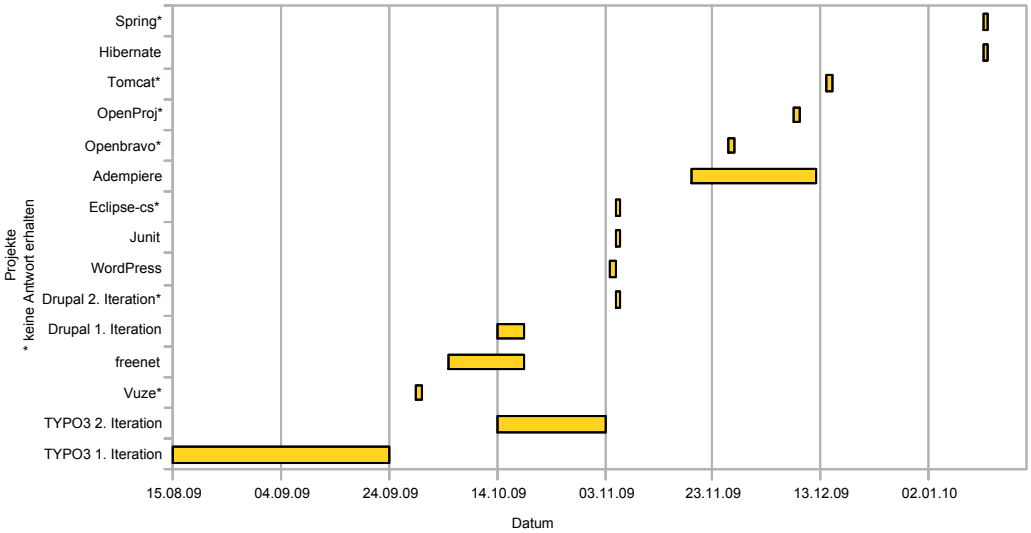

Abbildung 5.1: Die kontaktierten Projekte

taktiert wurden. Es soll dadurch verdeutlicht werden, dass die Projekte nicht nur nacheinander kontaktiert wurden, wie die folgenden Beschreibungen vermuten lassen, sondern teilweise auch gleichzeitig.

Zur Visualisierung des Emailverkehrs werden innerhalb der Vorstellungen der Resultate eines Projektes sogenannte *Thread Arcs* verwendet. Dabei werden die verschiedenen Emails chronologisch hintereinander als Punkte dargestellt. Antworten und Diskussionsabläufe werden durch Bögen (Arcs) zwischen diesen Punkten verdeutlicht[Ker03]

. Abbildung 5.2 zeigt einen exemplarischen Diskussionsverlauf. Die Farben repräsentieren dabei jeweils eine Person. Erstellt werden die Bilder mit dem Programm *ThreadViz*[1] welches als Masterarbeit von Alexander C. Hubmann-Haidvogel realisiert wurde. [Hub08] Durch dieses Vorgehen erhoffe ich mir ein nachvollziehbaren Diskussionablauf präsentieren zu können.

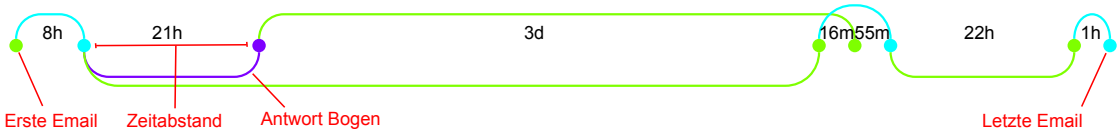

Abbildung 5.2: Exemplarischer Diskussionsverlauf auf einer Mailingliste

5.1 TYPO3

5.1.1 Das Projekt

TYPO3 ist ein CMS[2] für Webseiten, welches ursprünglich von Kasper Skårhøj entwickelt wurde [TYP09c]. Typo3 basiert auf der Scriptsprache PHP[3] und ist unter der GNU[4] General Public License Version 2 verfügbar[TYP09e].

5.1.2 Geschichte

Die Entwicklung von TYPO3 begann im Jahr 1997 und wurde ab 1998 kommerziell für superfish.com weiterentwickelt. 1999 trennte sich Kasper Skårhøj von superfish.com, erhielt aber die Rechte an TYPO3 und konnte so die Entwicklung alleine fortführen.
Im August 2000 wurde die erste Betaversion von TYPO3 veröffentlicht und stand der Open Source Gemeinde zur Verfügung. Es bildete sich eine immer größere Gemeinde um TYPO3 und am 24. Mai 2002 wurde die erste vollwertige Version

1 http://threadvis.mozdev.org/
2 Content Management System
3 http://php.net/index.php
4 **G**NU is **n**ot **U**nix

veröffentlicht: TYPO3 Version 3.0. Sehr bald folgte die Version 3.5, welche den Extension Manager enthielt. Der Extension Manager machte es möglich, Erweiterungen (Extensions) für TYPO3 zu schreiben, ohne den Kern zu ändern, d.h. TYPO3 konnte so modular erweitert werden. Durch diese Möglichkeit entstand ebenfalls eine Gemeinde, welche Erweiterungen für TYPO3 schrieb und noch schreibt.
Ende 2004 wurde die TYPO3 Association mit der Aufgabe, die Kernentwickler zu unterstützen, gegründet [TYP09c].
Im April 2007 gab Kasper Skårhøj die Weiterentwicklung von TYPO3 ab. Die Entwicklung findet momentan in zwei Zweigen statt: Den 4.x- und den 5.x-Zweig. Der TYPO3 5.x-Zweig wurde von Robert Lemke initiiert und besitzt eine völlig neue Architektur.

5.1.3 Das Team

Um die Entwicklung von TYPO3 zu gewährleisten haben sich mittlerweile eine Vielzahl von Gruppen gebildet, die jeweils über eine oder mehrere Mailinglisten verfügen. Viele Mitglieder in mehreren Gruppen aktiv sind. Die wichtigsten sind:

- TYPO3 v.4:
  Dieses Team ist verantwortlich für die Weiterentwicklung des 4.x-Zweiges.
  - CORE:
    Das TYPO3 v.4 Core Team besteht aus 29 Mitgliedern. Die zugehörigen Mailinglisten sind:
    * TYPO3-project-4-3:
      Für generelle Diskussionen
    * TYPO3-team-core :
      Diese Liste dient nur für Patch-Anfragen sowie Antworten auf Patch-Anfragen.
  - TYPO3 4.3 Projekt:
    Diese Kategorie beinhaltet Projekte, welche in das nächste Release

einfließen sollen, im Moment Version 4.3. Das aktuelle Projekt heißt MVC[1] - Framework[2] und besteht aus acht Mitgliedern.

- TYPO3 v.5:
  Dieses Team ist verantwortlich für die Entwicklung des 5.x Zweiges und besteht aus fünf Mitgliedern. Die zugehörigen Mailinglisten sind:
  - TYPO3-project-5_0-general:
    Für generelle Diskussionen
  - TYPO3-team-core-v5 :
    Diese Liste dient ebenfalls nur für Patch-Anfragen sowie Antworten auf Patch-Anfragen.
- DocTEAM
  Das Dokumentationsteam bestehend aus 16 Mitgliedern.
- FLOW3
  Dieses Entwicklungsframework ist aus TYPO3 v.5 entstanden, es besteht aus 13 Mitgliedern, zur Kommunikation werden die Mailinglisten von TYPO3 v.5 benutzt.
- Usability
  Diese Team versucht die Benutzbarkeit von TYPO3 zu verbessern und besteht aus zwölf Mitgliedern.
- DEV3
  DEV3 ist ein TYPO3 und FLOW3 Entwicklungsplugin für Eclipse. Das Team besteht aus zwölf Mitgliedern und benutzt eine eigene Mailingliste

1 Model View Controller (MVC) ist ein Architekturmuster, bei dem die Software in die drei Einheiten *Datenmodell (engl. model)*, *Präsentation (engl. view)* und *Programmsteuerung (engl. controller)* unterteilt wird. Dadurch können die einzelnen Komponenten wiederverwendet und spätere Änderungen einfach durchgeführt werden

2 Ein Framework, engl. für „Rahmenstruktur“, ist ein Programmiergerüst, das in der Softwareentwicklung verwendet wird

- forge.typo3.org
  Dieses Team besteht aus acht Mitgliedern und kümmert sich um die Website forge.typo3.org.

Es existiert eine weitere Mailingliste: typo3-dev. Diese wird für generelle Diskussionen im Bereich der Extension- und Kernentwicklung benutzt.

5.1.4 Projektauswahl

5.1.4.1 Begründung

TYPO3 ist das erste Open Source Projekt, das ich ausgewählt hatte. Dies geschah in erster Linie, da ich privat regelmäßig mit TYPO3 zu tun hatte. So ist es das Open Source Projekt mit dem ich persönlich am besten vertraut bin. Zudem sprachen weitere Gründe für dieses Projekt:

- Mit einer Größe von 29 festen Mitglieder und einer großen Community war durchaus die Chance gegeben, dass sich Entwickler in Echtzeit zusammenfinden und entwickeln können.
- Aus persönlichen Erfahrungen mit Extensionentwicklern wusste ich, dass Eclipse oft Anwendung findet.
- TYPO3 pflegt einen öffentlich einsehbaren Entwicklungsprozess, so dass genügend Informationen für Anwendungsmöglichkeiten von Saros zu finden sind.
- Ca. 780 Commits pro Monat weisen darauf hin, dass an TYPO3 aktiv entwickelt wird [TYP09b].
- Weil die Kommunikation ebenfalls öffentlich einsehbar ist, kann untersucht werden, in welchem Forum meine Anfrage am besten unterzubringen ist. Auf diesen Aspekt gehe ich im nächsten Abschnitt genauer ein.

5.1.4.2 Kontaktaufnahme

Da ein großer Teil der Kommunikation innerhalb der TYPO3-Gemeinde über Mailinglisten erfolgt, entschied ich mich, dieses Projekt nur über die Mailingliste zu kontaktieren, d.h. die Entwickler selbst anzuschreiben. Welche Themen auf welchen Listen diskutiert werden, ist in Regeln auf der Webseite festgelegt:

> „The TYPO3 community continues to grow at a rapid rate! As a result, the mailing lists / newsgroups are becoming much busier, so we felt it was time to post some basic rules and guidelines on how they should be used.
> We aren't trying to lay down the law to the community, but if we all follow some basic, common sense guidelines, the lists will be easier to use, and more effective, for everyone [...] “ [TYP09f]

> „Until now, only members of the core team were allowed to post messages to that list. However, in order to open up for community involvement we have decided to allow non-members write access to the list as well*. In order to keep the work in the list as effective as it used to be, postings to the core list are only allowed if they
>
> 1. are patch requests adhering to the format described below, or
> 2. are replies to existing patch requests stating whether the patch worked or not.
>
> It is strictly not allowed to start other new threads or discussions. For discussions use the TYPO3-dev list. All postings should be sent in plain text format (no HTML mails) and must carry the sender's real name in the "From"header of the mail.“ [TYP09a]

Nach Durchsicht der Archive der Mailinglisten erschienen mir folgende Listen als geeignet:

- typo3-project-5_0-general@lists.netfielders.de

  Diese Liste hatte keine besonders große Aktivität und weniger Mitglieder als typo3-dev. Innerhalb von 37 Tagen sind 47 Emails über diese Liste verschickt worden, was einem Durchschnitt von 1,2 Emails pro Tag entspricht. Die Liste wurde ausgewählt, weil dort allgemeine Diskussionen bezüglich der TYPO3-5.x-Entwicklung und des Entwicklungsprozesses an sich geführt wurden. Diese offene Diskussion machte es möglich, dass ich Saros zur Verbesserung des Entwicklungsprozesses vorschlagen konnte ohne die Listenregeln zu verletzen.

- typo3-dev@lists.netfielders.de

  Diese Liste ist mit 19.4 Emails pro Tag (718 in 37 Tagen) bedeutend aktiver und größer als typo3-project-5_0-general. Beide Listen sind thematisch ähnlich, mit dem Unterschied das typo3-dev thematisch nicht an den 5.x-Zweig geknüpft ist, sondern das gesamte TYPO3 Projekt umfasst. Da auf dieser Liste ebenfalls sehr offene Diskussionen stattfinden und Verbesserungsvorschläge aller Art gemacht werden, bot sich diese Liste ebenfalls für eine Kontaktaufnahme an.

5.1.5 Erste Iteration

Die erste Iteration beanspruchte den Zeitraum vom 15.08.09 bis zum 24.09.09.

5.1.5.1 Resultate

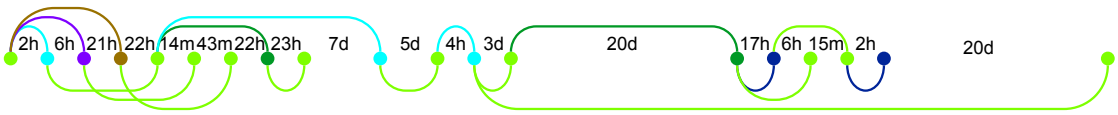

Abbildung 5.3: Emailverkehr auf der Liste typo3-project-5_0-general: TY1, TY2, TY3, TY4, TY5

Wie in Abb.5.3 zu sehen ist, antworteten vier Entwickler innerhalb von drei Tagen auf meine Email (A.1.1).

Der fünfte Entwickler fragte, ob ein Plugin wie Saros auch für Netbeans[1] zu haben sei (A.1.13). Drei Entwickler waren enthusiastisch und wollten Saros gerne ausprobieren. TY1,2 und 4 stellten aber fest, dass sie PHP-Unterstützung benötigen (A.1.1, A.1.1, A.1.3).
Daraufhin recherchierte ich, dass es mehrere Plugins bzw. Produkte für Eclipse gibt :

- PHPeclipse[2]
  PHPeclipse basiert auf Eclipse wird aber von einer eigenständigen Team gepflegt und entwickelt. PHPeclipse ist eine Abspaltung von Eclipse.
- ZendStudio[3]
  Kommerzielle Software basierend auf Eclipse.
- PDT [4]
  Die offizielle PHP Erweiterung für Eclipse.
- Aptana [5]
  Eclipse Plugin für mehrere Sprachen u.a. auch PHP.
- DEV3 [6]
  PlugIn zur Entwicklung von TYPO3 Anwendungen, welches auf PDT aufbaut, aber ebenfalls eigene Editoren mitliefert, z.B. den TypoScript Editor.

Diese Plugins bzw. Produkte testete ich mit Saros. Außer mit PDT und DEV3 war Saros mit allen Plugins kompatibel. TY1 merkte allerdings an, dass innerhalb des TYPO3-Projektes PDT das am meisten verwendete Plugin ist:

> „I guess most of us really need PDT in order to try it out properly :-)“ [A.1.1].

1 http://netbeans.org/
2 http://www.phpeclipse.com/
3 http://www.zend.com/de/products/studio/
4 http://www.eclipse.org/pdt/
5 http://aptana.com/
6 http://www.dev3.org/

Ich wies darauf hin, dass wir in dem nächsten Relesase dieses Problem gelöst haben werden, woraufhin TY1 mir versicherte, dass er Saros dann testen wolle (A.1.1).

Da der PHP Editor von DEV3 auf PDT aufbaut, war Saros mit diesem ebenfalls nicht kompatibel, doch auch der mitgelieferte TypoScript[1] Editor konnte mit Saros nicht benutzt werden. Dies lag daran, dass der Editor bei Textänderungen Eclipse nicht während des Schreibens, sondern erst beim Speichern benachrichtigt. Daher konnte Saros Textänderungen immer nur nach dem Speichern an andere Sessionteilnehmer verschicken und nicht während des Schreibens. Dies wurde auf dem Bugtracker von DEV3 angemerkt[2], da das Problem an DEV3 lag und nicht an Saros.

TY2 äußerte Zweifel an der Einsetzbarkeit von Saris in OSSP[3], da die Teamfindung durch die unterschiedlichen Arbeitszeiten erschwert sein könne:

> „I think it can be a quite useful tool, even if I admit, that in such an open-source project the working times will be very, very different so that it might be difficult for teams to find together.“ [A.1.1]

Daraufhin untersuchte ich das Repository, um Daten für oder gegen diese These zu finden.

Aus Abb. 5.4 und Abb. 5.5 geht hervor, dass hauptsächlich von Montag bis Freitag zwischen 12-18 Uhr an dem Projekt gearbeitet wird, wobei Dienstag und Freitag die Hauptarbeitstage zu seien scheinen.

Eine Gegenüberstellung der Commits der einzelnen Entwickler, wie in Abb. 5.6, zeigt, dass es durchaus zeitliche Überschneidungen gibt, d.h es gibt Fälle, in denen mindestens zwei Entwickler zur gleichen Zeit an dem Projekt gearbeitet haben. Es liegt ebenfalls die Vermutung nahe, dass Robert Lemke die Hauptarbeit in dem Projekt leistet. Genauer betrachtet ergibt sich, wie in Tabelle 5.1 zu sehen, das Robert Lemke 58 Prozent der Commits 2009 gemacht hat.

1 http://typo3.org/documentation/document-library/core-documentation/doc_core_tsbyex/0.0.16/view/
2 http://forge.typo3.org/issues/show/4772
3 Open Source Software Projekt

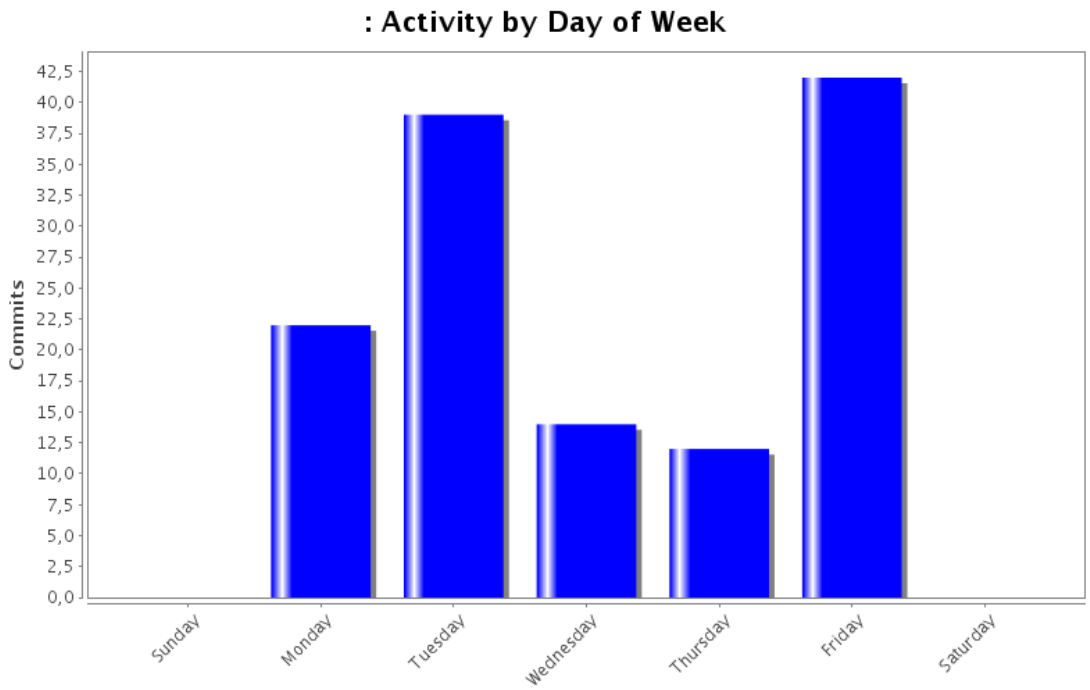

Abbildung 5.4: TYPO3v5: Verteilung der Commits über die Wochentage im Jahr 2009

Entwickler	Codezeilen	Änderungen
robert	687 (58.3%)	71 (67.6%)
f-fish	220 (18.7%)	24 (22.9%)
bwaidelich	204 (17.3%)	1 (1.0%)
andi	67 (5.7)	9 (8.6%)

Tabelle 5.1: Codezeilen und Änderungen der Entwickler am Projekt TYPO3v5 im Jahr 2009

Da ich einen Bug in DEV3 gefunden hatte und das DEV3-Entwicklungsteam auf jeden Fall Eclipse als IDE einsetzt, erschien es mir erfolgversprechend DEV3 über die Mailingliste dev3-dev@lists.cross-content.com zu kontaktieren. Ich antwortete darauf, dass wir daran arbeiten DEV3 zu unterstützen (A.1.4).
Dennoch gab es keine Antwort auf meine Email. Auf der Mailingliste typo3-dev gab es ebenfalls keinerlei Reaktion. Da es jedoch auf der kleineren Mailingliste typo3-project-5_0-general viel positive Resonanz gab, kontaktierte ich ebenfalls die

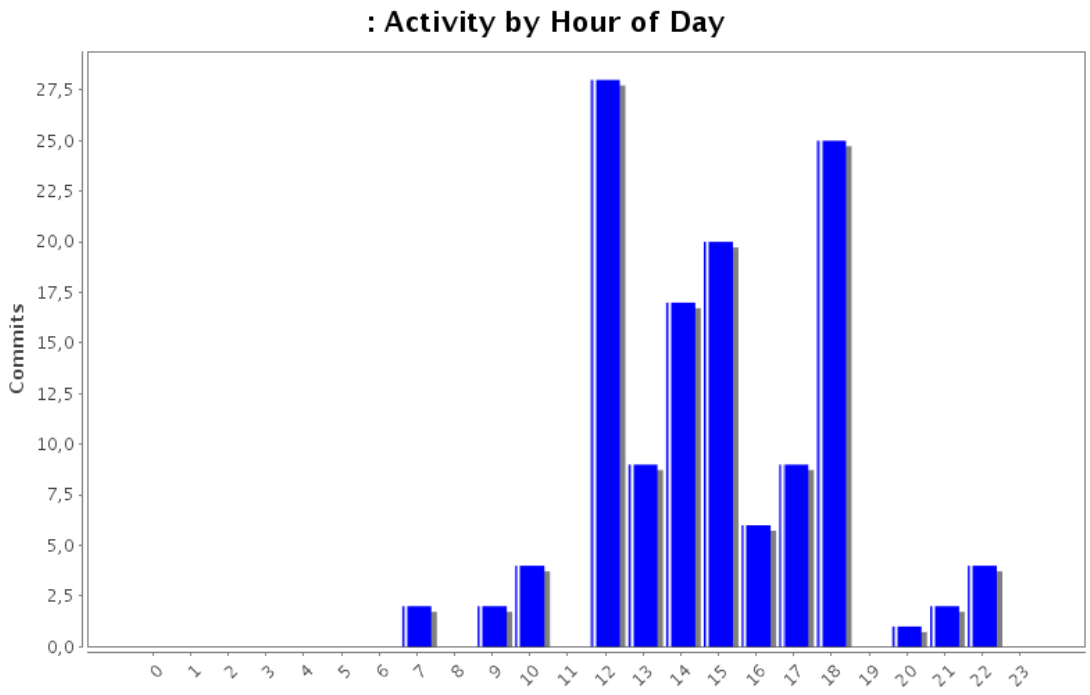

Abbildung 5.5: TYPO3v5: Verteilung der Commits über die Uhrzeiten im Jahr 2009

kleinere Liste typo3-project-4-3 (A.1.9). Diese Mailingliste ist nur für das aktuelle Release von TYPO3 4.3 zuständig und von der Größe und Aktivität vergleichbar mit typo3-project-5_0-general. In der Kontaktemail wies ich darauf hin, welche Mailinglisten von TYOP3 ich bereits kontaktiert hatte und das ich schon gute Resonanz bekommen hätte.

Nachdem unsere Forschungsgruppe die Version 9.8.21.r1660[1] von Saros veröffentlicht hatte und damit PDT nun benutzbar war, berichtete TY1 über seine ersten Erfahrungen mit Saros, die sehr positiv ausfielen:

> „It seemed to work, and it looks like a really nice approach!“ [A.1.7]

Er hatte mit TY4 zusammengearbeitet (A.1.12). Dabei entdeckte er einen Defekt:

1 http://sourceforge.net/news/?group_id=167540

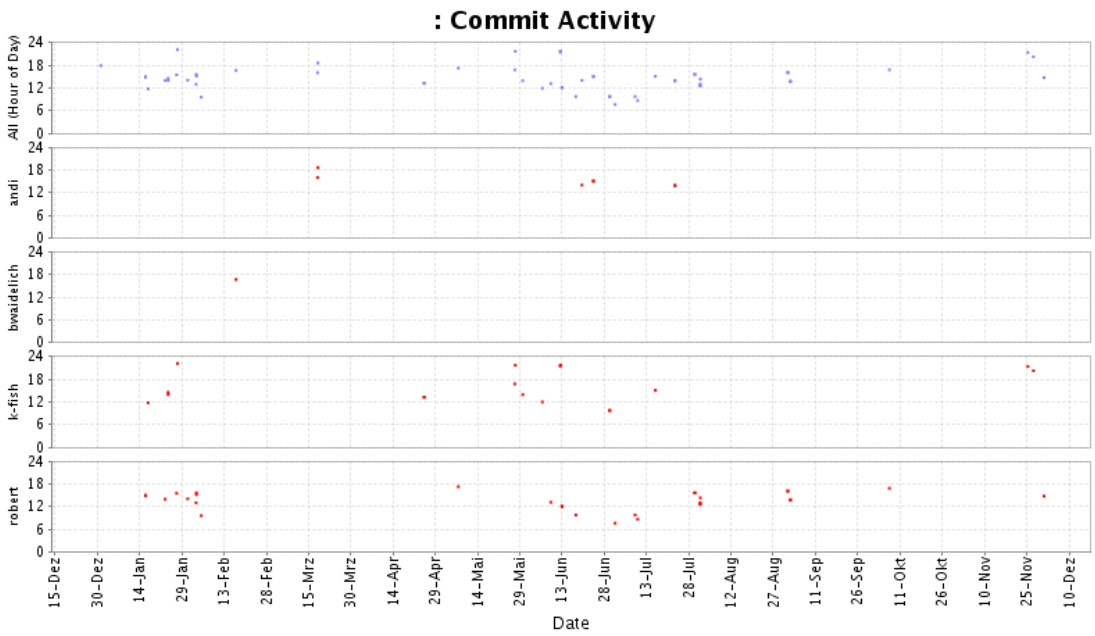

Abbildung 5.6: Die im Jahr 2009 durchgeführten Commits der Typo3v5 Entwickler

Die Contribution Annotation funktionierte mit dem PDT-Editor nicht (A.1.7). Dies war jedoch kein schwerwiegender Defekt und sollte sich auf die Arbeit kaum ausgewirkt haben. TY1 wollte auch weiterhin mit Saros arbeiten. TY4 hatte vor, in Zukunft mit Netbeans zu arbeiten, weshalb er Saros dann nicht weiter benutzen könne. Die Teambildung geschah über Twitter.[1]

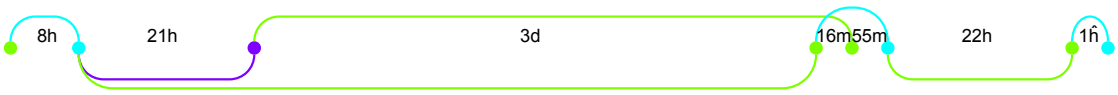

Abbildung 5.7: Emailverkehr auf der Liste typo3-project-4-3: TY6 und TY7

Wie in Abb. 5.7 zu sehen ist, antworteten über die typo3-project-4-3 Mailingliste nur zwei Entwickler. TY6 äußerte sich kritisch in Bezug auf PP. Die Nachteile von PP sind seiner Meinung nach:

1 http://twitter.com/skurfuerst/statuses/3663910441

- Nur wenige Menschen können Paarprogrammieren.
- Die beiden Partner müssen das gleiche „Level“ haben, damit es nicht zu Master/Slave Programmierung.[1]
- PP führt selten zu einem guten Resultat.

Ferner merkte er an, dass seine Erfahrungen aus einem großen kommerziellen Projekt stammen. Er forderte die Liste auf, andere Erfahrungen zu schildern (A.1.9 ; A.1.11).
TY7 widersprach TY6 und zählte viele Vorteile von PP auf, welche sich stark von TY6 unterscheiden:

- PP macht Spaß.
- PP hilft Wissen zu verbreiten.
- PP ist sehr produktiv:
  - Spart Zeit bei Durchsichten.
  - Mit PP werden Aufgaben schneller abgearbeitet.
  - „Sozialer Druck“ (Pair Pressure) sorgt für eine Fokussierung auf die Aufgabe.

Er war sich aber nicht sicher, ob sich diese Vorteile auch auf DPP bzw. Saros im konkreten übertragen lassen und fragte sich, wie dort Kommunikation stattfände, da es nicht möglich sei sich verbal zu verständigen (A.1.10).
Ich erwiderte ihm darauf, dass für die verbale Kommunikation meistens externe Programme wie z.B. Skype[2] oder Mumble[3] eingesetzt werden. Ferner erläuterte ich ihm die Unterschiede zwischen PP und DPP und wies im Besonderen darauf hin, dass mit Saros *gleichzeitig* in einem Dokument gearbeitet werden kann, was in PP

1 Meint in diesem Fall, das nur der eine Partner die Arbeit erledigt, der andere daneben sitzt und quasi nichts macht. Es kommt also nicht zu einem offenen Dialog und Gedankenaustausch.
2 http://www.skype.com
3 http://mumble.sourceforge.net/

nicht vorgesehen ist (A.1.11).
Keiner der Beiden äußerte sich darüber, ob er Saros testen will.

Während der ersten Iteration wurden zwei Artikel auf zwei verschiedenen Blogs[1][2] veröffentlicht. Beide nahmen Bezug auf meine Emails an TYPO3 bzw. zitierten diese fast vollständig.

Von vier kontaktierten Listen kam es auf zweien zu einer Diskussion. Insgesamt haben sechs Entwickler geantwortet und zwei von ihnen haben Saros erfolgreich getestet.

Mit Hilfe der qualitativen Inhaltsanalyse, vgl. Abschnitt 4.3, kann die gesamte Kommunikation folgendermaßen kategorisiert und zusammengefasst werden:

1. Ablehnung
   a) Technische Ablehnung
      i. Mangelhaftes Produkt
         - Defekt:

           „[...]I thought the typed text by the person is highlighted as well, no?“ [TY1 A.1.7].

         - Abhängigkeit von einer Entwicklungsumgebung (Eclipse):

           „Since Netbeans is *awesome* when it comes to supporting PHP 5.3, I am no longer using Eclipse/PDT“ [TY4 A.1.12].

   b) Konzeptionelle Ablehnung
      i. Keine sinnvolle Arbeitsmethode oder Prozess
         - Schlechte Erfahrungen mit Pair Programming:

1 http://fritzthomas.com/overall/467-saros-distributed-pair-programming-plugin-for-eclipse/
2 http://www.maxhorvath.com/2009/08/saros-distributed-pair-programming-plugin-for-eclipse.html

„[...]I do not really believe into pair programming.“ [TY6 A.1.9]
„To me pair programming is more a distraction than creativity. I am ok to *discuss* how to create something but when it comes to coding I found that one person coding produces better results.“ [TY6 A.1.11]
„The concept sounds nice but in practice I never saw satisfying result.“ [TY6 A.1.9]

c) Umfeld bezogene Ablehnung

i. Produkt unpassend für das Projekt

- Fehlende Funktionalität:

  „I guess most of us really need PDT in order to try it out properly :-)“ [TY1 A.1.1].

- Zu wenige Entwickler würden das benutzen:

  „I think there are people who can do it but it is rare.“ [TY6 A.1.9]

ii. Konzept unpassend für das Projekt

- Entwickler arbeiten zu asynchron:

  „[...]in such an open-source project the working times will be very, very different so that it might be difficult to for teams to find together.“ [TY2 A.1.1]

2. Akzeptanz

a) Technische Akzeptanz

i. Sinnvolles Produkt

- Interesse/sinnvoller Vorschlag:

  „Yep, I'll give it a try soon after my holidays!“ [TY1 A.1.1]
  „I would be happy to try it with somebody[...]“ [TY2 A.1.1]
  „That sounds quite cool :-)“ [TY1 A.1.1]

„Sounds like a very cool project!“ [TY4 A.1.3]
„Ok, I'll give it a try[...]“ [TY4 A.1.3]
„Looking forward to try it again :)“ [TY1 A.1.7]

- Gute Lösung (Saros getetest)

  „I think it can be a quite useful tool,[...]“ [TY2 A.1.1]
  „It seemed to work, and it looks like a really nice approach!“ [TY1 TY1-31.08.09-14.16]

3. Unklarheiten
   a) Technische Nachfragen
      i. Installation:

         „Where on http://www.eclipse.org/pdt/downloads/ can I download the all-in-one build[...]“ [TY1 A.1.6]

      ii. Produktvergleich
         - Zusammenhang mit anderen Produkten:

           „How does it relate to: http://wiki.eclipse.org/DocShare_Plugin ?“ [TY1 A.1.1]
           „Is it something like Bespin?“ [TY3 A.1.2]

   b) Konzeptionelle Nachfragen
      i. Effektivität:

         „I wonder how the communication in DPP works if you cannot talk to each other.“ [TY7 A.1.10]

5.1.5.2 Fazit der ersten Iteration

Ich hatte Bedenken, dass ich die Kontaktemail zu formell formuliert hatte, da die meisten Antworten umgangssprachlich ausfielen:

„Sounds like a very cool project!“ [A.1.3]

Doch da meine Email in verschiedenen Blogs fast im gleichen Wortlaut veröffentlicht wurde, hat es sich im hachhinein als sinnvoll herausgestellt eine formellere Email geschrieben zu haben. Daher werde ich für den Kontakt des nächsten Projektes diesen Text beibehalten.

Die Listen, auf denen eine Diskussion stattfand, waren kleinere Listen mit wenig Emailverkehr, typo3-project-5_0-general mit 1,2 Emails pro Tag und typo3-project-4-3 mit 6,6 Emails pro Tag im Zeitraum der Untersuchung, im Vergleich zu 19,4 Emails auf typo3-dev, daher vermutete ich, dass es erfolgsversprechender ist, kleinere Listen zu kontaktieren und habe diese These bei der Kontaktaufnahme des nächsten Projektes überprüft.
Eine weitere offene Frage war, ob eine Kontaktaufnahme über den Maintainer erfolgreicher ist als über die Mailingliste, was bei großen Listen der Fall sein könnte. Dies versuchte ich in den nächsten Projekten ebenfalls zu untersuchen.

Twitter wurde von einem Entwickler benutzt, um einen Programmierpartner zu finden. Um eine höhere Bekanntheit und bessere Vernetzung von Saros zu erreichen wurde ein Twitter[1] und ein identi.ca[2] Account angelegt.

Ein weiteres Ergebnis der ersten Iteration ist, dass Saros, wie zuvor vermutet, PHP unterstützen muss, damit die TYPO3 Entwickler Saros benutzen können. Es wurden mehrere Produkte und Plugins getestet und Saros in dieser Hinsicht weiterentwickelt. Aus der Iteration resultierende Feature Requests und Bug Reports sind folgende:

- 2849925 Contribution annotation doesn't work with PDT[3]
  Behoben von Marc Rintsch
- 2833139 Saros doesnt work with PDT Eclipse[4]

1 https://twitter.com/saros_plugin
2 http://identi.ca/saros/all
3 http://sourceforge.net/tracker/?func=detail&aid=2849925&group_id=167540&atid=843359
4 http://sourceforge.net/tracker/?func=detail&aid=2833139&group_id=167540&atid=843359

Behoben von Marc Rintsch

Es ist sinnvoll die Programmiersprache des Projektes und ihre Integration in Eclipse sowie deren Kompatibilität mit Saros genau zu kennen, bevor ein Projekt kontaktiert wird. Dies vermeidet, wie im Fall von TYPO3 geschehen, das bestimmte Plugins noch nicht unterstützt werden und die Entwickler Saros nicht realitätsnah testen können.

Um zukünftigen Entwicklern zeigen zu können, welche Programmiersprachen bzw. Plugins unterstützt werden, wurde eine Webseite angelegt, auf der alle getesteten und noch zu testenden Plugins eingetragen sind[1]. Jede zukünftige Kontaktemail wird einen Verweis auf diese Seite enthalten.

Der Einsatz von Eclipse ist im PHP Bereich keine Selbstverständlichkeit, einige Entwickler benutzten andere IDE's wie Netbeans oder einfach Texteditoren. Es ist daher untersuchenswert innerhalb welcher Sprache Eclipse eher Anwendung findet.

Es gab durchaus sehr unterschiedliche Meinungen zu DPP bzw. zu PP. Da verschiedene Entwickler verschiedene Erfahrungen gemacht hatten. (Vgl. Abschnitt A.1.9 und Abschnitt A.1.10) Daher wurde eine Webseite[2] angelegt, welche die Vorteile von PP bzw. DPP für Open Source Software Projekte aufzeigt. Diese kann bei Bedarf angegeben oder zitiert werden. Es ist natürlich klar, dass sich jemand, der schon schlechte Erfahrungen gemacht hat, davon schwer überzeugen lässt. Aber potentiell Interessierte finden hier Argumente und Erfahrungswerte von anderen Entwicklern.

Da TY1 seinen Programmierpartner erst über Twitter suchen musste, kam ich auf die Idee, bei der nächsten Kontaktaufnahme eine Einführungssitzung mit mir anzubieten.

1 https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugins
2 https://www.inf.fu-berlin.de/w/SE/DPPforOSS

Obwohl 2009 Robert Lemke ca. 50% der Arbeit an dem Projekt leistete und nur vier Entwickler beteiligt waren, ist es bemerkenswert, dass es trotzdem zu keiner generellen Ablehung kam.

Es waren weiterhin keine Antworten auf meine Fragen gekommen, weshalb ich hier das Ende der ersten Iteration setzte.

5.1.5.3 Weiteres Vorgehen

Nach dem nächsten Release wird der Defekt 2849925 nicht mehr vorhanden sein. Nach diesem schickte ich noch einmal eine Ankündigung über die Mailingliste und befragte einzelne Entwickler, mit denen ich bereits Kontakt aufgebaut hatte und die Saros testen wollten, noch einmal gezielt. Ich hoffte so auf Resonanz, da der Defekt extra für TYPO3 behoben wurde.

Sollte es weiterhin keine Reaktionen geben, werde ich mich aus diesem Projekt zurückziehen.
Die Liste 4.3 wurde nicht weiter iteriert, da dort nur eine konzeptionelle Diskussion über DPP stattfand und niemand in Erwägung gezogen hatte, Saros zu benutzen.

5.1.6 Zweite Iteration

Die zweite Iteration beanspruchte den Zeitraum vom 14.10.2009 bis zum 03.11.2009. Wie in der ersten bereits vorgeschlagen, informierte ich die Listen typo3-project-5_0-general und typo3-dev über das neue Release (A.1.18, A.1.18).
Leider gab es auf keiner der beiden Listen eine Reaktion. Ich sah daher davon ab, die Entwickler, welche interessiert waren, noch einmal direkt anzuschreiben. Bis zum Ende der 2.Iteration erhielt ich keine weiteren Emails von den TYPO3 Listen.

Ich stieß bei den Vorbereitungen der zweiten Iteration auf das TYPO3 Security Team, welches einige vielversprechende Anwendungsmöglichkeiten für Saros bietet, worauf ich im folgenden genauer eingehen werde.

5.1.6.1 TYPO3 Security Team

Das TYPO3 Security Team verwaltet und veröffentlicht Sicherheitslücken sowohl im TYPO3 Kern als auch in den Erweiterungen. Das Team führt selbst Durchsichten durch, kann aber auch kontaktiert werden, falls ein Benutzer eine Sicherheitslücke findet. Wird das Team auf eine Sicherheitslücke aufmerksam gemacht, sichtet es den betroffenen Quellcode. Sollte sich die Sicherheitslücke im TYPO3 Kern befinden, wird eine Lösung erarbeitet.
Betrifft diese Sicherheitslücke eine Extension so wird der Entwickler dieser kontaktiert und gebeten, diese Lücke zu beheben. Der Entwickler schickt dann einen Patch und die reparierte Version seiner Extension an das Team, welches wiederum Durchsichten ausführt. Wird die Sicherheitslücke nicht behoben oder reagiert der Entwickler nicht, kann das Team die Extension aus dem Repository nehmen und Benutzer auffordern sie zu deinstallieren.

Erkannte Sicherheitslücken werden auf verschiedene Arten veröffentlicht:

- Bulletin[1]
- Mailingliste[2]
- News feed[3]

Diese Vorgehensweise, erkannte Sicherheitslücken sofort zu veröffentlichen und nicht geheim zu halten, wird „Policy of least disclosure[4]“ genannt. Wobei nur die Lücke an sich veröffentlicht wird, nicht wie sie ausgenutzt werden kann, dass heißt es werden keine „proof of concepts“ oder „exploits[5]“ veröffentlicht.

1 http://typo3.org/teams/security/security-bulletins/
2 http://lists.netfielders.de/cgi-bin/mailman/listinfo/typo3-announce
3 http://news.typo3.org/
4 http://secure.t3sec.info/blog/post/2009/07/24/policy-of-least-disclosure-explained/
5 http://de.wikipedia.org/wiki/Exploit

5.1.6.2 Auswahl

Begründung

Im Gegensatz zu normalen Entwicklern ist das Besondere an diesem Team, dass eine ihrer Haupttätigkeiten Durchsichten sind:

> „Once the TYPO3 Security Team receives a notification of an incident, one or more members review it and consider its impact.“ [TYP09d]

Die Durchsichten werden auch von mehreren Personen des Teams übernommen. Dies sah ich als ein mögliches Einsatzsenario von Saros an. Ein weiterer Grund, warum sich Saros für dieses Team und ihre Arbeit besonders eignen könnte ist, dass sie ein sehr eng zusammenarbeitendes Team sind. Man kennt sich persönlich und so spielen eventuelle Berührungsängste eine geringe Rolle:

> „There also exists an security mailing list, used to discuss potential issues, future improvements, etc. That list is internal and only open for TYPO3 Security Team members, personally known to us (we have met all of these people). They are not just reading, they are actively helping us in sorting stuff out, discussing the best solutions, etc.“ [TYP09d]

> „Real-life meetings mainly take place during the TYPO3 Snowboard Tour.“ [TYP09g]

Kontaktaufnahme

Das TYPO3 Security Team hat keine öffentlich zugängliche Mailingliste. Die einzige Möglichkeit Kontakt aufzubauen war über die Emailadresse security@typo3.org. In der Email (A.1.18) habe ich gleich zwei Einsatzmöglichkeiten vorgeschlagen:

1. Ich bot an, eine Durchsicht einer Extension mit jemanden aus dem Team mit Hilfe von Saros durchzuführen. Dies tat ich, weil:
   a) sie selbst auf ihrer Webseite dazu auffordern:

> „If you are interested in donating e.g. for a security review of specific extension, please contact us at security@typo3.org.“ [TYP09h]

b) ich selbst Erfahrungen habe, Erweiterungen für TYPO3 zu programmieren.

2. Das Team könnte unter sich Durchsichten mit Hilfe von Saros durchführen oder zusammen mit einem Extension-Entwickler seine Lösung eines Sicherheitsproblems sichten.

> „We expect you to carefully review your entire extension, not only the particular area where an issue has been discovered. Should you find more issues: Let us now about it.“ [TYP09h]

Resultate

Ich erhielt nach fast drei Wochen vom TYPO3 Security Team eine Absage. Saros schien ihrer Meinung nach für ihren Arbeitsablauf nicht geeignet zu sein. Jedoch beschränkt sich die Ablehnung nur auf das Security Team, persönliches Interesse bestand schon (A.1.24). Hierbei lässt sich jedoch nicht erkennen, ob technische oder konzeptionelle Gründe der Auslöser waren, da diese nicht näher begründet worden sind. Die einzige Einordnung, die sich machen lässt, ist:

1. Ablehnung
   a) Umfeld bezogene Ablehnung
      i. Produkt unpassend für das Projekt
         - Könnte nützlich sein, aber nicht für uns:

> „Your project sounds really interesting. Nevertheless we do not think it would be of great use for our workflows.“ [A.1.24]

5.1.6.3 Fazit der zweiten Iteration

Von den drei TYPO3 Mailinglisten kamen im Rahmen der zweiten Iteration keine Emails, eventuell war die Zeitspanne von fast 20 Tagen zu lang, die zwischen meiner letzten Email der ersten Iteration und dem Anfang der zweiten Iteration. Die Ablehnung von meinem Vorschlag seitens des Security Teams lässt sich nicht eindeutig interpretieren, außer der Tatsache, dass Saros von ihnen als kaum nützlich angesehen wird. Da das TYPO3 Security Team meinen Vorschlag verworfen hat und es sonst keine anderen Reaktionen auf meine Emails gab, sah ich von einer weiteren Iteration ab und beendete die Untersuchung dieses Projektes an dieser Stelle.

5.2 Vuze

5.2.1 Das Projekt

Vuze ist ein in der Programmiersprache Java[1] geschriebener BitTorrent[2] Client. Mit Vuze ist es möglich, Inhalte jeglicher Art zu veröffentlichen und zu nutzen. Diese Inhalte werden anderen Vuze Benutzern in Kanälen, ähnlich einem Fernsehsender, bereit gestellt. Für Vuze stehen eine Vielzahl an Plugins zur Verfügung, welche hauptsächlich von privaten Personen, aber auch von Universitäten oder Firmen, entwickelt werden [Fab10].
Geleitet wird das Projekt von der Firma Vuze Inc. Der Kern der Software ist unter der GPL[3] verfügbar.

5.2.2 Geschichte

Vuze wurde 2003 unter dem Namen Azureus veröffentlicht [Sou10c]. 2006 benannte sich die Firma Azureus Inc. in Vuze Inc um und die Software wurde nun unter dem

1 http://java.sun.com/
2 http://www.bittorrent.com/
3 GNU General Public License

Namen Vuze vertrieben. Die Entwickler hatten das Ziel Azureus in ein soziales Netzwerk, wie z.B. YouTube[4] zu verwandeln, in welchem sowohl die Benutzer als auch Firmen Inhalte veröffentlichen können. Aus diesem Wunsch folgte auch eine komplette Überarbeitung der GUI[5]. Heute ist Vuze einer der verbreitetsten BitTorrent Clients und besitzt mehrere Verträge mit großen Firmen, wie z.B. BBC ([Sou10s] [Ale10]).

5.2.3 Das Team

Das Management von Vuze besteht aus sechs Personen: Olivier Chalouhi - Chief Technology Officer und Initiator des Azureus Projektes 2003, Gilles BianRosa - Chief Executive Officer, Jay Monahan - General Counsel, Milan Parikh - Chief Financial Officer , Chris Thun - VP, Marketing and Product und Christophe - Gillet Director Of Product. [Vuz10]
Die Sourceforge Projekt Administratoren sind: Olivier Chalouhi, Alon Rohter, Paul Gardner und Arron M. [Sou10p]

Hierbei ist zu sehen, dass es ein reines Management-Team gibt, welches sich hauptsächlich um die Belange der Firma kümmert. Allerdings ist der Initiator von Vuze sowohl im Management (Chief Technology Officer), als auch an der aktiven Entwicklung der Software beteiligt.

5.2.4 Projektauswahl

5.2.4.1 Begründung

Als zweites Projekt habe ich bewusst ein Projekt ausgewählt, das sich sehr von dem TYPO3 Projekt unterscheidet. Im folgenden werden die am interessantesten Unterschiede genannt.

1. Der Quellcode wird zwar öffentlich zur Verfügung gestellt, aber im Gegensatz zu TYPO3 wird Vuze von einer Firma entwickelt.

4 http://www.youtube.com/
5 Grafische Benutzeroberfläche, engl. Graphical User Interface

2. Daraus resultiert wahrscheinlich auch der öffentlich nicht einsehbare Entwicklungsprozess. Auf diesen Punkt wird im nächsten Abschnitt genauer eingegangen.

3. Die verwendete Programmiersprache ist nicht PHP, sondern Java.

Ein weiterer Grund war, dass im Repository eine .projekt Datei war. Diese Datei wird automatisch von Eclipse angelegt, sobald ein neues Projekt eröffnet wird. Daher ist es sehr wahrscheinlich, dass bei Vuze mit Eclipse gearbeitet wird. Ein von einer Firma kontrolliertes Projekt kann, wie in 2.2.6.2 erwähnt, ein anderes Verhalten gegenüber externen Anfragen an den Tag legen, als Projekte die gemeindebasiert sind. Von Interesse ist hierbei, ob auf meine Anfrage offener oder verschlossener reagiert wird.

Wie in 5.1.5.2 schon erwähnt, ist es untersuchenswert bei welchen Programmiersprachen Eclipse häufiger Anwendung findet. Meine Vermutung war, das Eclipse in der Java Welt verbreiteter ist, als in der PHP Welt, da Eclipse der Nachfolger von IBM Visual Age for Java 4.0 ist und somit ursprünglich eine reine Java IDE darstellte [Ull08].

5.2.4.2 Kontaktaufnahme

Bei Vuze findet der Entwicklungsprozess nicht öffentlich statt, das heißt, Diskussionen und Entscheidungen können weder eingesehen, noch kann partizipiert werden. Vuze verfügt demnach weder über eine öffentliche Mailingliste, noch über einen Bugtracker. Die einzige Möglichkeit sich mit diesem Projekt in Verbindung zu setzen, war entweder über das Management oder über die bei Sourceforge eingetragenen Projekt-Administratoren. Ich entschied mich, dass Projekt über die Administratoren zu kontaktieren, da diese höchst wahrscheinlich an der Entwicklung von Vuze aktiv beteiligt sind. Die Administratoren wurden jeweils über das von Sourceforge bereitgestellte Kontaktformular angeschrieben (A.1.14).

5.2.5 Resultate

Von den vier Administratoren reagierte, trotz mehrerer Versuche, keiner auf meine Email.

5.2.6 Fazit

Es fiel schwer ein Fazit zu ziehen, da keine Resultate erzielt wurden. Es lassen sich aber einige Thesen aufstellen, warum Vuze nicht reagierte:

- Die Kontaktaufnahme über Maintainer scheint weniger erfolgversprechend als die Entwickler, bzw. das gesamte Projekt per Mailingliste direkt anzuschreiben.
- Saros scheint für Projekte welche von einer Firma geleitet werden und/oder keine öffentlichen Entwicklungsprozesse betreiben weniger geeignet oder
- solche Projekte könnten generell für äußere Anfragen weniger offen sein.

Diese Vermutungen sollen beim nächsten Projekt genauer untersucht werden. Ein iteratives Vorgehen war bei diesem Projekt nicht möglich.

5.2.7 Weiteres Vorgehen

Da hier kein klares Fazit aus dieser Kontaktaufnahme gezogen werden konnte, erscheint es weiterhin sinnvoll, sich mit Java Projekten in Verbindung zu setzen, sowie den Kontaktversuch über die Maintainer zu evaluieren. Nur so lässt sich herauszufinden, ob es sich bei Vuze um einen Einzelfall handelt oder ob sich empirisch nachweisbar zeigen lässt, dass es sich bei der Nichtreaktion um eine typisches Verhalten dieser Art von Projekten handelt.

5.3 Freenet

5.3.1 Das Projekt

Freenet ist eine in Java geschriebene Peer-to-Peer Software (vgl. [Sch02]), die es ermöglicht, ein anonymes, dezentralisiertes Netzwerk über das Internet aufzubauen. Das Netzwerk ist angelegt wie das Word Wide Web, als Transportschicht wird das HTTP[1] eingesetzt. Es ist möglich Seiten, so genannte Freesites, und Inhalte zu publizieren, die dann mit einem üblichen Webbrowser gesichtet werden können. Ziel von Freenet ist es, Menschen die Möglichkeit zu geben, ihre Meinung frei zu äußern und sich informieren zu können, ohne Zensur oder Repressalien zu befürchten [fre09b].

5.3.2 Geschichte

Freenet basiert auf dem Paper „A Distributed Decentralised Information Storage and Retrieval System“ von Ian Clarke das er 1999 veröffentlichte [Cla99]. Kurze Zeit danach begannen Clarke und ein paar Freiwillige diese Idee umzusetzen. Im März 2000 wurde die erste Version von Freenet unter der GPLv2[2] veröffentlicht. Durch Spendengelder[3] war es dem Projekt möglich, die gemeinnützige Firma „The Freenet Project Inc.“ zu gründen und Matthew Toseland als Vollzeitprogrammierer zu beschäftigen. Im Moment ist Freenet in der Version 0.7.5 verfügbar.

5.3.3 Das Team

Das Projekt besteht aus zehn Entwicklern, mit teilweise unterschiedlichen Aufgabenfeldern, wobei sie darauf hinweisen, dass hunderte von Menschen seit dem Projektstart an Freenet mitgearbeitet haben [fre09a]. Ian Clarke führt das Projekt nach wie vor an.

1 Hypertext Transfer Protocol
2 GNU General Public License Version 2
3 http://freenetproject.org/sponsors.html

5.3.4 Projektauswahl

5.3.4.1 Begründung

Das Projekt wurde zeitgleich mit dem Projekt Vuze kontaktiert. Zum einen wurde es ausgewählt, da es ebenfalls in der Programmiersprache Java geschrieben ist und zum anderen, da es ein gemeindebasiertes Projekt ist, welches von einer kleinen Gruppe initiiert wurde. Es handelt sich genau wie bei Vuze um eine Peer-to-Peer Software. So konnte ich zwei ähnliche Java Projekte untersuchen, die aber unterschiedlich organisiert sind. Freenet ähnelt von der Organisationsstruktur und der Entstehungsgeschichte her eher TYPO3 als Vuze, wenn auch Freenet ein wesentlich kleineres Projekt ist. Ein weiterer Grund war, dass im Repository von Freenet eine .projekt Datei vorhanden war, was darauf hinweist, dass Eclipse zum Programmieren benutzt wird.

5.3.4.2 Kontaktaufnahme

Freenet betreibt insgesamt sieben Mailinglisten, wobei sich zwei thematisch für eine Kontaktaufnahme eigneten: Die Tech und die Development Liste. Bei genauerer Betrachtung erschien mir aber die Development Liste erfolgversprechender, da diese zum Entwickeln von Freenet benutzt wird und alle Entwickler diese Liste abonniert haben. Am 05.10.2009 kontaktierte ich das Projekt über die Mailingliste: devl@freenetproject.org (A.1.16). Diese Liste hatte eine Aktivität von ca. 6,5 Emails pro Tag und war so vergleichbar mit der TYPO3 Mailingliste typo3-project-4-3. Die Möglichkeit nur Ian Clarke zu kontaktieren verwarf ich, weil durch die gleiche Kontaktart wie bei TYPO3, eine bessere Vergleichbarkeit mit diesem Projekt besteht, zudem ist er ebenfalls Mitglied der Mailingliste.

5.3.5 Resultate

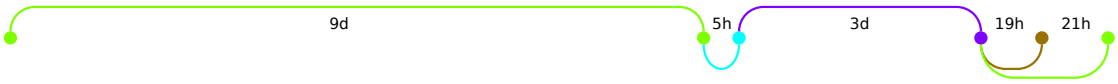

Abbildung 5.8: Emailverkehr auf der Freenet-Mailingliste: FR1, FR2, FR3

Auf meine erste Email bekam ich keine Reaktion, so dass ich neun Tage später einen zweiten Anlauf unternahm, siehe 5.8 (A.1.18). Daraufhin meldete sich FR1 und meinte, dass Saros im Freenet nicht angewendet werden könne, da die Entwickler sehr verteilt über die Welt arbeiten und es selten vorkommt, dass mehrere Entwickler zur gleichen Zeit an Freenet arbeiten (A.1.18).
Daraufhin machte FR2 drei Tage später den Vorschlag, dass es nützlich sei, in ihrem Repository die Zeilen farblich zu markieren, welche schon gesichtet worden sind:

> „Though it would be useful if we had the capability to mark code as reviewed line-by-line in the git repository. Imagine some color-coded display of the source code: The more people have reviewed a certain line of code, the more it becomes dark green. Not reviewed code would be red.“ [A.1.20]

Ich antwortete ihm, dass dies mit Saros nicht möglich sein, da dies in die Funktionalität ihres Git[1] Repository fällt (A.1.22). FR3 wies darauf hin, dass ein Werkzeug, welches kollaborative Quelltextdurchsichten ermöglicht, schon einmal aufgesetzt worden sei, aber niemand von Freenet es benutzt habe. Zusammengefasst ergeben sich folgende Resultate:

1. Ablehnung
   a) Konzept unpassend für das Projekt
      - Entwickler arbeiten zu asynchron

        > „[...]most of the development does not occur in parallel by multiple developers in real time.“ [FR1 A.1.18]
        > „But anything which requires live interaction between developers won't work, we all work at different times of day.“[FR2 A.1.20]
        > „[...]this sort of thing is probably more useful in environments

1 http://git-scm.com/

where people are in close physical and temporal proximity to each other (eg. an office).“ [FR1 A.1.18]

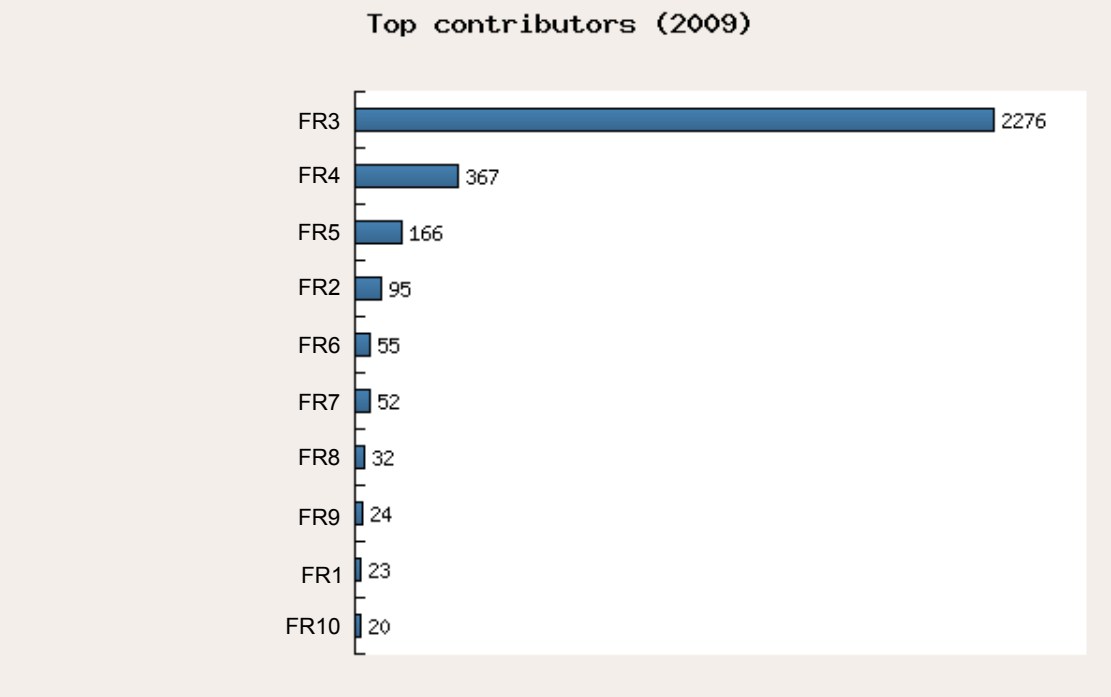

Abbildung 5.9: Freenet: Die zehn häufigsten Commits 2009

In Abb. 5.9 sind zehn Entwickler zu sehen, die im Jahr 2009 die meisten Commits gemacht haben. Dabei wird klar, dass Matthew Toseland, als Angestellter des Projektes, die Hauptprogrammierarbeit leistet.

Abb. 5.10 zeigt, an welchen Wochentag und zu welcher Uhrzeit, von 2005-01-21 bis 2010-02-14, die meisten Commits gemacht wurden. Zu sehen ist hierbei, dass die meisten Commits zwischen 15-20 Uhr und 23-01 Uhr, jeweils von Dienstag bis Samstag durchgeführt wurden.

Rückschlüsse auf die geografische Verteilung der Entwickler liefert Abb. 5.11, da die Anzahl der Commits aus den verschiedenen Zeitzonen dargestellt werden. Die meisten Commits wurden in der Mitteleuropäischen Zeitzone durchgeführt.

Weekday	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	27	30	16	18	24	15	20	10	15	12	18	27	33	55	77	101	63	92	79	89	96	109	91	112
2	140	115	55	34	14	13	40	17	25	29	14	27	33	74	144	144	155	217	227	199	171	159	117	106
3	196	114	75	70	9	7	8	22	39	13	42	36	42	78	153	135	177	201	242	201	279	191	141	143
4	187	195	91	47	15	12	21	10	13	28	23	32	50	111	150	166	146	192	188	193	255	199	144	197
5	226	121	96	31	29	12	5	18	21	26	47	21	54	92	195	242	200	236	260	276	257	206	185	234
6	214	169	109	57	28	6	11	5	24	13	15	27	46	89	97	176	231	220	203	239	196	250	203	145
7	132	74	24	18	22	10	15	6	6	18	16	45	42	51	43	68	32	56	46	58	47	43	40	44

Abbildung 5.10: Freenet: Verteilung der Commits über Uhrzeit und Wochentage

Timezone	Commits
-0800	5
-0700	2
-0500	11
-0400	48
+0000	13997
+0100	1222
+0200	176
+0530	8
+0800	127

Abbildung 5.11: Freenet: Verteilung der Commits auf die Zeitzonen

5.3.6 Fazit

Anhand der Resultate ist zu sehen, dass das Konzept von DPP in diesem Projekt nicht anwendbar zu sein scheint, da die Entwickler zu asynchron an dem Projekt arbeiten. Dies kann mehrere Gründe haben. Einerseits kann die Projektgröße von zehn Entwicklern zu klein sein, um sich in Echtzeit zusammen zu finden. Andererseits waren bei TYPO3 5.0 nur fünf Entwickler angegeben und es kam nicht zu einer derartigen Ablehnung.
Eine weitere Möglichkeit wäre die geografische Verteilung der Entwickler. Wenn die Freenet-Entwickler in sehr unterschiedlichen Zeitzonen leben, kann es schwierig sein zur selben Zeit an dem Projekt zu arbeiten. Abb. 5.10 und Abb. 5.11 lassen vermuten, dass dies nicht der Fall ist, jedoch sind diese Statistiken stark beeinflusst durch die hohe Anzahl an Commits von Matthew Toseland. So werden wahrscheinlich

hauptsächlich seine Arbeitszeiten und seine Zeitzone wiedergeben.
Darin liegt, denke ich, der eigentliche Grund, warum die Entwickler behaupten, dass sie zu asynchron für Saros arbeiten: Die meiste Arbeit wird von nur einer Person durchgeführt, d.h. diese ist ungleich über die Entwickler verteilt. So besteht für die anderen Entwickler eine geringere Wahrscheinlichkeit gleichzeitig an dem Projekt zu arbeiten. Wenn diese auch noch geografisch sehr verteilt arbeiten, wird diese Wahrscheinlichkeit noch geringer.

5.3.7 Weiteres Vorgehen

Durch die klare Absage von den Entwicklern sah ich von einer zweiten Iteration ab.

5.4 Drupal

5.4.1 Das Projekt

Drupal ist ein webbasiertes Content Management System, welches unter der General Public Licence veröffentlicht wird. Die verwendete Programmiersprache ist PHP. Drupal legt seinen Schwerpunkt auf den Aufbau von Sozialen Netzwerken und Communities. Ähnlich wie TYPO3 besteht Drupal aus einem Kern, welcher die Hauptfunktionalitäten liefert, zusätzlich können aber weitere Module installiert bzw. programmiert werden [dru09a].

5.4.2 Geschichte

Dries Buytaert begann 1998 mit der Entwicklung von Drupal. Ursprünglich war Drupal als Message Board für ein kleines Netzwerk von Freunden gedacht, die sich einen Internetanschluss per Wireless LAN teilten. Dries Buytaert studierte zu der Zeit an der Universität Ghent in Belgien. Nachdem er sein Studium abgeschlossen hatte, entschied er sich, sein Message Board unter der Domain drop.org ins Internet zu stellen. Im Zeitraum 2000-2001 entstand eine größere Benutzergemeinde, welche kontinuierlich neue Ideen und Verbesserungen lieferte. Damit die Benutzergemeinde

ihre Ideen für Drupal selbst verwirklichen konnte, stellte Dires seine Software unter dem Namen Drupal am 26.April 2001 als Open Source zur Verfügung [dru09b].

5.4.3 Das Team

Dries Buytaert ist heute noch Projektleiter und permanenter Core-Committer. Es wird in Drupal zwischen zwei Gruppen unterschieden:

1. Core Committer
   Ein kleines Team, das Änderungen an dem Kern von Drupal vornehmen darf und somit auch Schreibrechte auf das CVS[1] Repository hat. Zu diesem Team gehören:
   - Branch Maintainer
     Von dieser Gruppe wird jeweils eine Person ausgewählt, die dann verantwortlich für das aktuelle Release ist. Sie besteht aus fünf Mitgliedern.
   - Maintainer
     Sie werden von Dries ausgewählt und können keine eigenen Entscheidungen treffen, sind aber verantwortlich für kleine Teile des Kerns. Diese Gruppe besteht aus 26 Mitgliedern, wobei es Überschneidungen mit den Branch Maintainern gibt.
2. Core contributer
   Sie schicken hauptsächlich Patches oder schreiben Dokumentationen für den Kern, wobei es jedem Außenstehenden möglich ist, Patches für Drupal zu schreiben.

5.4.4 Projektauswahl

5.4.4.1 Begründung

Drupal wurde ausgewählt als es keine Antworten von Freenet gab. Da TYPO3 bis zu diesem Zeitpunkt die meisten Resultate lieferte, entschloss ich mich ein

1 Concurrent Versions System

ähnliches Projekt zu kontaktieren. Drupal ist sowohl technisch als auch von der Organisationsstruktur und Geschichte mit TYP03 vergleichbar. Da Drupal ebenfalls PHP als Programmiersprache benutzt, mussten keine neue Plugins mit Saros getestet werden und es konnte die These von TYPO3, dass Eclipse im PHP-Bereich wenig Anwendungen findet, untersucht werden. Eine Suche nach dem Wort „Eclipse“ auf der Entwickler-Mailingliste ergab allerdings 60 Treffer, so dass vermutlich einige Entwickler doch mit Eclipse arbeiten.

5.4.4.2 Kontaktaufnahme

Das Projekt kontaktierte ich über die Mailingliste. Drupal hat insgesamt elf Mailinglisten, wobei nur die Liste development@drupal.org thematisch für die Kontaktaufnahme in Frage kam, da sich die anderen z.B. nur um die Webseite oder Dokumentation kümmern. Über die Liste wurde in der Zeit der ersten Iteration 115 Emails verschickt, sie hatte somit eine Aktivität von 6,3 Emails pro Tag und wies dementsprechend eine ähnliche Größe wie die TYPO3-4-3 und Freenet-Liste auf.

5.4.5 Erste Iteration

Die erste Iteration beanspruchte den Zeitraum von 14.10.2009 bis 01.11.2009

5.4.5.1 Resultate

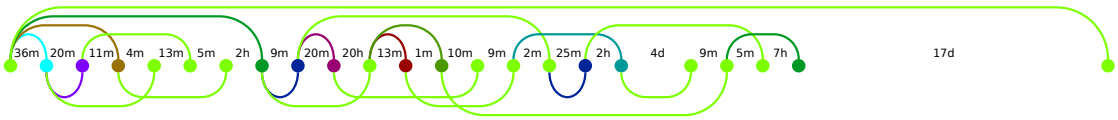

Abbildung 5.12: Emailverkehr auf der Drupal Mailingiste: DR1, DR2, DR3, DR4, DR5, DR6, DR7, DR8

Wie in Abb. 5.12 zu sehen ist, kam die erste Antworte nach 36 Minuten. Danach folgten weitere Emails und es entwickelte sich eine Diskussion an der neun Entwickler teilnahmen, welche hauptsächlich in den ersten beiden Tagen nach der Kontaktaufnahme stattfand. Meine Anfrage wurde positiv aufgenommen und man bedankte sich, dass ich die Liste auf Saros aufmerksam gemacht habe:

„Eike, this sounds pretty neat. Thanks for bringing it to our attention.“ [A.1.19]

DR1 antwortete als Erster und stellte ein paar Nachfragen (A.1.18):

- Ist der Overhead kleiner als bei VNC[1]?
- Werden Dateien lokal gespeichert?
- Führt das lokale Speichern nicht zu Problemen mit der Versionsverwaltung?

DR2 antwortete darauf, dass der Texteditor Emacs[2] eine ähnliche Funktionalität liefert wie Saros und dass immer nur eine Person in die Versionsverwaltung hoch laden sollte, da es sonst zu Problemen kommt (A.1.18).
Ich stellte daraufhin klar, dass es egal ist, wer nach einer Session das Projekt hochlädt, vorausgesetzt alle Entwickler bleiben bis zum Ende in der Session, da ja alle das gleiche Projekt lokal besitzen und es keine Unterschiede in den Dateien gibt.
DR3 sah Verbindungen zu Cola, einem Teil des Eclipse Comunication Frameworks[3], welches ähnliche Funktionalität wie Saros liefert und wollte wissen, ob es eine Verbindung von Saros zu diesem Projekt gibt. Auf meine Antwort, dass es dort keine Verbindung gäbe und ob er schon einmal mit ECF[4] gearbeitet hätte erhielt ich leider keine Antwort. Die Diskussionen endeten, wie in Abb. 5.12 zu sehen, recht schnell nach einer Antwort von mir. Nur aus der Reaktion von DR4 entwickelte sich eine längere Diskussion.
DR4 gab zu bedenken, dass es in Drupal keine Instanz gebe, welche den Entwicklern vorschreiben könne, Saros zu benutzen, außerdem würden die meisten Entwickler sehr verteilt und mit unterschiedlichen Programmen arbeiten, so dass sie keine Möglichkeiten hätten, zur gleichen Zeit zusammen zu finden. Er sprach aber Saros eine generelle Nützlichkeit nicht ab.

1 Virtual Network Computing
2 http://technomancy.us/129
3 http://www.eclipse.org/ecf/
4 Eclipse Comunication Framework

> „That said, I'm sure such a tool would prove useful to certain developers if they choose to use it.“ [A.1.18]

Ferner wollte er wissen, ob Saros abhängig von Programmiersprachen ist, d.h. ob Saros zum Beispiel nur mit Java funktioniert.

DR5 gab ihm recht und beschrieb einen typischen Entwicklungsablauf von Drupal, in welchem seiner Meinung nach Saros keinen Platz findet:

> „Most of the Drupal development that I've seen is either: A) Written by one person, then reviewed / improved asynchronously in the issue tracker rather than working at the same time. B) Written by a team of people for a project in-house, in which case it would be the same as basically any other in-house PHP development. [A.1.18]“

Er selbst benutze zwar Eclipse, würde aber Saros nur im Rahmen von Teleheimarbeit[1] benutzen. Jedoch gebe es seiner Meinung nach bei einigen Modul- und Kernentwicklern durchaus Anwendungsmöglichkeiten von Saros, wenngleich es auch keinen generellen Bedarf gäbe.
DR6 widersprach dieser Aussage und war der Meinung, dass Drupal ein Werkzeug wie Saros unmittelbar einsetzen könne. Er lieferte gleich die passende Anwendungsmöglichkeit mit:

> „With code freeze 38 hours away, efficiency is key to getting the patches from the exception list into core. If more than one person is working on a patch, it really makes sense to use something like this and have everyone looking at the same code at the same time so that it's not a constant string of upload to the issue queue, have a minor bug pointed out, reroll with bug fixes.“ [A.1.18]

Durch eine Untersuchung des Repository versuchte ich eine Übereinstimmung der Daten mit den oben zitierten Aussagen zu finden. Die Auswertung ergab, dass nur zwei Maintainer im Jahr 2009 Commits durchgeführt haben (siehe Tabelle 5.2). In

1 http://de.wikipedia.org/wiki/Telearbeit

Entwickler	Codezeilen	Änderungen
dries	5142 (58.8%)	101946 (56.9%)
webchick	3598 (41.2%)	77065 (43.1%)

Tabelle 5.2: Codezeilen und Änderungen der Entwickler am Projekt Drupal im Jahr 2009

Abb. 5.13 ist zu sehen, dass die beiden Maintainer nur geringe Überschneidungen in ihren Arbeitszeiten haben. Dries scheint hauptsächlich in der Zeitspanne von 24 bis 6 Uhr zu arbeiten, Webchick arbeitet überwiegend von 23 bis 14 Uhr und von 8 bis 0 Uhr. Überschneidungen könnte es also zwischen 23 und 6 Uhr geben, jedoch lässt die Grafik vermuten, dass sich die Arbeitstage ebenfalls unterscheiden. Dies könnte auch die gleichmäßige Verteilung der Commits in den Abbildungen 5.15 und 5.14 erklären.

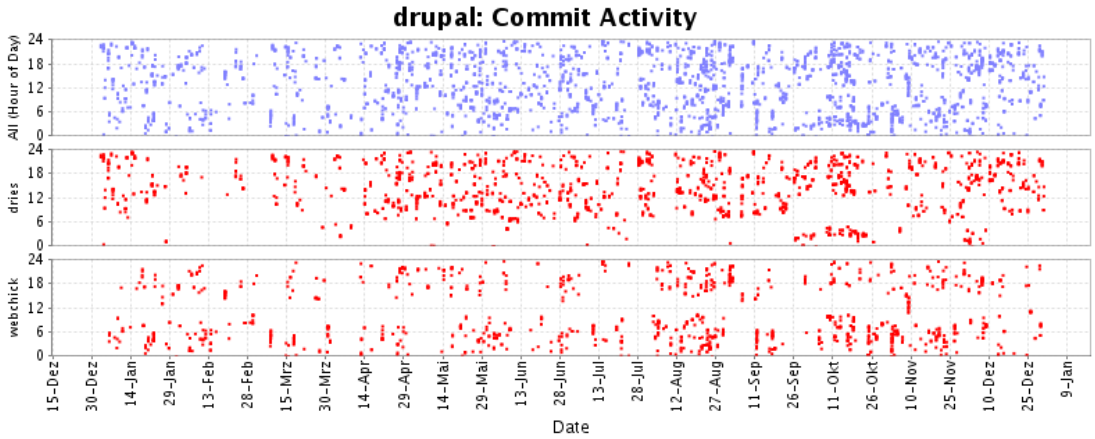

Abbildung 5.13: Die im Jahr 2009 durchgeführten Commits der Drupal Entwickler

DR7 war der Meinung, dass sich Saros besonders zum Anlernen von Neulingen oder zur Beratung eignet (A.1.19). DR9 stimmte dem zu und ergänzte, dass Saros sich zur Einführung in Drupal eignen könne. Er benutze sonst Zeilennummern, um sich zu orientieren sobald über Quelltext diskutiert wird. Saros könnte dies seiner Meinung nach vereinfachen. Er wollte Saros auf jeden Fall ausprobieren (A.1.19). DR8 war hingegen der Meinung, dass Saros zwar nützlich sei, aber nur in sehr

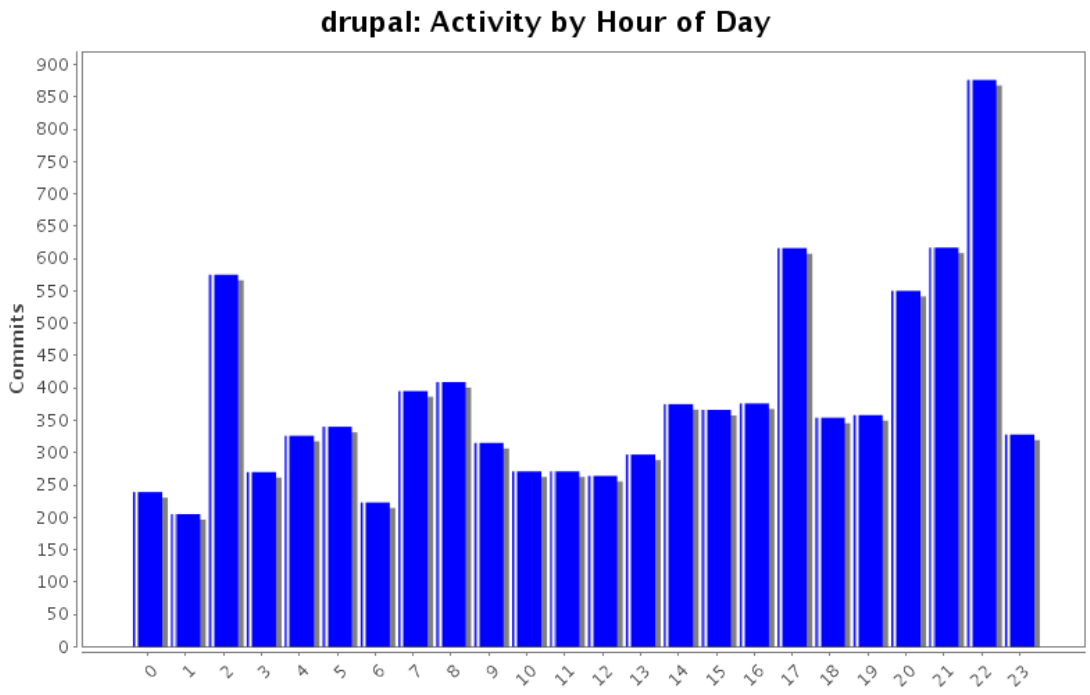

Abbildung 5.14: Drupal: Verteilung der Commits über die Uhrzeiten im Jahr 2009

begrenzten Situationen und sich für Drupal nicht wirklich eigne, obwohl in Drupal und anderen Open Source Projekten durchaus zusammen in Echtzeit gearbeitet werde. Dies passiere jedoch meistens via IRC[1]. Zum Quelltextaustausch werde der betreffende Text einfach in den Chat hineinkopiert.

> „Having said that the irc paste combo is used exactly in the manner you suggest[kollaboratives Arbeiten in Echtzeit]. Not just by the drupal crowd, but by most projects I've touched“ [A.1.19].

Daher räumte er Saros keine großen Anwendungsfelder ein, zumal viele Entwickler schon in Echtzeit zusammenarbeiten, dafür aber ihre eigenen Lösungen haben und die Entwickler nicht von einer Plattform oder IDE abhängig sein wollen.

1 Internet Relay Chat

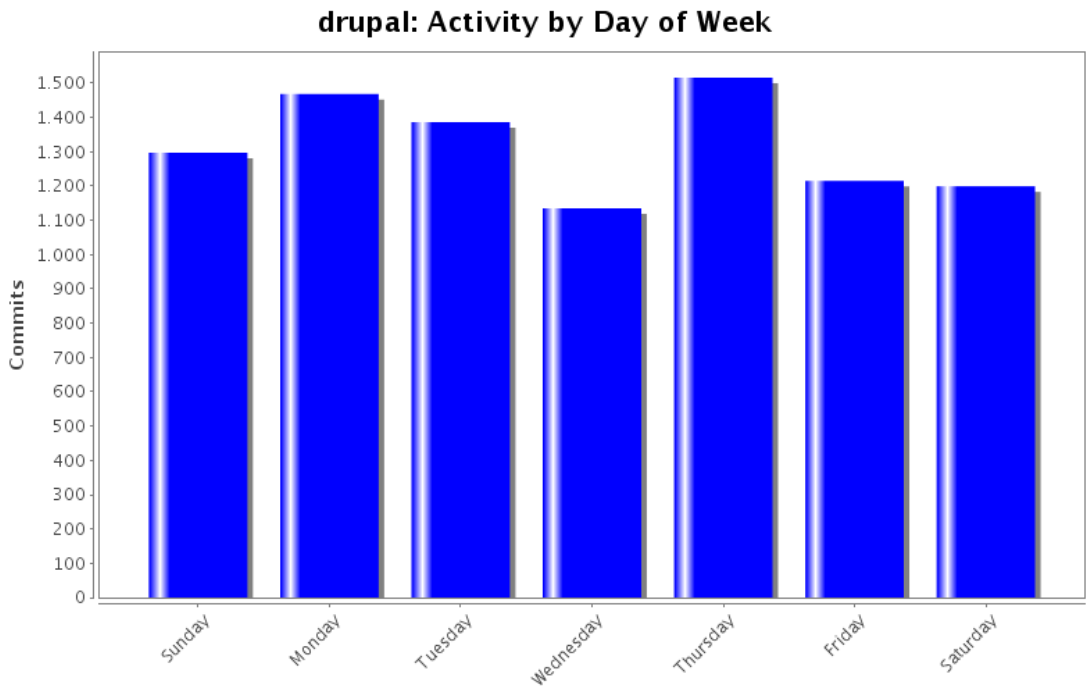

Abbildung 5.15: Drupal: Verteilung der Commits über die Wochentage im Jahre 2009

> „Is there space for this tool - yes. Is it big? No. Until you can collaborate with others - eclipse, emacs, textmate, put your favourite environment, it is not going to be attractive. People don't like to be dependent on a specific platform. Everyone has their own“ [A.1.19].

DR5 teilte diese Meinung, sprach zusätzlich noch an, dass es vor dem kommenden Release zu wenig Zeit gebe und diese besser genutzt werde um an Drupal zu arbeiten, als sich in neue Werkzeuge einzuarbeiten (A.1.19). Darauf antwortete ich, dass Saros z.B. nur für Durchsichten eingesetzt werden kann und jeder Entwickler trotzdem weiter in seiner favorisierten Entwicklungsumgebung weiterentwickelt. DR4 antwortete darauf, das Eclipse an sich nicht besonders nützlich zum Entwickeln von Drupal ist, daher glaubte er auch nicht, dass Entwickler nur für Saros Eclipse benutzen.

> „I can't see anyone using Eclipse just for Saros and other for everything

else“ [A.1.22].

Insgesamt gesehen ergibt sich folgendes Bild unter Verwendung des Kategoriensystems:

1. Ablehnung
   a) Technische Ablehnung
      i. Kein sinnvoller Vorschlag
         - Es gibt schon so etwas:

           „Not to rain on someone else’s parade but you can do the same in Emacs:“ [A.1.18]
      ii. Mangelhaftes Produkt
         - Abhängigkeit von einer Entwicklungsumgebung (Eclipse):

           „People don’t like to be dependent on a specific platform. Everyone has their own.“ [DR8 A.1.19]
           „At the moment I’m using Eclipse. Really, it is NOT a casual-use tool. Especially for Drupal, it’s not worth it unless you adopt it and take the time to customize it. I can’t see anyone using Eclipse just for Saros and other for everything else.“ [DR4 A.1.22]
   b) Umfeld bezogene Ablehnung
      i. Produkt unpassend für das Projekt
         - Muss jeder selber entscheiden:

           „[...]there is no Drupal Team that could decide en masse to use a tool like this.“ [DR4 A.1.18]
         - Es wird schon so etwas benutzt:

           „Having said that the irc paste combo is used exactly in the manner you suggest[kollaboratives Arbeiten in Echtzeit]. Not just by the drupal crowd, but by most projects I’ve touched.“ [DR8 A.1.19]

- Zu wenige Entwickler würden das benutzen:

  „Some other module / core contributors might find it useful for those tasks, but it doesn't seem like a common need.“ [DR5 A.1.18]
  „It might work in some cases, but I suspect it won't reach wide adoption.“ [DR5 A.1.19]

- Zu große Einarbeitungszeit:

  „Time is very tight, so any delays messing with connection problems, etc. aren't really acceptable. So, both parties would need to have already used it in the past and frequently for it to be usable during crunch time.“ [DR5 A.1.19]

- Das Produkt könnte nützlich sein, aber nicht für uns.

  „I probably won't ever use something like Saros unless I got a tele-working position somewhere.“ [DR5 A.1.18]
  „The tool is usable, in some limited fashion - project teams etc... It is not a Drupal tool“ [DR8 A.1.19]

ii. Konzept unpassend für das Projekt

- Entwickler arbeiten zu asynchron:

  „Drupal is so distributed that we all have our own development workflows using a variety of tools, often in our basements rather than our offices“ [DR4 A.1.18]
  „Most of the Drupal development that I've seen is either: A) Written by one person, then reviewed / improved asynchronously “ [DR5 A.1.18]

- Entwickler arbeiten ohnehin schon in einem Büro zusammen:

  „Most of the Drupal development that I've seen is: [...]B) Written by a team of people for a project in-house, in which case it would be the same as basically any other in-house PHP development“ [DR5 A.1.18]

2. Akzeptanz
   a) Technische Akzeptanz
      i. Sinnvolles Produkt
         - Interesse/sinnvoller Vorschlag:

           „Sounds very interesting :)“ [DR2 A.1.18]

           „That said, I'm sure such a tool would prove useful to certain developers if they choose to use it.“ [DR4 A.1.18]
           „Eike, this sounds pretty neat.“ [DR7 A.1.19]
         - Schon nach so etwas gesucht:

           „I was just looking for something like this!“ [DR6 A.1.18]
         - Anwendungsvorschlag:

           „It would also be a good tool for training and consulting,where you could save a trip to the customer's office and work with them on code directly.“[DR7 A.1.19]
   b) Umfeld bezogene Akzeptanz
      i. Produkt passend für das Projekt
         - Generell von Nutzen für das Projekt:

           „As far as Drupal goes, I think that something like this would be particularly useful for core in a time like...oh,say....now.“ [DR6 A.1.18]
         - Anwendungsvorschlag für das Projekt:

           „With code freeze 38 hours away, efficiency is key to getting the patches from the exception list into core. If more than one person is working on a patch, it really makes sense to use something like this [...] and have everyone looking at the same code at the same time so that it's not a constant string of upload to the issue queue, have a minor bug pointed out, reroll with bug fixes. Two sets of eyes on any of those

patches at any one time would be awesome, and I think it would help get them ready to be committed.“ [DR6 A.1.18]
„This is exactly what I find myself doing with my developers that are new to the drupal way I plan on installing this today and integrating it into my eclipse workflow for a test run. [...] It’s a novel solution to an old problem of communicating via line numbers :)“ [DR9 A.1.19]

3. Unklarheiten
   a) Technische Nachfragen
      i. Performance
         - Bandbreite:

           „What’s the overhead like? Would it be smaller than that of VNC[...]?“ [DR2 A.1.18]

      ii. Kompatibilität
         - Versionsverwaltungen:

           „When both parties save an edited file, does it get stored in the local workspace as usual? Wouldn’t it cause conflicts if you are using version control?„ [DR2 A.1.18]

         - Programmiersprachen:

           „Is it language-dependent?“ [DR4 A.1.18]

      iii. Produktvergleich
         - Zusammenhang mit andern Projekten:

           „This sounds very similar to http://codesurgeonblog.com/2008/06/cola-real-time-shared-editing.html which was demonstrated last year. How is Saros related to that effort?“ [DR3 A.1.18]

5.4.5.2 Fazit der ersten Iteration

Es war mir nicht möglich Saros in Drupal zu integrieren, es gab insgesamt nur zwei Entwickler die Saros testen wollten und von denen bekam ich keine weitere Rückmeldung. Interessant war, dass es durchaus eine Nachfrage zum kooperativem Arbeiten in Echtzeit gibt und dieses auch schon zum Teil praktiziert wird. Daher erlaubt die Arbeitsweise einiger Drupal-Entwickler durchaus eine Nutzung von Saros. Ferner besteht auch Interesse an solchen Werkzeugen. Besonders die erwähnte Zusammenarbeit per IRC kommt der mit Saros nahe, da:

- in Echtzeit zusammengearbeitet wird, das heißt beide/mehrere Entwickler sind zur selben Zeit am Computer.
- der Quelltext ausgetauscht und diskutiert wird.
- mit mehreren an einer bestimmten Stelle gearbeitet wird.

Es wurden folgende Anwendungsmöglichkeiten von Saros vorgeschlagen:

1. Durchsichten von Patches, ob diese bei dem nächsten Release in den Kern kommen sollen. Es gibt bei Drupal einen Feature Freeze vor jedem Release, das heißt es werden keine neuen Features für den Kern mehr angenommen und nur noch Reparaturen gemacht. Diese Reparaturen sind oft schon als Patches im Bugtracker vorhanden. Die Kernentwickler entscheiden nun, welche Reparaturen noch mit in den Kern aufgenommen werden, dazu müssen sie jeden in Frage kommenden Patch sichten (A.1.18).
2. Mit der Hilfe von Saros können neue Entwickler direkt durch den Quelltext von Drupal geführt und anhand dessen Erklärungen gemacht werden.
3. Man kann Kunden bestimmte Abläufe im Quelltext erklären (A.1.19).

Es gab aber auch einige Meinungen, welche nicht viel Nutzen in Saros sahen. Dazu wurden verschieden Argumente geäußert:

1. Mehrere Entwickler sahen es als problematisch an, dass sie zu verteilt arbeiten, sowohl räumlich als auch zeitlich. Wobei die Aussage betreffend der Zusammenarbeit per IRC diesem Argument widerspricht.

2. Sie haben zu wenig Zeit, um sich in neue Werkzeuge einzuarbeiten.
3. Eclipse ist nicht beliebt als Entwicklungsumgebung und die Zusammenarbeit kann auch mit anderen Werkzeugen erreicht werden, ohne abhängig von Eclipse zu sein.

An Nachfragen bzw. Unklarheiten traten folgende Punkte auf:

1. Gibt es Probleme mit der Versionsverwaltung, wenn Saros benutzt wird?
2. Ist Saros abhängig von Programmiersprachen, bzw. mit welchen Sprachen ist Saros kompatibel?
3. Benötigt Saros eine ähnliche Bandbreite wie VNC?

Diese Fragen werde ich in die FAQ aufnehmen, damit zukünftige Benutzer sich besser über Saros informieren können.
Einige Entwickler äußerten sich über den Entwicklungsablauf innerhalb von Drupal. Interessanterweise haben aber nur zwei Entwickler dieses Jahr Commits durchgeführt, so dass sich die Frage stellte, ob sie früher aktiver an der Entwicklung von Drupal beteiligt waren, oder ob sie nicht aktiv an dem Kern mitarbeiten und ihre Aussagen nur die Entwicklung von Erweiterungen betreffen. Wie Abb. 5.16 zeigt, wurden seit ca. März 2008 ausschließlich Codezeilen von Dries und Webchick hinzugefügt. Eventuell ist aber auch die Commit-Politik geändert worden ist, so dass nur noch Dries und Webchick Commit-Rechte haben. Hierfür lassen sich aber keine weiteren Informationen finden. Daher ist davon auszugehen, dass die Aussagen der Entwickler entweder nicht die Kernentwicklung von Drupal betreffen, oder die Auswertung der Commits den Arbeitsablauf von Drupal nicht exakt widerspiegelt. So könnten z.B. einige Entwickler Patches geschrieben haben, welche dann aber von Dries oder Webchick eingespielt wurden. Da meine Email positiv aufgenommen und auch verstanden wurde, sehe ich im Moment keine Verbesserung der Kontaktaufnahme über Mailinglisten. Um die Erfolgschancen zu erhöhen, Saros in ein Projekt integrieren zu können, habe ich bei Drupal mehr Angebote gemacht als z.B. bei TYPO3. Das heißt, ich habe den Entwicklern eine Einführung in Saros vorgeschlagen. Es wurde aber nicht auf mein Angebot eingegangen.

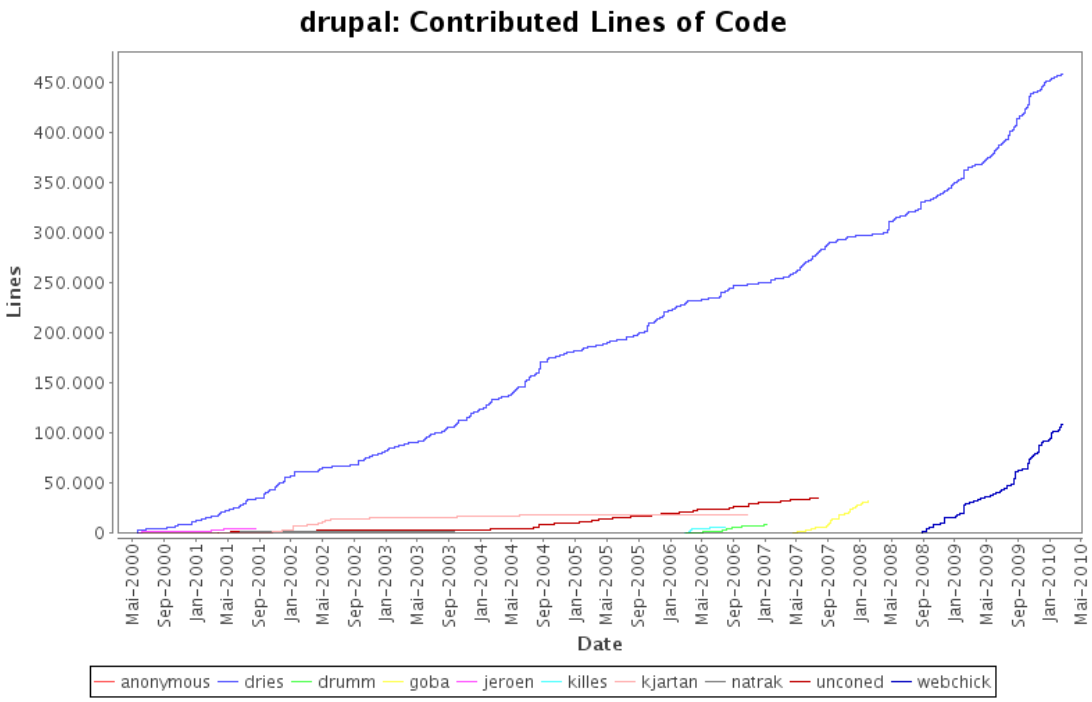

Abbildung 5.16: Codezeilen pro Drupal Entwicker seit Mai 2000

5.4.5.3 Weiteres Vorgehen

Wie schon bei TYPO3 festgestellt, ist die Nutzung von Eclipse im PHP-Bereich keine Selbstverständlichkeit. Es scheint bei Drupal eine Abneigung gegen große Entwicklungsumgebungen zu geben, daher ist es sinnvoll in Zukunft Projekte zu kontaktieren, welche garantiert Eclipse benutzen.

5.5 WordPress

5.5.1 Das Projekt

WordPress ist ein System zur Veröffentlichung und Verwaltung von Inhalten auf einer Webseite. Es eignet sich besonders zum Aufbau eines Blogs[1] und wird daher

1 http://de.wikipedia.org/wiki/Blog

eher als Blogingsoftware bezeichnet und nicht als CMS, wobei WordPress aber durch Erweiterungen und Plugins auch zum Aufbau von komplizierteren Webseiten benutzt werden kann. WordPress ist wie Drupal und TYPO3 in PHP geschrieben und als GPL lizenziert.

5.5.2 Geschichte

Michel Valdrighi entwickelte 2001 unter dem Namen b2/cafelog[1] den Vorgänger von WordPress, welcher ebenfalls unter der GPL verfügbar ist. Nachdem Michel Valdrighi die Weiterentwicklung von b2 einstellte, gab Matt Mullenweg im Januar 2003 in seinem Blog bekannt, dass er auf der b2-Codebasis eine neue Webblog-Software schreiben wolle.[MAT09]. Mike Little schloss sich diesem Projekt an und es entstand am 03.01.2004 die erste stabile WordPress Version. Heute ist WordPress die am weitesten verbreitete Blogingsoftware mit über 8,1 Millionen Downloads der aktuellen Version 2.8[WOR09b].

5.5.3 Das Team

Das WordPress Entwicklungsteam besteht aus zwei Teams:

1. Hauptentwickler:
   Ryan Bore ,Mark Jaquith, Matt Mullenweg, Andrew Oz, Peter Westwood

2. Entwickler:
   Michael Adams, Nikolay Bachiyski, Donncha O Caoimh, Austin Matzko, Joseph Scott, Andy Skelton

Welche Unterschiede zwischen Hauptentwickler und normalen Entwickler bestehen, wird von WordPress nicht näher erläutert.

1 http://cafelog.com/

5.5.4 Projektauswahl

5.5.4.1 Begründung

WordPress wurde zur gleichen Zeit und aus ähnlichen Gründen wie Drupal ausgewählt (5.4.4.1). Das Projekt ähnelt TYPO3 und Drupal sowohl technisch als auch organisatorisch, da es gemeindebasiert ist und von einer einzelnen Person initiiert wurde. Mit elf Entwicklern hat WordPress eine ausreichende Größe, so dass Überschneidungen zwischen den Arbeitsabläufen der einzelnen Entwickler gegeben sein können und somit eine Anwendung von Saros Erfolg versprechender sein könnte.

5.5.4.2 Kontaktaufnahme

Da die Hauptkommunikation des Projektes über Mailinglisten erfolgte und um eine Vergleichbarkeit mit TYPO3 und Drupal herstellen zu können, entschied ich mich WordPress ebenfalls über eine Mailingliste zu kontaktieren. WordPress selbst hat elf Mailinglisten mit unterschiedlichen Themen. Für eine Kontaktaufnahme kam aber nur eine in Frage, die wp-hackers-Liste, da nur hier über die eigentliche Entwicklung von WordPress diskutiert wird [WOR09a]. Über die Mailingliste wurden in der Zeit der erste Iteration 367 Emails verschickt, das entspricht 20,4 Emails pro Tag. Die Liste ist damit ähnlich aktiv, wie die typo3-dev Liste (5.1.4.2).

5.5.5 Erste Iteration

Die erste Iteration nach dem Zeitraum vom 14.10.2009 bis zum 01.11.2009 in Anspruch.

5.5.5.1 Resultate

Abbildung 5.17: Emailverkehr auf der WordPress Mailingiste: WP1, WP2

Wie in der Abbildung **??** zu sehen ist, gab es nur zwei Reaktionen auf meine Email, die erste kam nach zwei Stunden. Die Email von WP1 war eine der längsten Emails, die ich bis dahin bekommen hatte. WP1 äußerte eine starke Abneigung gegen Paarprogrammierung:

> „I've had to do this sort of thing before (not distributed, but pair programming). It is, by far, the worst idea ever conceived by man(agers). I would rip my own arm off and beat myself to death with it before I'd ever do this sort of thing again.“ [A.1.18]

Er war der Auffassung, dass Paarprogrammieren nur dann Sinn macht, wenn beide Entwickler erfahren und auf dem gleichen Niveau sind oder die Entwickler noch keine Erfahrung haben und gerade von der Universität kommen. In allen anderen Fällen wird seiner Erfahrung nach der „bessere“ Programmierer von dem „schlechteren“ auf dessen Niveau herabgezogen. Ich versuchte daraufhin zu argumentieren, dass Saros auch andere Anwendungsmöglichkeiten hat, als nur zum Programmieren benutzt zu werden und zählte folgende Vorteile und Anwendungsmöglichkeiten auf.

- Wissenstranfer,
- Neulinge einarbeiten,
- Minimierung von Defekten im Quelltext,
- Durchsicht von Security Patches

Weiterhin stellte ich die Frage, ob es unter den WordPress-Entwickler überhaupt diese von ihm angesprochenen Niveau-Unterschiede gäbe (A.1.22). WP2 antwortete auf meine Email und unterstützte WP1, er merkte noch an, dass sich Paarprogrammierung zum Wissenstransfer und zum Einarbeiten von Neulingen eignet, es aber bessere Methoden dafür gebe. Die Aussagen von WP1 und WP2 lassen sich also wie folgt kategorisieren:

1. Ablehnung

a) Konzeptionelle Ablehnung

i. Keine sinnvolle Arbeitsmethode oder Prozess

- Abneigungen gegen Pair Programming:

> „I've had to do this sort of thing before (not distributed, but pair programming). It is, by far, the worst idea ever conceived by man(agers). [...] The reality of pair programming is that one guy is better than the other guy, and spends half his time explaining the problem and solution to the other guy, instead of simply coding the damn thing. The lesser of the two programmers only drags the other guy down to his level, and wastes both of their time.“ [WP1 A.1.18]
>
> „However, if the intent is to produce clean, bug-free code in a fast and efficient manner, what Otto pointed out is true in many/most situations: you're going to handcuff a gifted programmer and produce worse code, slower.“ [WP2 A.1.23]

Von WP1 kam keine weitere Reaktion.

5.5.5.2 Fazit der ersten Iteration

Es gab zum Teil starke Abneigungen gegen Paarprogrammierung, begründet durch schlechte Erfahrungen, die damit gemacht worden sind. Da diese Abneigungen schwer zu überwinden sind und auf Saros projiziert wurden, wurde Saros nicht getestet. Hieraus wurde der Schluss gezogen, dass Saros von PP distanziert werden sollte und folglich die Benutzung von Saros nicht als PP oder DPP dargestellt wird. Zwar wird mit dem Begriff DPP schon von PP distanziert, ich vermutete aber, da das Wort Pair Programing in DPP vorkommt, viele Entwickler DPP mit PP gleichsetzten. Eine Idee war z.B. von *kollaborativer Zusammenarbeit mit Saros* zu sprechen.

5.5.5.2.1 Weiteres Vorgehen

Da auf der Mailingliste nur zwei Entwickler geantwortet haben und diese Reaktion sehr negativ und abweisend waren, sehe ich von einer zweiten Iteration ab.

5.6 JUnit

5.6.1 Das Projekt

JUnit ist ein Testframework für in Java geschriebene Programme, welches ermöglicht automatische Tests einzelner Methoden oder Klassen, sogenannte Unit-Tests, durchzuführen. Das Framework ist selber in Java geschrieben und wurde von Kent Beck und Erich Gamma entwickelt. JUnit wird gerne als Hilfsmittel innerhalb agiler Software Entwicklung[1] angewendet, z.B. bei Testgetriebener Entwicklung vgl. [Bec02].

5.6.2 Das Team

Das JUnit Team besteht aus vier Projekt-Administatoren: Erik G. H. Meade, Erich Gamma, David Staff und Kent Beck, wobei die drei letztgenannten Hauptentwickler sind.

5.6.3 Projektauswahl

5.6.3.1 Begründung

Bei diesem Projekt gab es mehrere Indizien, dass Eclipse zur Entwicklung eingesetzt wird. Zum einen leitete Erich Gamma die Entwicklung der JDT[2] für Eclipse. Ab der Eclipse-Version 3.2 wurde JUnit in Eclipse integriert. Zum anderen wird auf der Mailingliste öfter über Eclipse diskutiert und es ist eine .project-Datei im Repository vorhanden.

1 http://agilemanifesto.org/
2 Java development Tools

Eine weitere Möglichkeit, warum Saros für dieses Projekt interessant sein könnte ist, dass Kent Beck Verfechter der agilen Softwareentwicklung ist [AGI09] und Saros ein Hilfsmittel dieser Methoden sein kann, insbesondere für DPP.

5.6.3.2 Kontaktaufnahme

Es gab zwei Möglichkeiten dieses Projekt zu kontaktieren. Entweder über die Entwickler-Mailingliste oder die Hauptentwickler über das Sourceforge-Kontaktformular direkt anzuschreiben. Ich entschied mich die Mailingliste zu nutzen, da ich bis jetzt negative Erfahrungen mit der zweiten Methode gesammelt hatte. Da es viele Gründe gab, weshalb Saros in diesem Projekt genutzt werden könnte, wollte ich hier keinen neuen Versuch starten ein Projekt über die Maintainer zu kontaktieren, um eine möglichst hohe Erfolgschance zu haben.
Bei genauerer Betrachtung der Mailingliste stellte ich fest, dass diese nicht häufig benutzt wird. Im Jahre 2009 wurden gerade mal 60 Emails über die Liste geschickt, insgesamt, von 2002 bis 2009, waren es nur 1348 Emails, das sind 12 Emails pro Monat[SOU10m]. Da aber keine andere Kommunikationsmöglichkeit betreffend der Entwicklung zu finden bzw. angegeben war, blieb ich bei der Entscheidung das Projekt über diese Liste zu kontaktieren. (A.1.25)

5.6.4 Resultate

Abbildung 5.18: Emailverkehr auf der JUnit Mailingliste: DR1

Ich erhielt insgesamt nur zwei Antworten, DR1 zeigte sich interessiert an Saros und wollte wissen ob das Plugin einfach so zu installieren sei oder ob er vorher etwas beachten müsse. (A.1.25) Ich erklärte ihm kurz, wie er Saros installieren kann, bzw. dass er ein Jabber- oder Google-Mail- Benutzerkonto benötige, um Saros benutzen zu können und verwies auf die Saros-Webseite. Die zweite Antwort war eine Benachrichtigung von JU2, dass er meine Email an jemanden weitergeleitet hat, der sich potentiell für Saros interessieren könnte (A.1.25).

1. Akzeptanz
   a) Technische Akzeptanz
      i. Sinnvolles Produkt
         - Interesse/sinnvoller Vorschlag

           „That definitely sounds interesting.“ [JU1 A.1.25]
2. Unklarheiten
   a) Technische Nachfragen
      i. Installation

         „Is the plug-in ready to go, or is there something I should know first?“ [JU1 A.1.25]
   b) Konzeptionelle Nachfragen
      i. Effektivität

5.6.5 Fazit

Trotzt der Annahme, dass sich dieses Projekt gut für eine Kontaktaufnahme eignet, erhielt ich sehr wenig Resonanz von diesem Projekt. Es stellte sich also die Frage, warum nur zwei Entwickler geantwortet hatten. Ich vermute, dass die geringen Antworten nicht mit Saros als Produkt zusammenhängen, sondern vielmehr mit der generell Nutzung der Mailingliste. Da aber keine andere öffentliche Kommunikationsmöglichkeit angegeben war, ist zu vermuten, dass Diskussionen, die Entwicklung von JUnit betreffend, entweder nicht öffentlich geführt werden oder an dem Projekt nicht mehr aktiv gearbeitet wird. Eine Untersuchung des Repositorys ergab allerdings, dass an diesem Projekt sehr wohl noch gearbeitet wird[GIT10]. Ich gehe daher davon aus, dass auf anderen, nicht zugänglichen Kanälen kommuniziert wird. Die Hauptentwickler noch einmal gesondert anzuschreiben erschien mir nicht sinnvoll, da sie höchstwahrscheinlich Mitglieder der Liste sind und mein Anliegen schon bekommen haben.
Es bleibt die Vermutung, dass Projekte mit geringer öffentlicher Kommunikation

(12 Emails pro Monat) für meine Anfragen weniger geeignet sind. Ich sah daher in diesem Fall von einer weiteren Iteration ab.

5.7 Eclipse-cs

5.7.1 Das Projekt

Eclipse-cs ist ein Plugin, welches das Quellcode-Analyse-Progamm Checkstyle[1] in Eclipse integriert. Die Software hilft dabei, dass vorher definierte Programmierrichtlinien eingehalten werden; sollte es zu Verletzungen dieser kommen, wird der Entwickler benachrichtigt. Das Projekt wurde von David Scheider gegründet und ist unter der GNU Lesser General Public License 2.1 verfügbar [OHL10a].

5.7.2 Das Team

Das Projekt besteht aus fünf Entwicklern: David Schneider, Fabrice Bellingard, Levon Saldamli, Nicolas De loof und Lars Ködderitzsch, wobei letzterer der momentane Entwicklungsleiter ist [SOU10q].

5.7.3 Projektauswahl

5.7.3.1 Begründung

Wie schon in vorherigen Kapiteln (vgl. z.B. 5.4.5.1.) zu sehen ist, benutzen einige Entwickler Eclipse nicht zur Entwicklung. So ist es sinnvoll ein Projekt auszuwählen, welches auf jeden Fall mit Eclipse arbeitet. Da es sich bei eclipse-cs um ein Eclipse-Plugin handelt ist dies bei diesem Projekt gegeben; alle Entwickler arbeiten mit Eclipse [SOU10d]. An dem Projekt wird aktiv gearbeitet und es erscheinen regelmäßig neue Versionen [OHL10b].

1 http://checkstyle.sourceforge.net/

5.7.3.2 Kontaktaufnahme

Das Projekt besaß keine öffentliche, die Entwicklung betreffend Mailingliste, es war nur ein Forum für generelle Diskussionen vorhanden. Ich entschied mich daher die angegebenen Entwickler direkt zu kontaktieren. Die Kontaktaufnahme fand über die Sourceforge-Kontaktformulare statt (A.1.25).

5.7.4 Resultate

Trotz wiederholter Nachfrage erhielt ich von den Entwicklern keine Antwort.

5.7.5 Fazit

Eclipse-cs war damit das zweite Projekt, das ich auf diese Art kontaktiert hatte und von dem ich keine Antwort erhielt; das erste Projekt war Vuze (vgl. 5.2.4.2). Die Projekte haben nicht viele Gemeinsamkeiten, Vuze wird im Gegensatz zu eclipse-cs von einer Firma geleitet und betreibt keinen öffentlichen Entwicklungsprozess. Zwar bietet eclipse-cs auch keine Möglichkeit, von außen an Entwicklungsdiskussionen teilzunehmen, aber im Gegensatz zu Vuze werden mehrere öffentlich zugängliche Tracker betrieben, um den Benutzern die Möglichkeit zu geben, an dem Projekt teilzunehmen. [SOU10e]
Ich stelle daher die Vermutung auf, dass diese Kontaktart und Projekte, die nicht über die Möglichkeit verfügen an die Entwicklung betreffenden Diskussionen teilzunehmen, sich weniger für eine Kontaktaufnahme eignen. Dies gilt es bei weiteren Projekten zu untersuchen.

5.8 ADempiere

5.8.1 Das Projekt

ADempiere ist eine Software mit deren Hilfe Unternehmen ihre Ressourcen (z.B. Kapital oder Personal) planen und verteilen können, dies wird als ERP[1]-System bezeichnet. Die Entwicklergemeinde folgt dem *Bazaar*-Leitbild, das vom Autor und Programmierer Eric S. Raymond geprägt wurde, um die Bedürfnisse und Beiträge der Gemeinschaft besser beachten zu können, vgl. Abschnitt 2.2.1.

5.8.2 Geschichte

ADempiere spaltete sich 2006 von Compiere[2] ab, da Compiere teilweise auf dem Weg war Closed Source zu werden. ADempiere übernahm die Grundmodule vom Compiere und wird seitdem von einer Gemeinde weiterentwickelt. [RED10] Im Jahre 2008 gründete sich *ADempire Deutschland e.V.* mit dem Ziel:

> „die Weiterentwicklung von ADempiere im Hinblick auf die Lokalisierung, Öffentlichkeitsarbeit und die Sicherstellung der freien Rechte des Systems“ [ADE10c]

zu gewährleisten.

5.8.3 Das Team

Das Team besteht aus insgesamt 102 aktiven Mitglieder und wird von zehn Projekt-Administratoren betreut: Colin Rooney, Carlos Ruiz, goanookie, Heng Sinnwessel, Red1 D. Oon, Ramiro Vergara, Teo Sarca, Trifon Trifonov und Victor Perez Juarez. Von den 102 Mitgliedern sind 66 als Entwickler klassifiziert [SOU10b].

1 Planung der Unternehmensressourcen, engl. Enterprise Resource Planning
2 http://www.compiere.com/

5.8.4 Projektauswahl

5.8.4.1 Begründung

ADempiere ist ein Projekt, das sich meiner Vermutung nach von allen bisher angeschriebenen Projekten am besten für eine Kontaktaufnahme sowie für den Einsatz von Saros eignet. Die Programmiersprache des Projektes ist Java und es wird mit Eclipse gearbeitet, da eine .project-Datei im Repository liegt. Zudem ist es gemeindebasiert und folgt dem Basar-Prinzip, wodurch es gegenüber äußeren Anregungen aufgeschlossen und eine öffentliche Entwicklung praktizieren sollte. Tatsächlich ist der Entwicklungsprozess dieses Projekte sehr offen. Es gibt öffentliche Dokumentationen über viele Prozesse, die in dem Projekt stattfinden und darüber, wie Entscheidungen getroffen werden.

Ein besonderes Augenmerk legt ADempiere auf die Qualitatssicherung. Diese findet hauptsächlich mit Hilfe von 14 Tracker statt, zu denen unter anderem Tracker für Offene Ausgaben, Bugs, Features und Supportanfragen gehören. Dabei ist jeder angehalten sich an dem Projekt, z.B. über einen der Tracker, zu beteiligen. Neue Benutzer können beispielsweise die Software installieren und testen, um dann über Defekte oder Unklarheiten zu berichten oder ihr spezifisches Domänenwissen in das Wiki einfließen zu lassen. Es werden weitere Vorschläge für erfahrene Benutzer, Entwickler und Committer gemacht, wie man sich beteiligen kann, vgl. [ADE10b]. Ein wichtiger Aspekt der Qualitätssicherung und eventuelle Anwendungsmöglichkeit von Saros ist der *Bug Day* bzw. *Bug Triage*, auf den im nächsten Abschnitt genauer eingegangen wird.

5.8.4.2 Kontaktaufnahme

Zunächst untersuchte ich die Mailinglisten. Das Projekt hatte fünf Mailinglisten, wobei über zwei nur CVS-Logs und Tracker-Logs geschickt wurden. Das heißt, über diese Mailinglisten wurden automatisch die Änderungen des jeweiligen Systems geschickt. Die übrigen Mailinglisten hatten so wenig Verkehr, dass diese nicht als Kommunikationswege dienen konnten (insgesamt 307 Nachrichten seit 2007).

[SOU10a] Das Projekt betriebt ebenfalls 39 Foren mit durchschnittlich ca. 1.117 Einträgen pro Monat, das macht ca. 36 Einträge pro Tag. Da die Aktivität in den Foren sehr viel höher war, als auf der Mailingliste, lag es nahe, dass diese auch zur Entwicklung benutzt werden. [SOU10j] Zwei Foren fielen bei genauerer Analyse besonders auf: *Community Process* und *Developers*. Leider gab es keine nähere Beschreibung, was für eine Art von Themen für welches Forum bestimmt ist, wie das üblicherweise bei Mailinglisten der Fall ist. So musste ich mir anhand des Archives selbst ein Bild machen. Es ergab sich, das beide thematisch Foren sich für eine Kontaktaufnahme eignen würden.
Da das Projekt eine sehr offene Struktur und Prozessbeschreibung anbietet, entschied ich mich nach direkten Anwendungsmöglichkeiten für Saros zu suchen, um diese bei der Kontaktaufnahme den Entwicklern vorzuschlagen.
Der *Bug Day* schien mir eine gute Anwendungsmöglichkeit zu sein. Jeden vorletzten Donnerstag im Monat treffen sich alle Entwickler im Internet, um Bugs aus dem Tracker gemeinsam zu reparieren. Dabei gibt es drei verschiedene Treffen für folgende Kontinente:

1. Asien: 02:00 h GMT
2. Europa: 09:00 h GMT
3. Amerika: 15:00 h GMT

Für jedes dieser Treffen gibt es einen Mentor, der den Bug Day koordiniert und, wie alle anderen Entwickler, den ganzen Tag über im IRC zu erreichen ist. Der Bug Day hat jedoch nach Aussage des Projekt-Wikis nicht nur die Funktion vorhandene Defekte zu reparieren, sondern soll das Teamwork steigern und erfahrene Entwickler mit Neueinsteigern zusammen bringen [ADE10a].

Es boten sich also zwei Kontaktmöglichkeiten an, einmal einen Eintrag im Forum mit dem Vorschlag Saros während des Bug Day einzusetzen, oder die beiden Betreuer direkt anzuschreiben. Ich entschied mich, den kommenden Bug Day zu fokussieren und zu versuchen, Saros in diesen Prozess zu integrieren. Da das Projekt dem Basar-Leitbild folgte, hielt ich es für sinnvoller, alle Entwickler mit einzubeziehen und kontaktierte das Projekt also über das Forum (A.1.27).

5.8.5 Erste Iteration

5.8.5.1 Resultate

Die erste Antwort auf meinen Eintrag kam 17 Minuten später von AD1. Der Entwickler war gerade online, woraufhin ich mich in den IRC-Kanal *#adempiere* einklinkte. AD1 war dort ebenfalls eingeloggt und begrüßte mich auch gleich. Er war gerade dabei, die Saros-Webseite und das Video zu begutachten. AD1 stellte folgende Fragen zu Saros, welche ich ihm mit der Hilfe von Sandor Szücs, einem Entwickler aus dem Saros Team, der ebenfalls im Kanal eingeloggt war, beantwortete:

1. Auf was setzt Saros auf, oder ist alles selbst geschrieben? (A.1.27)
2. Wurde Saros schon einmal in „echten“ Projekten eingesetzt und welche Erfahrungen wurden da gemacht? (A.1.27)
3. Was ist der langfristige Plan für Saros? (A.1.27)
4. Wie viele Entwickler können Saros gleichzeitig nutzen. (A.1.27)
5. Was ist unsere Motivation hinter Saros? (A.1.27)
6. Wie ist diese Idee zum ersten mal entstanden? Wer hat damit angefangen? (A.1.27)
7. Wieviel Bandbreite braucht Saros? (A.1.27)
8. Wie viele Entwickler arbeiten an Saros? (A.1.27)
9. Gibt es Erkenntnisse darüber ob die Produktivität durch Saros steigt? (A.1.27)
10. Gibt es in Saros einen Modus 'nur lesen', so dass der Eingeladene den Quelltext nicht verändern kann? (A.1.27)
11. Funktioniert Saros auch auf einem Mac? (A.1.27)

Der Grund für seiner vielen Fragen war, dass er die thailändische Regierung von OSS überzeugen möchte und dafür Beispiel-Projekte sucht. Dass Saros an einer Universität entwickelt wird, schien ihn ganz besonders zu interessieren. Allerdings ging er auf diesen Punkt nicht näher ein. (A.1.27)

AD1 war sehr begeistert von Saros und konnte sich den Einsatz in OSS Projekten durchaus vorstellen:

> „clearly Saros looks revolutionary“ [A.1.27]
> „i can imagine a whole OSS team working with multiple java artifacts at the same time“ [A.1.27]

Zu diesem Zeitpunkt hatte er Saros noch nicht getestet, sondern sich seine Meinung anhand der Sarosbeschreibung auf der Webseite gebildet. AD1 sah folgende Anwendungsmöglichkeiten bzw. Vorteile von Saros:

- Zusammenarbeit von mehren Entwicklern an einem bestimmten Artefakt,
- Präsentation und Lernwerkzeug (A.1.27),
- Wissensverteilung (A.1.27),
- Peer-Review druchführen (A.1.27),
- Schwierige Probleme zusammen lösen (A.1.27),
- Bessere Einsicht in die Arbeit anderer (A.1.27),
- Neue Erfahrungen können gesammelt werden (A.1.27)

Nach einer ca. einstündigen Stunde Diskussion schlug ich AD1 vor, mit mir eine Sitzung aufzubauen. Zum gleichen Zeitpunkt meldete sich AD2. Er wunderte sich darüber, dass seine Benutzerliste in Saros leer war, als er ein Projekt freigeben wollte und fragte, wie er dort Benutzer hinzufügen könne. Ferner wollte er wissen, wie die Freigabe beendet wird. Er hatte während der Diskussion mit AD1 Saros installiert. (A.1.27)

AD1 hatte Probleme Saros zu installieren. Saros erschien nach der Installation

zwar in dem Plugin-Ordner von Eclipse, wurde aber innerhalb von Eclipse nicht angezeigt. Dies Problem konnte bis zum Ende der 1.Iteration nicht behoben werden. (A.1.27)
Da AD2 Saros schon installiert hatte versuchte ich zunächst mit AD2 eine Sitzung aufzubauen, während dessen gab Sandor AD1 weiterhin Hilfestellung bei der Saros-Installation. Innerhalb von drei Minuten hatten AD2 und ich uns gegenseitig als Freunde im Roster hinzugefügt(A.1.27).
Ich versuchte AD2 in ein Test-Projekt einzuladen, was nicht funktionierte, da sich AD2 zu diesem Zeitpunkt schon in einer Sitzung befand, welche er selbst erstellt hatte. Ich wies ihn darauf hin, worauf er versuchte, mich einzuladen. Der Einladungsprozess dauerte sehr lange. Nach drei Minuten stellte sich heraus, dass AD2 versuchte das Projekt ADempiere mit mir zu teilen, was bedeutete das er mir ca. zwei Gigabyte Daten hätte schicken müssen. Ich erklärte ihm, dass es zu lange dauern würde diese Menge an Daten über Saros zu synchronisieren und beendete daraufhin die Einladung. Er schlug vor, ein kleineres Projekt zu benutzen. (A.1.27, A.1.27)
Als er dieses mit mir teilen wollte funktionierte dies nicht, da er noch in der alten Sitzung war, er verstand aber auf Grund der Fehlermeldung das Problem und beendete die alte Sitzung. Ich erschien in seinem Einladungsdialog, aber bei der Einladung erhielt AD2 immer die Fehlermeldung IBB[Error] im Roster. AD2 vermutete, dass die Verbindung unterbrochen wurde. Ich untersuchte daraufhin meine Logdateinen und stellte fest, das dies nicht der Fall war und ich immer noch Daten von ihm bekam. Dies teilte ich ihm auch mit und zwei Minuten später hatten wir die erste Sitzung aufgebaut. (A.1.27)
Nach dem wir in der aktuell offenen Datei etwas Pseudo-Quelltext geschrieben hatten und dies auch problemlos funktionierte, schlug ich vor, den *Follow-Mode*, vgl. Abschnitt 2.3.3, auszuprobieren um AD2 die Funktionalitäten von Saros näher zu bringen. Dafür musste ich ihm erst erklären, wie er mir die Driver-Rolle übergeben kann. Nachdem das geschehen war, zeigte ich ihm die verschiedenen Möglichkeiten von Saros, was ihn sehr begeisterte.

„ok - work pretty snappy too

a great teaching tool! [...]
very good guys ... well done“ [A.1.27]

AD2 handelte eher autodidaktisch. Anders als bei AD1 hatte ich bei AD2 den Eindruck, er habe Saros einfach installiert ohne sich die Dokumentation durchzulesen. Er war die ganze Zeit Host und fand sich nicht sofort in Saros zurecht, ich konnte ihn auch nicht führen, da ich keine Driverrechte hatte und wir so immer ins IRC wechseln mussten um uns zu verständigen. Nach kurzen Erklärungen war ihm dann aber das Prinzip von Driver/Observer und die grundsätzliche Funktionsweise schnell klar.
Nachdem AD2 sich einen Überblick von Saros verschafft hatte, überlegten wir gemeinsam, wie mit Saros an dem zwei Gigabyte großen ADempiere Projekt gearbeitet werden kann. AD2 sah nur zwei mögliche Vorgehensweisen: entweder gleichzeitig mehrere kleine Projekte zu teilen oder das gesamte Projekt an sich zu belassen. Da Saros keine multiplen Projekte unterstützt, blieb nur die zweite Möglichkeit. Ich schlug vor, mir erstmal den aktuellen Trunk[1] von ADempiere herunterzuladen, damit nicht so viele Daten synchronisiert werden müssen. (A.1.27)
AD3 wollte wissen, ob es nicht die Möglichkeit gibt, das Synchronisieren auf bestimmte Dateien zu beschränken, z.B. nur .java und .xml-Dateien. Da so nur ein Bruchteil von ADempiere synchronisiert werden muss. Dies war mit dem noch experimentellen *Partially Sharing*, vgl. Abschnitt 2.3.3, von Saros möglich, daher schlug ich vor, dieses mit dem aktuellen Trunk zu benutzen. AD2 eröffnete wieder die Sitzung und merkte an, dass es sinnvoll wäre, wenn Saros sich die gemachten Einstellungen im *Partially Sharing* speichern würde, so das die Einstellungen nicht jedes mal neu gemacht werden müssen. (A.1.27)
Als AD2 seine Einstellungen abgeschlossen hatte und mich einlud wählte ich ein neues Projekt, da die Daten jetzt ca. 150 Megabyte betrugen. Jedoch mussten wir die Einladung abbrechen, da ich nach einiger Zeit keine Daten mehr empfing. Dieses Problem wurde später analysiert, im Fazit wird darauf näher eingegangen werden. (A.1.27)

1 Der Hauptentwicklungszweig einer Versionsverwaltung wird als Trunk bezeichnet

AD2 lud mich noch einmal ein und diesmal wählte ich ein existierendes Projekt, nämlich den aktuellen Trunk von ADempiere, um die Synchronisation möglichst klein zu halten. Dabei kam es zu folgendem Problem: Dadurch, dass AD2 mir nur bestimmte Daten und Ordner anbot, wurden durch die Synchronisation auf meiner Seite alle anderen Dateien und Ordner des Projektes ohne Vorwarnung gelöscht. Dies sahen wir beide als Problem an.
Das Parially Sharing war also so nicht zu verwenden. Daher versuchten wir das gesamte Projekt zu teilen, in dem wir beide den aktuellen Trunk benutzten. Hierbei trat wieder das oben schon genannte Problem auf, das AD2 mir ab einem gewissen Zeitpunkt keine Daten mehr schickte und wir die Einladung abbrechen mussten. Ebenfalls traten wieder irritierende Fehlermeldungen im Roster auf.
In der darauffolgenden Diskussion stellten wir weiterhin fest, das dies kein sinnvolles Vorgehen ist, zwar ist nach dem Herunterladen des Trunks das Projekt auf beiden Seiten synchron, aber nach dem Kompilieren von ADempiere werden große Binärdaten erstellt, die sich so von Entwickler zu Entwickler durch Zeitstempel unterscheiden. Da Saros aber über die Hashwerte der Dateien die Synchronisation herstellt, müssten diese Dateien immer nach jedem Kompilieren synchronisiert werden. (A.1.27) Um Saros mit ADempiere benutzen zu können, musste es also möglich sein die Dateien, welche verteilt werden sollen, zu filtern.
Um 19:56 Uhr veröffentlichte AD2 im Forum eine Zusammenfassung unserer Erfahrungen im Bezug auf Saros und ADempiere. Er beschreibt, wie wir vorgegangen sind und was dabei heraus gekommen ist. Ferner merkte er an, dass die Dokumentation bzw. Hilfe von Saros gut zu verstehen ist. (A.1.27)
AD1 antwortete um 23:37 Uhr darauf, das er die Saros Installation noch einmal versucht hatte und es diesmal ohne Probleme funktionierte. Er habe jetzt aber das Problem, das er sich nicht zu einem Jabber-Server verbinden kann.

> „However on pushing the connect icon in Roster, it connects to this url bernas.com.my which is known to be a local company doing rice mills for the government (lol, but since when did i plunder the folks rice?) and supply me as an error.“ [A.1.27]

Ich bedankte mich darauf bei AD2 für seine umfangreiche Zusammenfassung und wies darauf hin, das ich am 24., 25. und 26. November im ADempiere zu erreichen sein werde, falls noch jemand Saros vor dem Bug Day ausprobieren möchte. Jedoch erhielt ich bis zu dem Bug Day keine weiteren Anfragen.

Eine Zusammenfassung und Interpretation der Forumsbeiträge und des Chatmitschnitts ergibt folgendes Bild.

1. Akzeptanz
   a) Technische Akzeptanz
      i. Sinnvolles Produkt
         - Interesse/sinnvoller Vorschlag:

           „clearly Saros looks revolutionary“ [AD1 A.1.27]
         - Anwendungsvorschlag:

           „but if lets say 5 can work on it and the rest can just view it, it can be a great teaching tool [...] and process of distributing knowlege increases“ [AD1 A.1.27]
         - Vorschläge für Weiterentwicklung:

           „if it can be stored for later replay.. awesome possibilities“ [AD1 A.1.27]
           „can the synch be reduced to .java files? [...] and xml files?“ [AD3 A.1.27]
           „it would be nice if you could store & retrieve the partial project selected so that each time you would not hav eto manually unselect bunches of the project“ [AD2 A.1.27]
         - Gute Lösung (Saros getestet):

           „ok - work pretty snappy too a great teaching tool! never mind pair programming! very good guys ... well done“ [AD2 A.1.27]
   b) Konzeptionelle Akzeptanz

i. Sinnvolle Arbeitsmethode oder Prozess

- Sonstiges:

  „the idea seems great.“ [AD3 A.1.27]

c) Umfeld bezogene Akzeptanz

i. Produkt passend für das Projekt

- Anwendungsvorschlag für das Projekt:

  „we do have a serious peer review constraint our trunk has to be well guarded and yet we want faster evolution with some Saros, i can imagine more closer devs insight into how each other work can help the goodwill“ [AD1 A.1.27]

2. Unklarheiten

a) Technische Nachfragen

i. Performance

- Bandbreite:

  „hows the bandwidth requirement?“ [AD1 A.1.27]

ii. Kompatibilität

- Betriebssystem:

  „so it can work on Mac?“ [AD1 A.1.27]

iii. verwendete Technologien:

„is it done from scratch or built upon an earlier idea?“ [AD1 A.1.27]

iv. Sonstige:

„what can be the max users?“ [AD1 A.1.27]
„can there be a read only mode spawnyd ?“ [AD1 A.1.27]
„reading about Mumble part hmm it has voice?“ [AD1 A.1.27]
„when you share a project ... you get a user list ... where

does that list come from? because mine is empty“ [AD2 A.1.27]
„also once you share a project how do stop sharing?“ [AD2 A.1.27]
„so must you have the same project open as me to share?“ [AD2 A.1.27]
„does it look at timestamps or revision numbers or something like that?“ [AD2 A.1.27]

b) Konzeptionelle Nachfragen

i. Effektivität

„did u try calculating the impact of using such tool? i mean positive rate of productivity?“ [AD1 A.1.27]

ii. Sonstige:

„wonder if this is good for ’pair programming’ ?“ [AD1 A.1.27]
„what are you guys plans for this? will it remain in the unviersity?“ [AD1 A.1.27]
„what are you other motivation for such an addon?“ [AD1 A.1.27]
„how did that idea first appear?“ [AD1 A.1.27]
„so how many devs u have working on this?“ [AD1 A.1.27]

5.8.5.2 Fazit der ersten Iteration

Bei der Kontaktaufnahme war besonders hilfreich, dass ich mich, nachdem ich eine Antwort von AD1 erhalten hatte, sofort in den Chat-Kanal von ADempiere eingeloggt habe. Dadurch entwickelte sich eine Diskussion wie sie über das Forum warscheinlich nicht entstanden wäre. So war es möglich, Fragen unmittelbar zu beantworten und Hilfe bei der Installation zu geben. Zudem konnte ich mit AD2 eine Saros-Sitzung aufbauen, was ebenfalls daran lag, das ich vorher mit ihm per IRC gechattet hatte und wir uns damit besser koordinieren konnten. Ein weiterer

Vorteil ist, das man so unmittelbares Feedback bekommt, das wahrscheinlich, wenn die Kommunikation per Mail oder Forum stattgefunden hätte, gar nicht mitgeteilt worden wäre. Die Kommunikation per IRC ist eine andere als per Email oder Forum, es wird nicht so intensiv über den zuschreibenden Text nachgedacht. So werden scheinbar belanglose Aussagen kommuniziert, die im Nachhinein aber durchaus interessant und lehrreich sein können. Die Auseinandersetzung ist lebhafter und so auch reicher an Erkenntnissen. Allerdings ist die Auswertung dieser Diskussionen viel schwieriger als per Email oder Forum geführte Diskussionen. Die Gespräche im Chat können nahezu in Echtzeit und mit mehreren Personen ablaufen. Das heißt, es müssen nachher mehrere miteinander verflochtene Diskussionsfäden, die teilweise kontextabhängig sind, strukturiert und zusammengefasst werden. Nicht zu vergessen, das meistens auch noch andere Gespräche in dem Kanal stattfinden, die nichts mit der aktuellen Anfrage zu tun haben.

Durch die Saros-Sitzung mit AD2 wurde mir klar, dass es sinnvoller ist, unerfahrenen Saros-Benutzer selbst einzuladen, um sie mit Saros vertraut zu machen, als sich von ihnen einladen zu lassen. So hätten sich einige, kleinere Probleme vermeiden lassen können. Das Wechseln zwischen Saros und IRC ist ebenfalls hinderlich gewesen. So wäre es zukünftig sinnvoll einen IRC Chat innerhalb von Saros zu haben.
Ein weiteres Problem war, dass kein normales Verteilen des Projektes möglich war und ein partielles Verteilen von Daten möglich sein muss. Die momentane Implementierung weist einige Fehler auf. Zum einen funktioniert sie gar nicht, wenn ein leeres Projekt ausgewählt wird. Zum anderen löscht sie auf der Clientseite ohne Vorwarnung nicht verteilte Dateien.
Die Größe des Projektes stellte aber noch weitere Probleme dar. Ich versuchte ADempiere noch einmal einem lokalen Test zu unterziehen. Sowohl Host also auch Client hatten das gleiche Projekt, den aktuellen Trunk von ADempiere, ohne diesen einmal kompiliert zu haben. Es waren also exakt die gleichen Daten auf beiden Seiten vorhanden. Beim Einladungsprozess stürzte aber das Eclipse des Hosts immer ab. Die resultierende Fehlermeldung *java.lang.OutOfMemoryError: Java heap space* lies vermuten, das dies ein Arbeitsspeicherproblem darstellt.

Im folgenden zähle ich die aus der 1. Iteration resultierten Defekte und Feature-Requests auf:

Defekte:

1. STATUS is irritating/ wrong while inviting[1]
2. IBB[Error] while invitation[2]
   Beschreibt das Problem bei dem AD2 dachte, das die Verbindung abgebrochen wurde, obwohl sie noch nicht vorhanden war.
3. Partial Sharing into an existing Project [3]
   Beschreibt das Problem, dass nichtverteilte Dateien auf der Clientseite ohne Vorwarnung gelöscht werden können.
4. Partial sharing into a new Project[4]
   Beschreibt den Defekt, dass das Partial sharing in ein neues Projekt nicht funktioniert.
5. Sharing a very big Project[5]
   Beschreibt das Problem, das bei großen Projekten das Fileset nicht erstellt werden kann, da der Speicher überläuft.

Feature-Requests:

1. Save Saros sessions for later replay[6]
   Ein von AD1 vorgeschlagenes Feature zum Aufzeichnen von Programmiersitzungen.

1 http://sourceforge.net/tracker/?func=detail&aid=2904494&group_id=167540&atid=843359
2 http://sourceforge.net/tracker/?func=detail&aid=2906728&group_id=167540&atid=843359
3 http://sourceforge.net/tracker/?func=detail&aid=2903242&group_id=167540&atid=843359
4 https://sourceforge.net/tracker/?func=detail&aid=2903240&group_id=167540&atid=843359
5 https://sourceforge.net/tracker/?func=detail&aid=2903243&group_id=167540&atid=843359
6 http://sourceforge.net/tracker/?func=detail&aid=2906730&group_id=167540&atid=843362

2. Whiteboard
   Von AD1 wurde vorgeschlagen, Saros als Whiteborad zu benutzen. (A.1.27) Ich habe dies ebenfalls als Feature-Request aufgenommen, da Saros sich hierfür nur bedingt eignet. Einen Eintrag in den Feature-Tracker habe ich nicht vorgenommen, da hierfür schon eine Bachelorarbeit geplant ist[7].

3. Chat is Missing[1]
   Diese Anfrage gab es schon, wurde aber in dieser Iteration wieder angefragt.

4. Support Multiple Projects in session[2]
   Beschreibt die fehlende Funktionalität, die es ermöglicht mehrere Projekte gleichzeitig zu verteilen. Diese Anfrage war ebenfalls schon vorhanden.

5. Remember selection when sharing a part of a project[3]
   Beschreibt die von AD2 gewünschte Funktionalität, das die Einstellungen im Partially Sharing gespeichert werden.

5.8.5.3 Weiteres Vorgehen

Bis zum anstehenden Bug Day von ADempiere sollte es möglich sein ADempiere über Saros zu verteilen, dafür müssen entweder der Speicherdefekt behoben werden oder das Partially Sharing überarbeitet werden. Wobei zweiteres sinnvoller ist, da so auch der Einladungsprozess beschleunigt wird.

5.8.6 Zweite Iteration

Es wurde ein Workaround[4] erstellt mit dem es möglich war ADempiere über Saros zu verteilen. Es war nur eine Woche zwischen der ersten Kontaktaufnahme und dem

7 https://www.inf.fu-berlin.de/w/SE/ThesisDPPXVII
1 https://sourceforge.net/tracker/?func=detail&aid=2727894&group_id=167540&atid=843362
2 https://sourceforge.net/tracker/?func=detail&aid=2795851&group_id=167540&atid=843362
3 http://sourceforge.net/tracker/?func=detail&aid=2903246&group_id=167540&atid=843362
4 Unter einem Workaround versteht man die Umgehung eines Problems durch eine provisorische Hilfskonstruktion

Bug Day vorhanden, diese Zeit reichte nicht aus die eben genannten Defekte zu beheben. Die Idee des Workarounds war, das Saros als *Derived*[5] markierte Dateien nicht in die Synchronisation mitaufnimmt. Dies ist vor allem, dafür gedacht, dass z.B. Versionsverwaltungs-Dateien und Eclipse interne Dateien bei der Synchronisation nicht mitabgeglichen werden. Es ist möglich Odner und Dateien manuell als Derived zu markieren, diese werden dann ebenfalls nicht mit abgeglichen. Ich stellte die Anleitung zu diesem Workaround im Forum zur Verfügung (A.1.29).

5.8.6.1 Resultate

Nachdem ich mich während des Bug Days in den ADempiere Kanal eingeloggt hatte, erkundigte ich mich, ob der Workaround, welchen ich tags zuvor in das Forum eingetragen hatte, zur Kenntniss genommen worden ist und funktionierte. AD2 antwortete mir, dass er bis jetzt keine Zeit hatte sich damit zu beschäftigen, dies aber geplant habe (A.1.31).
AD4 meldete sich darauf und hatte das Problem, dass er Saros zwar installieren konnte, aber keine Verbindung zu einem Jabber-Server aufbauen konnte. Dieses Problem konnte nicht gelöst werden, da er einen Proxy[1] mit Authentisierung benutzte, was Saros nicht unterstützt (A.1.31).
Die Entwickler in dem Kanal waren sehr beschäftigt und diskutierten über zu lösende Defekte von ADempiere. Ich versuchte herauszufinden, ob jemand mit Saros arbeitete. Ich konnte in meinem Roster sehen, dass AD2 mit Saros online war, aber nicht, ob er oder andere auch wirklich mit Saros arbeiteten. Auf meine Anfrage danach erwiderte er, dass er Saros gerade nicht nutze und momentan auch mit der Lösung einiger Defekte von ADempiere beschäftigt sei. Er wolle Saros in den nächsten Tagen mit AD1 testen, aber nicht während des Bug Days, da da beide sehr viel zu tun hätten. Er würde mich dann über das Resultat informieren. (A.1.31)

5 Wird innerhalb von Eclipse als Flag für Dateien benutzt, welche nicht direkt editierbar sein sollen

1 Ein Proxy ist eine Kommunikationsschnittstelle in einem Netzwerk und arbeitet als Vermittler, der Anfragen stellvertretend entgegennimmt und dann an den Anfragenden weiterreicht

Ich bedankte mich daraufhin und blieb bis 17:31 Uhr im Kanal, für den Fall das noch Fragen kämen, unterließ aber weitere Nachfragen, da die Entwickler zu beschäftigt waren und ich das Gefühl hatte, dies wäre zu aufdringlich gewesen.

5.8.6.2 Fazit der zweiten Iteration

Der Workaround war zu spät verfügbar. AD2 hatte die Anleitung nicht gelesen und konnte so Saros an dem Tag auch nicht anwenden. Während des Bug Day waren die Entwickler zu sehr in ihr eigenes Projekt vertieft, als dass sie sich noch in Saros hätten einarbeiten können. Dafür hätte es im Vorfeld schon ein kritische Masse geben müssen, die Saros schon getestet hätten. Dies wurde bereits bei Drupal angemerkt.

> „Time is very tight, so any delays messing with connection problems, etc. aren't really acceptable. So, both parties would need to have already used it in the past and frequently for it to be usable during crunch time.“ [DR5 A.1.19]

Die zwei Wochen zwischen der Kontaktaufnahme und dem forcierten Bug Day haben nicht gereicht, um diese kritische Masse aufzubauen und hinderliche Defekte zu lösen.
Es wurde während der 2. Iteration folgender Defekt gefunden:

- Saros doesn't work with AuthProxy[1]
  Beschreibt das von AD4 gefundene Problem, dass Saros keine Verbindung über Proxys mit Authentisierung herstellen kann.

5.8.6.3 Weiteres Vorgehen

Da bei AD1 der letzte Stand war, dass er Saros installiert hatte, aber sich nicht zu einem Jabber-Server verbinden konnte, vgl. Abschnitt 5.8.5.1, erschien es sinnvoll in der nächsten Iteration den Ursprung dieses Problems herauszufinden.

1 https://sourceforge.net/tracker/?func=detail&aid=2906729&group_id=167540&atid=843359

5.8.7 Dritte Iteration

Ich versuchte in dieser Iteration das Problem von AD1 zu lösen.

5.8.7.1 Resultate

Die Fehlermeldung die AD1 bekam, wenn er versuchte sich zu einem Jabber-Server zu verbinden, war:

> „Could not connect to Server 'jabber.org' as AD1. Please check your username and password in the preferences“

Zunächst vermutete ich, dass der Benutzername oder das Passwort falsch geschrieben war. AD1 war sich aber sicher, das beides korrekt war und schickte mir den Benutzernamen und das Passwort zu, damit ich mich selbst überzeugen konnte. Mit diesen Daten war es mir möglich mich zu jabber.org zu verbinden.
Da dies nicht das Problem war, schlug ich vor einen anderen Jabbercilent zu benutzen, um herauszufinden, ob das Problem bei Saros oder an etwas anderem liegt. Er installierte daraufhin Adium[1] und konnte sich, nach ein paar kleineren Widrigkeiten, auch zu dem Jabber-Server verbinden. So musste das Problem innerhalb von Saros oder Eclipse liegen. (A.1.32, A.1.32) Daraufhin gingen wir die Eclipse-Netzwerkeinstellungen durch, da ich vermutete, das diese falsch eingestellt waren. Dies half aber ebenfalls nicht. (A.1.32)
Ich erkläre ihm daraufhin, dass ich seine Saros-Logdateien benötige, um das Problem besser verstehen zu können. Es handelte sich sehr wahrscheinlich um ein Saros-Problem. Ich erklärte ihm was er in den Saros-Einstellungen aktivieren muss, damit seine Logdateien unten seinen eingetragenen Pseudonym automatisch an unseren Server geschickt werden. Die Einstellungen wurden von ihm vorgenommen, aber die Logdateien waren auf unserem Server nicht aufzufinden.

1 http://adium.im/

5.8.7.2 Fazit der dritten Iteration

Ich vermutete, dass es einen Defekt in der Zustellung von Logdateien gab und beriet mich mit dessen Entwicklerin. Ein lokaler Test zwigte aber, das die Logdateien durchaus problemlos übertragen werden. Daher musste das Problem einen anderen Ursprung haben.

5.8.7.3 Weiteres Vorgehen

Sobald das Partially Sharing einwandfrei funktioniert, wäre es sinnvoll eine weitere Iteration in diesem Projekt durchzuführen. Vorher sehe ich darin keinen Sinn, da

1. AD1 sich aus nicht bekannten Gründen zu keinem Jabber-Server verbinden kann.
2. AD4 sich zu keinem Jabber-Server verbinden kann, da Saros keine Proxys mit Authentisierung unterstützt und dieses Problem auch vorerst nicht gelöst werden kann.
3. AD2 Saros zwar benutzen kann, aber ADempiere zu groß ist, um über Saros verteilt zu werden.

5.9 Openbravo

5.9.1 Das Projekt

Openbravo ist eine Firma, welche eine gleichnamige ERP- und PoS[1]-Software herstellt[OPE09b]. Openbravo ERP liegt in der Openbravo Public Lincense vor, welche auf der Mozilla Public License Version 1.1 basiert[OPE09c]. Openbravo PoS in der General Public License[OPE09d].

1 Point of Sales

5.9.2 Geschichte

Openbravo wurde Mitte der neunziger Jahren in der Engineering School Tecnun an der Universität Navarra in Spanien von Nicolas Serrano und Ismael Ciordia gegründet und war ursprünglich als Universitäts-Managementsystem gedacht. Im August 2001 gründeten die beiden und Moncho Aguinaga die Firma Tecnica, heute bekannt als Openbravo, um die Software auch auf Unternehmen auszuweiten[OPE09a].

5.9.3 Das Team

Das Team besteht heute aus über 100 Mitarbeitern. Das Management-Team umfasst neun Mitarbeiter, die teilweise ebenfalls leitende Positionen innerhalb der Firma inne haben. Hinweise auf die Anzahl der Entwickler konnten nicht gefunden werden.

5.9.4 Projektauswahl

5.9.4.1 Begründung

Openbravo gleicht technisch dem Projekt ADempiere mit dem ich gute Resultate erzielt hatte. Es ist ebenfalls in Java geschrieben und betreibt eine öffentlich einsehbare Entwicklungskommunikation. Das Projekt ist mit seiner Mitarbeiterzahl groß genug um Saros benutzen zu können. (Vgl. Abschnitt 4.2.2.1.

5.9.4.2 Kontaktaufnahme

Über die Entwicklermailingliste sind seit 2008 nur 63 Nachrichten verschickt worden. Dies legte die Vermutung nahe, dass die Hauptkommunikation über andere Kanäle geführt wird. Das Projekt betreibt ebenfalls 19 Foren, wobei davon nur zwei thematisch für eine Kontaktaufnahme in Frage kamen: *Contributing to Openbravo ERP* und *Developers*. Nach genauerer Analyse stellte ich jedoch fest, dass auch diese Foren nicht so intensiv genutzt werden, wie ein Projekt dieser Größe vermuten lässt. In dem *Contributing to Openbravo ERP*-Forum gab es kaum Einträge, Maximum waren drei Einträge pro Monat im Jahr 2009 und in den letzten sieben Tagen wurde gar kein Eintrag verfasst, was ein Ausschlusskriterium dieses Forums war

[SOU10h]. Das Developers-Forum hatte immerhin eine Aktivität von 324 Einträgen pro Montag im Jahr 2009, das sind ca. 10 Einträge pro Tag, womit sich diese Liste von der Aktivität her durchaus eignete [SOU10i].
Da das Projekt von einer Firma geleitet wird, zog ich die Kontaktaufnahme über den Maintainer ebenfalls in Betracht. Das Projekt richtet sich aber auch gezielt an Universitäten, um an dem Projekt teilzunehmen.

> „If you are responsible for a program in an academic institution and you have interest in participating in this program, please contact us at collaborate@openbravo.com.“ [OPE10a]

Deshalb entschiede ich mich, das Projekt über die angegebene Emailadresse zu kontaktieren (A.1.30).

5.9.5 Resultate

Trotz mehrmaliger Anfrage erhielt ich von diesem Projekt keine Antwort.

5.9.6 Fazit

Openbravo wurde nicht über die Mailingliste oder ein Forum kontaktiert. Zwar wurde hier nicht, wie bei Vuze und Eclipse-cs, der/die Maintainer angeschrieben, obwohl sich natürlich nicht sagen lässt, welche Personen die Email über collaborate@openbravo.com erhalten haben. So ist nicht klar, wie diese Kontaktaufnahme zu interpretieren ist. Es lässt sich aber feststellen, dass bis jetzt jede Form der Kontaktaufnahme, welche nicht über eine Mailingliste bzw. an alle Entwickler direkt durchgeführt wurde, nicht erfolgreich war. Ferner scheint sich, wie bei Vuze schon vermutet, ein von einer Firma geleitetes Projekt weniger für eine Anfrage von außen zu eignen (vgl. Abschnitt 5.2.6).

5.10 OpenProj

5.10.1 Das Projekt

OpenProj ist eine Projektmanagement-Software, welche das Ziel, hat Microsoft Project[1] und andere kommerzielle Produkte zu ersetzen und ist so kompatibel mit in Microsoft Project angelegten Projekten[OPE10b]. Angelegte Projekte werden u.a. mit Hilfe von einem Gantt-Diagramm[1] und PERT-Diagramm[2] visualisiert und geplant. Das Projekt ist in Java geschrieben und unter der CPALv1[3][4] verfügbar.

5.10.2 Geschichte

Die Software wurde ursprünglich von der Firma *Projity Incorporated* entwickelt. Heute gehört Projity Incorporated zu der Firma *Serena*, welche auch OpenProj weiterentwickelt.

5.10.3 Das Team

Von Serena wird nicht angegeben, wie viele Entwickler an OpenProj arbeiten, es sind aber 14 Entwickler bei der Projekt-Seite auf Sourceforge eingetragen, davon sind vier Personen Projekt Administratoren [SOU10o].

1 http://office.microsoft.com/de-de/project/default.aspx
1 Ein Gantt-Diagramm oder Balkendiagramm dient der Darstellung einer zeitlichen Abfolge von Ereignissen oder Aktivitäten. Erfunden wurde es von Henry L. Gantt und wird häufig im Projektmanagement eingesetzt
2 Ein PERT-Diagramm stammt aus der Netzplantechnik und dient der Darstellung von Abhängigkeiten im zeitlichen Ablauf von Projekten
3 Common Public Attribution License 1.0
4 http://opensource.org/licenses/cpal_1.0

5.10.4 Projektauswahl

5.10.4.1 Begründung

Das Projekt wurde ausgewählt, da es in Java geschrieben ist, eine große Entwicklergemeinde hat und die leitende Firma im Bereich der agilen Softwareentwicklung tätig ist[SER10]. Da das Projekt Vuze und Openbravo ähnelt, können sich in diesem Zusammenhang dort aufgestellte Vermutungen verifizieren bzw. widerlegen lassen.

5.10.4.2 Kontaktaufnahme

Das Projekt konnte über zwei Wege kontaktiert werden, zum einen über die Entwickler-Mailingliste, zum anderen über die Projekt-Administratoren. Es stellte sich aber heraus, dass die Mailingliste kaum benutzt wird (zwei Emails im Jahre 2009).[SOU10n] So blieb nur noch die Kontaktaufnahme über die Maintainer. Hierfür wurde wieder das Sourceforge-Kontaktformular benutzt, da die Emailadressen der Maintainer auf anderem Wege nicht verfügbar waren.
Der Emailtext wurde überarbeitet. Der Begriff *Distributed Paiprogrammming* wurde im Betreff durch *Collaborative Development* ersetzt. Im Text wurde der Zusammenhang von Saros mit agiler Softwareentwicklung klarer herausgearbeitet und die Funktionsweise von Saros genauer beschrieben (vgl. Abschnitt A.1.33).

5.10.5 Resultate

Die Maintainer wurden insgesamt zweimal angeschrieben, jedoch erhielt ich keine Antwort von ihnen.

5.10.6 Fazit

OpenProj war das vierte Projekt von dem ich keine Antwort erhalten habe. Der Kontakt fand wie bei Vuze, Elcipse-cs und Openbravo nicht über eine Mailingliste statt, sondern über die Maintainer. Das Projekt ähnelt Vuze am meisten, da es ebenfalls von einer Firma geleitet wird, welche keinen öffentlich einsehbaren Entwicklungsprozess betreibt. Zwar sind Bug- und Issuetracker vorhanden, aber

keine öffentliche Entwicklungs-Mailingliste oder ein Forum. Es erhärtet sich also der Verdacht, dass Projekte dieser Art für meine Anfragen nicht geeignet sind.

5.11 Tomcat

5.11.1 Das Projekt

Tomcat ist eine Implementierung der *Java Server Pages*[1] und der *Java Servlet*[2] Spezifikationen. So dient Tomcat dazu Java-Programme auf Webservern ausführen und übersetzen zu können. Tomcat verfügt über einen eigenen Webserver, lässt sich aber auch über das Connector-Plugin *mod_jk*[1] in einen Apache Webserver einbinden. Das Projekt ist Teil der *ASF*[2][3] und ist so in der *Apache Software License*[4] verfügbar[APA10b].
Die gemeinnützige Apache Software Foundation unterstützt Open Source Projekte, welche Teil der Apache Community sind. Alle Apache-Projekte müssen unter der Apache Software License veröffentlicht werden und zeichnen sich durch eine basisdemokratische Entscheidungsfindung aus. Die Entwicklungsprozesse sind offen und die Projekte diskussionsfreudig. Die Mitglieder der ASF verstehen sich als große Gemeinschaft und Projekt übergreifende Zusammenarbeit ist häufig und erwünscht [APA10a] [APA10c].

5.11.2 Geschichte

Das Projekt Tomcat wurde von *Sun Microsystems*[5] als eine Referenzimplementierung von der Java Servlet und Java Server Pages Spezifikation gestartet. Im Jahr 1999 wurde der Quelltext der Apache Foundation zur Verfügung gestellt und die

1 http://java.sun.com/products/jsp
2 http://java.sun.com/products/servlets
1 http://tomcat.apache.org/connectors-doc/
2 Apache Software Foundation
3 http://www.apache.org/
4 http://www.apache.org/licenses
5 http://www.sun.com/

erste Open Source Version veröffentlicht, die Tomcat Version 3.0. Von da an wurde Tomcat von Sun und anderen Organisationen weiterentwickelt und eine aktive Benutzergemeinde entstand.

5.11.3 Das Team

Das Entwicklungsteam besteht aus 52 Mitglieder, davon sind 20 aktiv. Die übrigen sind aber noch an dem Projekt interessiert und verfügen ihre alten Rechte, wie z.B. Commit-Rechte. Diese Gruppe wird *Emeritus Committers* genannt. Das Team bezeichnet sich selbst als meritokratisch organisiert, d.h. je mehr jemand in das Projekt einbringt, desto mehr Anerkennung und Gewicht wird er bekommen[APA10d].

5.11.4 Projektauswahl

5.11.4.1 Begründung

Das Projekt wurde ausgewählt, weil es in Java geschrieben ist und zur Entwicklung Eclipse eingesetzt wird. Dies erkannte ich daran, dass im Repository eine .project-Datei vorhanden war. Apache Projekte sollen, wie schon erwähnt, offen gegenüber Neulingen sein und einen öffentlichen Entwicklungsprozess führen. [APA10e].

5.11.4.2 Kontaktaufnahme

Die Möglichkeit der Kontaktaufnahme war bei diesem Projekt eindeutig, es kam nur die Mailingliste in Frage, da dies das Hauptkommunikationsmedium des Projektes ist. Ferner legte das Projekt Wert auf eine möglichst hierarchiearme Organisationsstruktur, weshalb es sich nicht anbot, das Projekt über einen Maintainer zu kontaktieren. Als Mailingliste bot sich nun die Entwickler-Liste an, auf der die Entscheidungen und Diskussionen des nächsten Release besprochen werden. Die Mailingliste war die aktivste, die ich bis jetzt kontaktiert hatte, im Jahre 2009 wurden 9.977 Emails über die Liste verschickt, das sind durchschnittlich ca. 831 im Monat und ca. 27 Emails pro Tag (vgl. 20,4 Emails pro Tag bei WordPress und 19,4 Emails pro Tag bei dev-typo3). Jedoch waren ein Großteil dieser Emails

Tracker-Benachrichtigungen und Commit-Logs, so dass nicht genau zu erkennen ist, wie viele Emails zu werten sind.

5.11.5 Resultate

Auf meine Anfrage wurde auf der Mailingliste nicht geantwortet.

5.11.6 Fazit

Die Tomcat-Mailingliste, war nach der dev-typo3-Liste die zweite Mailingliste, auf der keine Reaktionen auf meine Anfrage erfolgten. Bei der dev-typo3-Liste stellte ich die Vermutung auf, dass diese Mailingliste eine zu große Aktivität aufweist, so dass Anfragen schnell untergehen können (vgl. Abschnitt 5.1.4.2). Die Liste hatte, wie bereits erwähnt, eine Aktivität von 19,4 Emails pro Tag auf typo3-dev. Da die Tomcat-Mailingliste eine noch höhere Aktivität aufweist, lässt sich auch hier vermuten, dass meine Anfrage wahrscheinlich untergegangen ist.
So wäre zu überlegen, ob der Betreff der Email ansprechende formuliert werden sollte. Jedoch war der Betreff bei dev-typo3 und Tomcat nicht gleichlautend, da wie bei OpenProj nicht mehr der Begriff „Distributed Pair Programming“ benutzt wurde, sondern der Begriff „Collaborative development“. Dadurch ist nicht klar, ob diese Änderung eine Verbesserung darstellt.

5.12 Hibernate

5.12.1 Das Projekt

Hibernate ist ein Persistenz[1] - und ORM[2]-Framework für Java und dient dazu Objekte mit Attributen und Methoden in relationalen Datenbanken abzubilden. Ferner kann Hibernate als Datenbankabstraktionsebene dienen, da Programme

1 Persistenz bezeichnet die Möglichkeit, im Arbeitsspeicher vorhandene Objekte in Datenbanken oder Dateisystemen abzulegen
2 Objektrelationale Abbildung, engl. englisch object-relational mapping

über Hibernate auf die Datenbank zugreifen und Hibernate die notwendigen SQL Statements, in Abhängigkeit von der Datenbank, generiert. Hibernate ist unter der LGPL[3] verfügbar [HIB10].

5.12.2 Geschichte

Die Entwicklung von Hibernate begann Ende im Jahr 2001 und wurde von Gavin King initiiert, um die Möglichkeit zu schaffen auch komplexe Datenmodelle zu skalieren. Hibernate wurde als unabhängiges Open Source Produkt entwickelt. Jedoch wurde schnell klar, dass um den zukünftigen Erfolg des Projektes sicherzustellen, es nötig ist, professionelle Entwickler in Vollzeit mit dem Projekt zu beschäftigen. Seit Ende 2003 gehört Hibernate zu JBoss Inc.[1], welche kommerziellen Support und Training vertreiben [BC07].

5.12.3 Das Team

Das Hibernate Team besteht aus 37 Mitglieder, davon sind 23 Entwickler und sieben Projekt-Administratoren [SOU10l].

5.12.4 Projektauswahl

5.12.4.1 Begründung

Das Projekt von vielen Entwicklern getragen wird, von denen einige ebenfalls an Eclipse-Plugins und an der Integration von Hibernate in Eclipse arbeiten. Daher ist es wahrscheinlich, dass einige der Entwickler ebenfalls mit Eclipse arbeiten [SOU10k]. An Hibernate wird aktiv gearbeitet und ein Bug- und ein Issuetracker sind vorhanden, was auf einen öffentlichen Entwicklungsprozess hinweist.

3 GNU Lesser General Public License
1 http://www.jboss.com/

5.12.4.2 Kontaktaufnahme

Bei Hibernate waren mehrere aktive, öffentlich zugängliche Mailinglisten vorhanden. Die Kontaktaufnahme fand über die Entwickler-Mailingliste statt, da diese die einzige war, die thematisch zu meiner Anfrage passte und eine Kontaktaufnahme per Maintainer wieder nur über das Sourceforge-Kontaktformular möglich gewesen wäre. Mit 1.111 Emails im Jahre 2009, das sind ca. 95 Emails pro Monat und ca. 3 Emails pro Tag, lag die Aktivität der Mailingliste zwischen TYPO3v5 und Drupal [JBO10].

5.12.5 Resultate

Ich erhielt eine Antwort von HI1, der PP für nützlich hielt, wenn es auch seiner Erfahrung nach selten in OSS-Entwicklung Anwendung findet, da die Entwickler weit verstreut arbeiten. Er merkte aber an, dass in Hibernate schon diskutiert wurde, ob nicht die Peer-Review Software *Crucible* [1] für Quelltext-Durchsichten benutzt werden könne. Doch weder in der Mailingliste noch in den Benutzerforen konnte ich eine derartige Diskussion finden, welche durchaus interessant gewesen wäre. Er selbst könne Saros aber nicht benutzen, da er Eclipse nicht verwendet. Weitere Antworten erhielt ich trotz Nachfragen nicht. Zusammenfassend ergibt sich folgendes Bild:

1. Ablehnung
   a) Technische Ablehnung
      i. Mangelhaftes Produkt
         - Abhängigkeit von einer Entwicklungsumgebung (Eclipse):

           „Unfortunately I personally do not use eclipe so I myself am not a good fit for this.“ [HI1 A.1.36]
   b) Umfeld bezogene Ablehnung

1 http://www.atlassian.com/software/crucible/

i. Konzept unpassend für das Projekt

- Könnte nützlich sein, aber nicht für uns:

  „Pair programming is an incredible development tool that in my exerience is not generally used on OSS dev due to developers not being located close together.“ [HI1 A.1.36]

2. Akzeptanz

   a) Technische Akzeptanz

      i. Sinnvolles Produkt

- Interesse/sinnvoller Vorschlag:

  „Hey eike this sounds really interesting.“ [HI1 A.1.36]

- Schon nach so etwas gesucht:

  „We have been discussing using Crucible for Hibernate development for exactly alot of the points you list.“ [HI1 A.1.36]

Die Aussage in der Kategorie "*Könnte nützlich sein, aber nicht für uns*ist hierbei nicht klar auf Hibernate bezogen, dennoch denke ich, dass HI1 als Mitentwickler Hibernate in seine gemachten Erfahrungen einbezieht, was diese Kategorisierung rechtfertigt.

5.12.6 Fazit

Es fällt bei diesem Projekt ebenfalls schwer, ein Fazit zu ziehen, da zwar klare Aussagen gemacht, diese aber nur von einer Person getroffen wurden. Festzuhalten ist, dass in diesem Projekt ein Interesse an Produkten besteht, welche Peer-Reviews ermöglichen bzw. erleichtern. Leider war es, wie bereits erwähnt, nicht möglich diese Diskussion zu analysieren. Es könnte somit durchaus sein, dass dort schon eine Entscheidung gegen derartige Software getroffen wurde.
Dieses Projekt war das erste von einer Firma geleitete Projekt, welches ich über eine Mailingliste kontaktiert habe und damit auch das erste dieser Art, von dem

ich eine Antwort erhielt. So kann vermutet werden, dass es erfolgversprechender ist, Firmen-Projekte ebenfalls über die Mailinglisten bzw. ein Forum zu kontaktieren, als über die/den Maintainer.

5.13 Spring

5.13.1 Das Projekt

Spring ist ein Java-Framework mit dem Ziel die Entwicklung von Java/ Java EE[1] zu vereinfachen. Das Framework richtet sich dazu nach folgenden Prinzipien/Entwurfsmustern:

- Dependency Injection: Abhängigkeiten zwischen Objekten und Komponenten werden minimiert.
- Aspekt-orientierte Programmierung (AOP): Logische Aspekte des Anwendungsprogramms werden getrennt voneinander entwickelt.
- Vorlagen vereinfachen die Arbeit mit Programmierschnittstellen.

Es ist möglich Spring mit vielen andern Frameworks und Technologien zu kombinieren, z.B. auch mit Hibernate [SPR10a].

5.13.2 Geschichte

Spring basiert auf dem im Jahr 2002 erschienenen Buch *One-On-One J2EE Design and Development* von Rod Johnson, dem er den Spring-Quelltext beilegte [Joh02]. Im Jahr 2003 wurde das Spring-Projekt gegründet und war bei Sourceforge unter der Apache 2.0 Lizenz verfügbar. Die 1.0 Version wurde ein Jahr später veröffentlicht

1 Java Platform, Enterprise Edition ist ein Industriestandard für Java Programme, welcher hauptsächlich im Web-Bereich Anwendung findet

und im Jahre 2006 gewann Spring einen *JAX Innovation Award*[1] und einen *Jolt productivity award*[2].

5.13.3 Das Team

Das Team besteht aus 51 Mitgliedern, wobei 42 Mitglieder als Entwickler und davon sechs als Projekt-Administratoren, unter ihnen Rod Johnson, klassifiziert sind [SOU10r]. Alle aktiven Entwickler werden von der Firma SpringSource beschäftigt, welche seit September 2009 zu VMware gehört [HEI10].

5.13.4 Projektauswahl

5.13.4.1 Begründung

Spring hat eine ähnliche Entwicklung wie Hibernate durchgemacht und lässt sich auch vom Einsatzgebiet und Größe her gut mit Hibernate vergleichen. Das Projekt ist in Java geschrieben und verwendet Eclipse als IDE, da eine .project-Datei im Repository vorhanden ist. Mit 42 Entwicklern erscheint das Projekt ausreichend groß, um Saros sinnvoll anwenden zu können. Ein öffentlicher Bug- und Issuetracker sind ebenfalls vorhanden.

5.13.4.2 Kontaktaufnahme

Das Projekt besitzt eine Entwickler-Mailingliste, über welche jedoch sehr viel SPAM verschickt wird [SOU10f]. Daher erschien es mir unwahrscheinlich, dass die Liste wirklich für die Entwicklung von Spring genutzt wird. Nach genauerer Recherche fand ich heraus, dass die Entwickler mehrere Foren für die Entwicklung benutzten, welche nicht bei Sourceforge, sondern bei auf der SpringScource Webseite liegen. Insgesamt verfügt Spring über 54 Foren, wobei es in 22 Foren um Projekte, die den Kern von Spring betreffen, geht. Die drei aktivsten Foren waren *Core Container* mit

1 http://www.heise.de/newsticker/meldung/Spring-Framework-gewinnt-JAX-Innovation-Award-2006-123689.html
2 http://www.drdobbs.com/architect/187900423?pgno=10

37.321 Einträgen, *Data Acces* mit 51.283 Einträgen und *Web* mit 51.511 Einträgen. Ich erstellte einen Eintrag in dem Core Container Forum (A.1.36), da ich dieses Forum thematisch am passendsten fand, da es hier um die Entwicklung des Spring-Kerns geht und u.a. Rod Johnsons Moderator des Forum ist [SPR10b]. Die 37321 Einträge sind von August 2004 an gezählt worden, was bedeutet, dass das Forum eine Aktivität von durchschnittlich ca. 2-3 Einträgen pro Tag hat. Zwar wird das Projekt von einer Firma geleitet, womit ich bis jetzt schlechte Erfahrungen gemacht habe, aber Spring verfügt über öffentlich einsehbare Entwicklungsdiskussionen, was bei Projekten dieser Art nur bei Hibernate und Openbravo der Fall war. Jedoch wurde Openbravo über die Maintainer kontaktiert.

5.13.5 Resultate

Mein Eintrag wurde zwar 146 mal gelesen, ich erhielt aber keine Antwort [SPR10c].

5.13.6 Fazit

Spring war das zweite von einer Firma geleitete Projekt, welches über ein Forum/eine Mailingliste kontaktiert wurde. Bei Hibernate erhielt ich eine einzige Antwort, bei Spring hingegen keine. Es lässt sich also vermuten, dass nicht die Kontaktart (Maintainer vs. Forum/Mailingliste), sondern die Art des Projektes an sich das Problem ist. Mit der erwähnten Ausnahme von Hibernate hat kein Projekt, das von einer Firma geleitet wird, auf meine Anfragen geantwortet. Dabei ist, wie schon erwähnt, immer noch offen, ob Saros für Projekte dieser Art nicht geeignet ist, oder diese Projekte verschlossener gegenüber Anfragen von außen sind. Zwar hat das Forum eine durchschnittliche Aktivität von ca. zwei bis drei Einträgen pro Tag. Es könnte ebenfalls möglich sein, das mein Eintrag untergegangen ist. Dagegen spricht allerdings, dass an diesem Tag erstellte Einträge, auf welche ebenfalls nicht geantwortet wurde, ähnlich oft aufgerufen worden sind [SPR10c]. Die Anzahl der Aufrufe von meinem Beitrag müsste sich aber von den anderen unterscheiden, wenn mein Beitrag übersehen worden sein sollte.

6 Ergebnisse

In diesem Kapitel werde ich die Einzelergebnisse der Projekte zusammenfassen und analysieren. Resultierend daraus werden mögliche Antworten zu den einzelnen Forschungsfragen dargelegt. Es wird aber auch aufgezeigt, wo das ursprünglich formulierte Ziel nicht erreicht werden konnte. Abschließend erfolgt eine Bewertung der methodischen Vorgehensweise.

6.1 Forschungsfragen

Entgegen der in Abschnitt 3 beschriebenen Vermutung, konnte Saros in keinem Projekt als Prozess integriert werden. Somit lässt sich die Forschungsfrage *Wie ist Saros in OSS Projekten einsetzbar* nicht beantworten. Es können aber indirekte Rückschlüsse auf die Frage in *welchen Kontext Saros einsetzbar ist* gezogen werden, was später im Abschnitt 6.4 noch näher ausgeführt wird.
Welche Folgen sich aus dem Einsatz von Saros in OSS-Projekten ergeben lässt sich nur in Bezug auf Saros beantworten. Da hingegen die Projekte Saros nicht eingesetzt haben, konnte nicht beantworten werden, welche Folgen sich für die *Projekte* durch den Einsatz von Saros ergeben.
So wurden vor allem Erkenntnisse gesammelt, die die Frage betreffen, wie Saros in OSS Projekten eingeführt werden kann. Diese werden in den zwei nächsten Abschnitten besprochen.

6.2 Kontaktaufnahme

Die Kontaktaufnahme ist ein wesentlicher Bestandteil der Innovationseinführung. Durch Auswertung der vollzogenen Kontaktaufnahmen lassen sich Rückschlüsse auf die Forschungsfrage *Wie kann Saros in OSS-Projekten eingeführt* ziehen. So wird gezeigt, welche Kontaktmethode erfolgreicher war: Mailingliste/Forum oder Maintainer.

Bevor ein Projekt kontaktiert wird, sollte bekannt sein, welche Programmiersprache diese verwenden, um im Vorfeld die vorhandenen Eclipse-Plugins zu testen. Um den Entwicklern aufweisen zu können, welche Plugins und Sprachen von Saros unterstützt werden, wurde eine Webseite angelegt und verschieden Plugins getestet[1]. Daraus folgte eine Bachelorarbeit[2], mit dem Ziel weitere Plugins mit Saros zu testen.

In der Tabelle 6.1 sind alle kontaktierten Projekte aufgelistet. Dabei ist angegeben, ob sie von einer Firma geleitet werden oder ob sie gemeindebasiert sind. Die Anzahl der *Antworten* beinhaltet nur wieviele *direkte* Antworten ich auf meine Anfrage hin erhalten habe. So zählte ich nur Antworten, die sich auf meinen Kontakttext bezogen und nicht auf eventuelle andere Antworten von anderen Entwicklern oder mir. Viele Entwickler sind erst dann in die Diskussion eingestiegen als schon ein paar Antworten vorhanden waren. Diese lassen aber nur wenig Rückschlüsse auf die gewählte Kontaktmethode zu. Die *öffentliche Kommunikation* bezieht sich nur auf Kommunikation, die die Kernentwicklung besprechen. D.h. ein Projekt verfügt nur dann über eine *öffentliche Kommunikation*, wenn das Medium, indem Diskussionen stattfinden die die Kernentwicklung betreffen, öffentlich einsehbar und zugänglich ist.

Vergleich man zunächst die verschiedenen Kontaktmethoden miteinander so erkennt man, dass kein Projekt, das über den Maintainer kontaktiert wurde, reagiert hat. Wohingegen von den Projekten, die über Mailingliste/Forum kontaktiert

1 https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugins
2 https://www.inf.fu-berlin.de/w/SE/ThesisDPPXVI

Projekt	Organisation	Antworten	Kontaktart	Nachrichten pro Tag	öffentliche Kommunikation
TYPO3 5.0	Gemeinde	3	Mailingliste	1,2	ja
TYPO3 4.3	Gemeinde	1	Mailingliste	4,2	ja
Freenet	Gemeinde	1	Mailingliste	6,5	ja
Drupal	Gemeinde	3	Mailingliste	6,3	ja
WordPress	Gemeinde	1	Mailingliste	20,4	ja
JUnit	Gemeinde	1	Mailingliste	0,3	ja
ADempiere	Gemeinde	2	Forum	9,1	ja
Hibernate	Firma	1	Mailingliste	3	ja
Eclipse-cs	Gemeinde	0	Maintainer	-	nein
Vuze	Firma	0	Maintainer	-	nein
Openbravo	Firma	0	Maintainer[1]	-	ja
OpenProj	Firma	0	Maintainer	-	nein[2]
Tomcat	Gemeinde	0	Mailingliste	27	ja
Spring	Gemeinde	0	Forum	2,6	ja
TYPO3 dev	Gemeinde	0	Mailingliste	19,4	ja

[1] Das Projekt wurde über eine Emailadresse kontaktiert, die speziell für Universitäten gedacht war, vgl. Abschnitt 5.9.4.2
[2] Die Mailingliste wurde kaum benutzt (zwei Emails im Jahre 2009).

Tabelle 6.1: Überblick über die druchgeführen Kontaktmethoden.

wurden, sieben von zehn antworteten, einige davon sogar bis zu drei mal. Daher erhärtet sich der Verdacht, dass die Art der Kontaktaufnahme über die Mailingliste/Forum die erfolgversprechendere Methode ist.

Die zweithäufigste Ablehnung meiner Anfrage ist die Kategorie *Schlechte Erfahrungen mit Pair Programming*, vgl. Tabelle 6.3. Hierbei wurde Saros bzw. DPP von den Entwicklern mit PP gleichgesetzt. Dass Saros neben DPP noch vielfältigere Möglichkeiten bietet, ist dabei nicht beachtet worden. Es ist also ratsam Saros in der Kontaktaufnahme nicht mit PP in Verbindung zu setzen. Der Begriff DPP ist nicht so verbreitet und legt die Verbindung zu PP nahe, daher wurde dieser umschrieben durch *collaborative development*. Dies habe ich schon bei WordPress festgestellt und ab dem Projekt Openbravo auch in meiner Kontaktnachricht berücksichtigt. Allerdings konnte ich dadurch keine signifikanten Verbesserungen nachweisen.

Nur zwei von insgesamt neun kontaktierten Mailinglisten zeigten keine Reaktion auf meine Anfrage. Die beiden Listen hatten im Vergleich zu den Anderen eine hohe Aktivität. Nur die WordPress-Liste zeigte eine ähnlich hohe Aktivität. Es lässt sich daher vermuten, dass es ratsamer ist, Mailinglisten zu kontaktieren, die keine allzu hohe Aktivität aufweisen, ca. zehn Emails pro Tag.
Die Foren ähnelten den Mailingliste insofern als das in beiden Fällen alle Entwickler angeschrieben wurden und öffentlich auf die Beträge zugegriffen werden konnte.
Wie schon bei ADempiere festgestellt, war es sehr hilfreich sich nach der Kontaktaufnahme in den IRC Kanal des Projekte einzuloggen, so dieser vorhanden ist. So bekommt man unmittelbares Feedback und es lässt sich schnell eine fruchtbare Diskussion führen, wenn auch die Auswertung dieser aufwendiger ist, vgl. Abschnitt 5.8.5.2.

Der Kontakt zu ADempiere war der erfolgreichste, vor allem wenn man auch die nachfolgende Diskussion und das Testen von Saros miteinbezieht. Bei diesem Projekt habe ich einen gezielten Vorschlag gemacht, wo und wann Saros innerhalb des Projektes einsetzbar sein könnte, vgl. Abschnitt 5.8.4.2. Durch diesen gezielten Vorschlag konnte ich zeigen, dass ich mich mit dem Projekt befasst hatte und Ideen für konkrete Anwendungsmöglichkeiten liefern. Ich vermute, dass dies ein sehr erfolgversprechendes Vorgehen ist. Zwar wurde bei TYPO3 dem Security Team ein ähnlicher Vorschlag unterbreitet, dieser wurde aber mit der Begründung abgelehnt, dass Saros nicht in ihre Arbeitsabläufe passen würde. So wurde die Möglichkeit zumindest geprüft, weshalb die Kontaktaufnahme an sich positiv zu werten ist. Um Anwendungsmöglichkeiten zu finden ist es notwendig, dass viele Informationen über das Projekt zu Verfügung stehen und ihre Arbeitsprozesse öffentlich einsehbar sind.

6.3 Projekte

Im folgenden wird die Struktur der Projekte untersucht, bei denen die Kontaktaufnahme erfolgreich war. Es werden Kategorien herausgearbeitet, welche Projekte sich für eine Kontaktaufnahme für Saros eigenen. Dieser Abschnitt liefert so Antworten auf die Forschungsfrage "*Wie kann Saros OSS Projekten eingeführt werden*".

	Akzeptanz	**Anlehnug**	**Organisation**	**Eclipse Nutzung**	**Sprache**
TYPO3	9/4[1]	9/5	Gemeinde	variiert	PHP
Drupal	9/5	13/4	Gemeinde	variiert	PHP
WordPress	0	2/2	Gemeinde	variiert	PHP
Freenet	0	3/2	Gemeinde	ja	Java
JUnit	1/1	0	Gemeinde	ja	Java
ADemiere	8/3	0	Gemeinde	ja	Java
Hibernate	2/2	2/1	Firma	ja	Java

[1] Anzahl der Aussagen/Anzahl der Entwickler

Tabelle 6.2: Akzeptanz/Ablehnung von gesponsorten/gemeindebasierten Projekten

6.3.1 Organisation

Von vier kontaktierten Projekten, die von einer Firma geleitet werden, antwortete nur eines, nämlich Hibernate, vgl. Tabelle 6.1. Dabei ist allerdings zu beachten, dass die Projekte dieser Art, bis auf Hibernate, nur über den/die Maintainer kontaktiert wurden. So ist nicht klar zu sagen, ob die Organisation des Projektes oder die Kontaktart ausschlaggebend dafür ist, dass niemand reagierte.
In den meisten Fällen wurde diese Kontaktart nur gewählt, weil es keine andere Möglichkeit gab. So verfügten alle Projekte, die über den/die Maintainer kontaktiert wurden über keine öffentliche einsehbare Kommunikation, die die Entwicklung betrifft. Openbravo bildete hier eine Ausnahme. So lässt sich zumindest vermuten, dass sich die Projekte, die von einer Firma geleitet werden und keinen öffentlichen Austausch anbieten, weniger für meine Anfrage eignen.
Wie im Abschnitt 6.2 schon erwähnt, eigen sich Projekte vermutlich ebenfalls besser,

je offener sie sind und je mehr Informationen über sie verfügbar sind.

6.3.2 Eclipse Nutzung

Wie in Tabelle 6.2 zu sehen ist, variierte bei den Projekten, die die Programmiersprache PHP benutzen, die Verwendung von Eclipse von Benutzer zu Benutzer. DR8 formulierte dies recht treffend.

> „People don't like to be dependent on a specific platform. Everyone has their own.“[A.1.19]

Argumente dieser Art waren der dritthäufigste Grund der Ablehnung, vgl. *Abhängigkeit von einer Entwicklungsumgebung (Eclipse)* in Tabelle 6.3. Alle Aussagen dieser Kategorie, bis auf die von HI1, wurden von Entwicklern von PHP-basierten Projekten geäußert. Bei Projekten, die in Java programmiert sind, wurde Eclipse häufiger benutzt. Von den Java-Projekten, die geantwortet haben, benutzten sogar alle Eclipse und äußerten sich überwiegend positiv. Davon haben zwei Projekte (Spring,Eclipse-cs) die Eclipse benutzten gar nicht auf meine Anfrage reagiert. Wie anfangs vermutet, stellte sich heraus, dass Projekte, die Eclipse zur Entwicklung benutzen, sich besser für die Nutzung von Saros als andere.

6.3.3 Größe

Projekte die auf meine Anfrage geantwortet haben hatten ein Größe von 23 bis 66 Entwicklern, vgl. Tabelle 6.4. Zwar hatte JUnit (mit vier Entwicklern) auch geantwortet, daraus entwickelte sich aber keine Diskussion.

6.3.4 Arbeitsverteilung

Dass die Entwickler Saros nicht benutzen können, wurde am häufigsten mit einem asynchronen Arbeiten begründet, vgl. Tabelle 6.3. Diese Ablehnungen kamen aus den Projekten TYPO3, Drupal und Freenet. Das asynchrone Arbeiten in den Projekten, kann verschiedene Gründe haben. Zum einen kann dies an der geografischen

Ablehnung			29/14 (55,7%)
Technische Ablehnung	Kein sinnvoller Vorschlag	Es gibt schon so etwas	1 (1,92%)
	Mangelhaftes Produkt	Defekt	1 (1,92%)
		Abhängigkeit von einer Entwicklungsumgebung (Eclipse)	4/4[1](7,8%)
Konzeptionelle Ablehnung	Keine sinnvolle Arbeitsmethode oder Prozess	Schlechte Erfahrungen mit Pair Programming	5/3 (9,6%)
Umfeldbezogene Ablehnung	Produkt unpassend für das Projekt	Fehlende Funktionalität	1 (1,92%)
		Muss jeder selber entscheiden	1 (1,92%)
		Es wird schon so etwas benutzt	1 (1,92%)
		Zu wenige Entwickler würden das benutzen	3/2 (5,7%)
		Zu große Einarbeitungszeit	1 (1,92%)
		Produkt könnte nützlich sein, aber nicht für uns	3/3 (5,7%)
	Konzept unpassend für das Projekt	Entwickler arbeiten zu asynchron	6/5 (11,5%)
		Entwickler arbeiten in einem Büro zusammen	1 (1,92%)
		Konzept könnte nützlich sein, aber nicht für uns	1 (1,92%)

[1] Anzahl der Aussagen/Anzahl der Entwickler

Tabelle 6.3: Übersicht der Ablehnungen in den jeweiligen Kategorien

Verteilung der einzelnen Entwickler liegen. Das heißt, wenn die Entwickler in zu unterschiedlichen Zeitzonen leben wird es ihnen schwer fallen, sich in einer Saros-Sitzung zusammen zu finden. Zum anderen kann die Arbeit ungleichmäßig verteilt sein, wie bei Freenet vermutet. So erledigt nur ein Entwickler die Hauptarbeit in dem Projekt während die anderen Entwickler unregelmäßig an dem Projekt arbeiten, da sich ein Treffen in Echtzeit nicht ergibt.

Projekt	Anzahl der Entwickler	Programmiersprache	Eclipse Nutzung	Organisation
TYPO3	34	PHP	variiert	Gemeinde
Vuze	8+4[1]	Java	Ja	Firma
Freenet	10	Java	Ja	Gemeinde
Drupal	26-30	PHP	variiert	Gemeinde
WordPress	11	PHP	?	Gemeinde
JUnit	4	Java	Ja	Gemeinde
Eclipse-cs	5	Java	Ja	Gemeinde
ADempiere	66	Java	Ja	Gemeinde
Openbravo	100[2]	Java	?	Firma
OpenProj	14	Java	?	Firma
Tomcat	20	Java	?	Gemeinde
Hibernate	23	Java	?	Firma
Spring	42	Java	Ja	Gemeinde

[1] Es handelt sich dabei um 100 Mitarbeiter, wie viele davon entwickeln ist nicht bekannt.
[2] Bei Vuze wurden acht Personen als Entwickler angegeben und vier als Projektadministratoren.

Tabelle 6.4: Überblick über die kontaktierten Projekte

Für die Akzeptanz von Saros spielt also auch die Größe des Projektes eine Rolle, je mehr Entwickler an dem Projekt arbeiten, desto wahrscheinlicher ist es, das zwei Entwickler gleichzeitig an dem Projekt arbeiten.
Es können aber auch Abläufe wie z.B. der Bugs Day von ADempiere existieren, so dass die Entwickler eine festen Termin haben an dem sich viele von ihnen zusammen finden. Ein gut besuchter IRC Kanal kann hierfür ein Indiz sein.

Zusammenfassend lässt sich sagen das Projekte, die sich für ein Einführung von Saros eignen, folgende Eigenschaften haben sollten:

1. Sie sollten Eclipse benutzen, daher sind Java Projekte den PHP Projekte vorzuziehen.

2. Sie sollten gemeindebasiert sein und nicht von einer Firma geleitet werden.

3. Sie sollten sowohl über öffentliche Kommunikationsmedien als auch über einen offenen Entwicklungsprozess verfügen.

4. Die Größe der Projekte sollte ca. zwischen 20 und 70 Entwickler betragen und die Arbeit gleichmäßig auf diese verteilt sein.

5. Die Entwickler der Projekte sollten in möglichst ähnlichen Zeitzonen arbeiten oder das Projekt sollte organisatorische Abläufe haben an denen viele Entwickler zur gleichen Zeit zusammenfinden.

6. Sie sollten möglichst über einen aktiven IRC Kanal verfügen.

6.4 Saros

Dieser Abschnitt liefert Rückschlüsse auf die Forschungsfrage "*Welche Folgen ergeben sich aus dem Einsatz von Saros in OSS Projekten?*". Her wird beschrieben, was Saros leisten muss, um von OSS-Projekten benutzt werden zu können.
Der erste Eindruck von Saros bzw. meinem Vorschlag war teilweise positiv, 19,4 Prozent der gemachten Aussagen waren in der Kategorie *Interesse/sinnvoller Vorschlag*, vgl 6.5. Die meisten Aussagen dieser Kategorie waren die ersten Antworten bzw. Äußerungen auf meine Email. Entwickler die Saros getestet hatten äußerten sich alle positiv, auch wenn sie einige Defekte fanden, vgl. Abschnitt 6.4.1. Von den zehn interessierten Entwicklern haben nur drei Saros getestet und anschließend ein Feedback gegeben.

Um Vorurteilen gegen bzw. Ablehnung aufgrund von schlechten Erfahrungen mit PP entgegenzuwirken, wurde ebenfalls eine Webseite angelegt, welche die Vorteile von PP bzw. DPP für Open Source Software Projekte aufzeigt. Ich war mir natürlich dessen bewußt, dass jemand der schon schlechte Erfahrungen mit PP gemacht hat, nicht einfach zu überzeugen ist. Hingegen finden hier interessierte Nutzer Argumente und Erfahrungswerte von anderen Entwicklern.

Folgende Anwendungsmöglichkeiten von Saros konnten sich die Entwickler der kontaktierten Projekte vorstellen und lieferten so Antworten auf die Forschungsfrage "*In welchem Kontext wird Saros benutzt?*"

- Neueinsteiger anlernen und Kunden beraten.

Akzeptanz			28/14 (53,8%)
Technische Akzeptanz	Sinnvolles Produkt	Interesse/sinnvoller Vorschlag	12/10 (19,3%)
		Anwendungsvorschlag	2/2(3,8%)
		Vorschläge für Weiterentwicklung	3/3 (5,7%)
		Schon nach so etwas gesucht	2/2(3,8%)
		Gute Lösung (Saors getestet)	3/3 (5,7%)
Konzeptionelle Akzeptanz	Sinnvolle Arbeitsmethode oder Prozess	Produktivitätssteigerung	1 (1,92%)
		Sonstige	1 (1,92%)
Umfeld bezogene Akzeptanz	Produkt passend für das Projekt	Generell von Nutzen für das Projekt	1 (1,92%)
		Anwendungsvorschlag für das Projekt	3/3 (5,7%)

Tabelle 6.5: Übersicht über positive Aussagen und ihren Katagorien

- Präsentation und Verbreitung von Wissen.
- Kollaboratives Arbeiten um Patches in der Kern zu integrieren.
- Peer-Review.

Interassanterweise tauchte hier PP gar nicht als Anwendungsmöglichkeit auf. Dies kann natürlich auch daran liegen, dass diese Verwendung auf der Hand lag und die Entwickler nur zusätzliche Einsatzmöglichkeiten vorschlugen.

Durch die Sitzung mit AD2 ist auch klar geworden, dass eine Einführung in Saros unkomplizierter ist, wenn der/die Teilnehmer eingeladen werden, vgl. 5.8.5.2. Als Host hat man die Kontrolle über die Sitzung, diese Rolle kann neue Benutzer zunächst überfordern, gerade wenn sie mit mehreren Personen in einer Sitzung sind.
Durch den Follow-Mode, kann der/können die Teilnehmer sich führen lassen, ohne selbst viel einstellen zu müssen. Dann sollten nach und nach die weiteren Möglichkeiten von Saros vorgeführt werden, erst der *Multidriver* und dann *exclusive*

Driver. Ist der Benutzer so einigermaßen mit Saros vertraut kann er die Host-Rolle übernehmen.

6.4.1 Defekte

Einige Entwickler haben Saros getestet und dabei folgende Defekte in Saros gefunden. Die Tabelle 6.6 zeigt die einzelnen Defekte und ihr Status wie sie im Bug-Tracker eingetragen sind. Die Defekte, welche PDT betreffen wurden von dem Projekt TYPO3 gefunden, alle anderen von dem Projekt ADempiere. Ein

Beschreibung	**Status**
Contribution annotation doesn't work with PDT	Behoben
Saros doesn't work with PDT-Eclipse	Behoben
STATUS is irritating/ wrong while inviting	Offen
IBB[Error] while invitation	Offen
Partial Sharing into an existing Project	Offen
Partial sharing into a new Project	Offen
Sharing a very big Project	Offen
Saros doesn't work with AuthProxy	Offen

Tabelle 6.6: Gefundene Defekte und ihr Status

schwerwiegender Defekt ist dabei, dass das Partial Sharing nicht funktioniert, für ADempiere war dies der Grund, warum sie Saros nicht benutzen konnten.

6.4.2 Features

Die Feature-Anfragen wurden alle von dem Projekt ADempiere vorgeschlagen. Die Tabelle 6.7 zeigt die verschiedenen Features und ihren Status, wie sie im Feature-Tracker eingetragen sind.

Beschreibung	**Status**
Save Saros sessions for later replay	Offen
Chat is Missing	in Arbeit[1]
Support Multiple Projects in session	Offen
IBB[Error] while invitation	Offen
Remember selection when sharing a part of a project	Offen
Whiteboard	in Arbeit[2]

[1] Wird in einer Bachelorarbeit realisiert, vgl. https://www.inf.fu-berlin.de/w/SE/ThesisDPPXV
[2] Wird in einer Diplomarbeit realisiert, vgl. https://www.inf.fu-berlin.de/w/SE/ThesisDPPXVII

Tabelle 6.7: Feature Anfragen

6.4.3 Unklarheiten

Von den geäußerten Unklarheiten wurden fünf Fragen in die FAQ's[1] auf der Saros Webseite mitaufgenommen. Viele Fragen resultierten aus der Sitzung mit AD2 und werden auf der Webseite schon erläutert, er stellte diese Fragen, da ich mit ihm in einem Chat war und er sich die Webseiten noch nicht angeschaut hatte.

- Is Saros language dependent or which languages does Saros support?
- I'm working with an version control system, can i use Saros?
- How many user does Saros support?
- Does Saros run on a Mac?
- How much Bandwidth do you need to run Saros?

Wie in Tabelle 6.8 zu sehen ist, gab es kaum Häufungen der Aussagen in den verschiedenen Kategorien.

1 https://www.inf.fu-berlin.de/w/SE/DPPfaq

Unklarheiten			24
Technische Nachfragen	Installation		1 (4,16%)
	Performance	Bandbreite	2/2 (8,3%)
	Kompatibilität	Versionsverwaltungen	1 (4,16%)
		Programmiersprachen	1 (4,16%)
		Betriebssystem	1 (4,16%)
	verwendete Technologien		1 (1,92%)
	Produktvergleich	Zusammenhang mit anderen Produkten	3/3 (12,5%)
	Sonstige		7/2 (29,1%)
Konzeptionelle Nachfragen	Verständigung		1 (4,16%))
	Effektivität		1 (4,16%)
	Sonstige		5/1(20,8%)

Tabelle 6.8: Übersicht über die kategorisierten Nachfragen

6.5 Bewertung des Untersuchungsplanes

Das methodische Vorgehen dieser Arbeit hat sich im nachhinein als sinnvoll herausgestellt. Mein Ansatz war flexibel genug, um ihn den Projekten anzupassen und diese dennoch vergleichbar zu halten. Das Kategoriensystem strukturierte die Auswertung erheblich. Aus fast 100 Seiten Textmaterial konnten damit alle von mir als wichtig erachteten Zitate gebündelt und untersucht werden.
Das iterative Vorgehen bei den Projekten stellte sich ebenfalls als produktiv heraus, da so nicht nur der Forschungsablauf strukturiert wurde, sondern vor allem Erkenntnisse in die nächste Iteration einfließen konnten. Hier stellte sich allerdings ein methodisches Problem, da dieses Vorgehen nur bei Projekten funktioniert, die Saros auch testen und Feedback darüber geben.
Im abschließenden Kapitel dieser Arbeit wird gezeigt werden, was eine weitergehende Forschung vor allem bezüglich der Einführung von Saros im Open Source Bereich beachten sollte.

7 Ausblick

In dieser Arbeit wurden Kriterien von Projekten erstellt, die sich für eine Anfrage und Nutzung von Saros eignen. Darauf aufbauend sollte es zukünftigen Arbeiten die Kontaktaufnahme und Einführung von Saros erleichtern. Zuvor sollten aber die noch offenen Defekte in Saros repariert werden, um über ein breiteres Anwendungsfeld zu verfügen. Hierbei ist das Partielle Sharing besonders wichtig, da dieses wahrscheinlich von allen Projekten mit großer Datenmenge benötigt wird. Dadurch könnt auch das vorhandene Interesse von ADempiere genutzt werden. Damit könnte in folgenden Arbeiten der dann kommende Bug Day forciert werden.

Ich denke, es ist für eine erfolgreiche Kontaktaufnahme sinnvoll die Projekte intensiver zu analysieren. Es gilt Anwendungsmöglichkeiten von Saros zu finden, damit diese dem Projekt gezielt vorgeschlagen werden können. Zudem könnte es ebenfalls nützlich sein, die Projekte im Vorfeld dahingehend zu untersuchen, was für eine Arbeitsverteilung gegeben ist. Zum Beispiel über die Commits in der Versionsverwaltung kann herausgefunden werden, wie viele Personen an dem Projekt arbeiten. Entwickelt nur eine Person daran, ist dies wahrscheinlich nicht gut geeignet. So sollte in zukünftigen Arbeiten der Fokus auf die Qualität und die die Quantität der Projekte gesetzt werden.

Da sich in dieser Arbeit gezeigt hat, dass es aus verschiedenen Gründen schwer möglich ist, Saros in ein OSS-Projekt zu integrieren, kann nun hinterfragt werden, ob das Hauptaugenmerk in zukünftigen Arbeiten überhaupt darauf gelegt werden sollte. Diese Schwierigkeit liegt zum Teil daran, dass Saros keine Software zur Projektorganisation ist und somit zunächst keinen zentralen Prozess darstellt, sondern von den einzelnen Entwickler benutzt werden muss. So ist der Entwickler

unmittelbar abhängig von Eclipse. Er muss ein Interesse, aber auch die Möglichkeit an kollaborativer Zusammenarbeit haben. Der Einsatz von Saros kann so nicht zentral entschieden werden, sondern nur durch einzelne Vorreiter in dem Projekt erreicht werden. Diese Vorreiter zu motivieren sollte meiner Meinung nach Aufgabe des Forschers sein. Wird Saros dann von einigen Entwicklern innerhalb eine Projektes benutzt, lassen sich die Froschungsfragen wie Saros und in welchem Kontext benutzt wird beantworten.

Um den potentiellen Benutzern auf der Webseite die verschiedenen Möglichkeiten von Saros näher zu bringen, könnte es sinnvoll sein, Videos zu erstellen, die zeigen, wie typische Arbeitsabläufe in OSS-Projekten mit Saros gelöst werden können. Zum Beispiel können zwei Entwickler einen Patch mit der Hilfe von Saros sichten und in ein Projekt integrieren oder jemanden, der neu in einem Projekt ist eine Einführung, geben.

A Anhang

A.1 Tagebuch

A.1.1 Samstag 15.08.09

13:40 Uhr

Kontakt mit TYPO3 Projekt aufgenommen (typo3-project-5_0-general, typo3-dev):

> Dear Typo3 Team,
>
> My name is Eike Starkmann and I'm working with Typo3 since 3 years for a small Internet agency called undkonsorten.com. During this time I wrote a couple of extensions. At the moment I'm working as part of the Saros Team at the Freie University in Berlin.
>
> Saros is an Eclipse plugin for collaborative text editing and distributed pair programming, i.e. it allows two or more developers to work together in real-time on the same files. It is similar to Gobby, SubEthaEdit or Google Docs but focuses on programming in Eclipse.
>
> In my opinion both the Typo3 core team (v.4/v.5) and the extension community can benefit from Saros because I think it brings many advantages to Open Source Software development:
>
> * Distributed Pair Programming is like a live peer review. This should help with finding good design, get rid of bugs, increase readability, etc.
>
> * Transferring knowledge should be easier to do when more than one person look at and work with the same code. This should also help to give new developers an introduction to the code.

* In contrast to screen sharing, Saros only shares your actions inside of Eclipse with regards to the project you are both working on (think privacy) and you are still independent to explore the project on your own.

Saros can be useful in the following contexts:

* Working on complicated problems in the code * Performing code reviews * Debugging * Code presentation * Code sprints * Introducing new developers to the project * ...

It is my master thesis to figure out whether Saros is useful when developing Free/Open Source Software.

What do you think? Could you and Typo3 benefit from doing pair programming using Saros?

If you are interested in Saros but still curious about how it works please visit our website or feel free to contact me.

I hope you will find Saros useful and give me feedback.

Kind regards, Eike Starkmann

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin

15:48 Uhr

1. Reaktion auf typo3-project-5_0-general von TY1:

```
> Saros is an Eclipse plugin for collaborative text editing and
> distributed pair programming, i.e. it allows two or more
> developers to work together in real-time on the same files. It
> is similar to Gobby, SubEthaEdit or Google Docs but focuses on
> programming in Eclipse.
That sounds quite cool  :-)
```

```
> What do you think? Could you and Typo3 benefit from doing pair
> programming using Saros?
Yep, I'll give it a try soon after my holidays!

I have two questions left:
How does it relate to: http://wiki.eclipse.org/DocShare_Plugin ? I
actually never really managed to get DocShare working, but I
thought this would be "The Eclipse Way"?

On https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugins , it
seems that PHP is not supported right now? I guess most of us
really need PDT in order to try it out properly  :-)
```

17:07 Uhr

1. Reaktion auf typo3-project-5_0-general von TY1: Ich antworte ihm:

```
Hi TY1,
thanks for your quick answer.

>> Saros is an Eclipse plugin for collaborative text editing and
>> distributed pair programming, i.e. it allows two or more
>> developers to work together in real-time on the same files. It
>> is similar to Gobby, SubEthaEdit or Google Docs but focuses on
>> programming inEclipse.
> That sounds quite cool :-)
>
>> What do you think? Could you and Typo3 benefit from doing pair
>> programming using Saros?
> Yep, I'll give it a try soon after my holidays!
Yes, that would be nice.
>
> I have two questions left:
> How does it relate to: http://wiki.eclipse.org/DocShare_Plugin ?
> I actually never really managed to get DocShare working, but I
> thought this would be "The Eclipse Way"?
It is not related to DocShare. DocShare is part of the Eclipse
Communication Frameworks, so yes this is the "Eclipse Way".
DocShare has widespread goals but Saros is more focused on doing
```

```
pair programming so this not really the same.
>
> On https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugins , it
> seems that PHP is not supported right now? I guess most of us
> really need PDT in order to try it out properly :-)
I supposed that. ;-) We have written some patches to get PDT and
DEV3 to work. These Patches will be part of the next release on
Friday 21.08. So if you check out Saros on Friday/Saturday PDT and
DEV3 should work. Aptana and phpeclipse are working right now out
of the box, but i don't know if some of you are using it.

Have an nice holiday, Eike
```

22:03 Uhr

2. Reaktion auf typo3-project-5_0-general von TY2:

```
I would be happy to try it with somebody as soon as we have all
PHP support we need, so if somebody is going to try it just sent
me a mail.

I think it can be a quite useful tool, even if I admit, that in
such an opensource project the working times will be very, very
different so that it might be difficult to for teams to find
together.
```

Positive Resonanz, benötigt aber PHP um mit Saros arbeiten zu können. Hat Bedenken wegen der Arbeitszeiten. Ich antworte ihm:

```
Hi TY2
> I would be happy to try it with somebody as soon as we have all
> PHP support we need, so if somebody is going to try it just sent
> me a mail.
PDT and DEV3 should work after the next release on Friday 21.08 ;-)
Since this is the first time I present Saros to an Open Source
Project, I'm not sure how to organize people interested in Saros
to work with it.
I hope this can be done in an selforganized way, meaning that the
teams can find together alone. But I also can offer help, if
```

```
needed.
>
> I think it can be a quite useful tool, even if I admit, that in
> such an open-source project the working times will be very, very
> different so that it might be difficult to for teams to find
> together.
Yes, I was thinking about that too and actually that is also
something, I would like to find out. As mentioned above, I'm not
sure how this "find together" can be done, or if it comes
automatically.
Feel free to send some suggestions ;-)

Kind regards, Eike
```

A.1.2 Sonntag 16.08.09

18:47 Uhr

3. Reaktion auf typo3-project-5_0-general von TY3:

```
Is it something like Bespin?
https://bespin.mozilla.com/
```

Fritz Thomas veröffentlicht einen Artikel auf seinem Blog, welcher Bezug nimmt auf meine Email an TYPO3: http://fritzthomas.com/overall/467-saros-distributed-pair-programming-plugin-for-eclipse/

Ich antworte ihm:

```
Hi Dan,

> Is it something like Bespin?
> https://bespin.mozilla.com/
Yes and no. The collaboration editing of code is the same idea,
but Saros is not running in an web browser, its an Eclipse Plugin.
At least the collaboration editing of code is the only similarity
to Saros, but we also support concurrent editing of up to 5 people
at the same time in the same file, which isn't possible in Bespin
as far as i know.

Kind regards, Eike
```

A.1.3 Dienstag 18.08.09

22:03 Uhr

4. Reaktion auf typo3-project-5_0-general von TY4:

```
Sounds like a very cool project!

On 17.08.09 17:07, Eike Starkmann wrote:
>> On https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugins , it
>> seems that PHP is not supported right now? I guess most of us
>> really need PDT in order to try it out properly  :-)
> I supposed that.  ;-)  We have written some patches to get PDT
> and DEV3 to work. These Patches will be part of the next release
> on Friday 21.08.
Ok, I'll give it a try after that.
```

Ich antworte ihm:

```
Hi TY4,

 > Sounds like a very cool project!
Thank you!

 > Ok, I'll give it a try after that.
Thank you, and don't forget to give me feedback ;-)


Kind redards, Eike
```

A.1.4 Mittwoch 19.08.09

Kontakt mit Projekt DEV3 aufgenommen und eine automatische Antwortmail bekommen, dass ein Entwickler bis zum 07.09.09 im Urlaub ist:

```
Dear DEV3 Team,

My name is Eike Starkmann and I'm working with Typo3 since 3 years
 for as mall Internet agency called undkonsorten.com. During this
time I wrote a couple of extensions. At the moment I'm working as
```

```
part of the Saros Team at the Freie University in Berlin.

Saros is an Eclipse plugin for collaborative text editing and
distributed pair programming, i.e. it allows two or more
developers to work together in real-time on the same files. It is
similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.

As announced on the Typo3 mailinglist my opinion is that both the
Typo3 core team (v.4/v.5) and the extension
community can benefit from Saros because I think it brings many
advantages to Open Source Software development:

* Distributed Pair Programming is like a live peer review. This should
help with finding good design, get rid of bugs, increase readability,
etc.

* Transferring knowledge should be easier to do when more than one
person look at and work with the same code. This should also help to
give new developers an introduction to the code.

* In contrast to screen sharing, Saros only shares your actions inside
of Eclipse with regards to the project you are both working on (think
privacy) and you are still independent to explore the project on your
own.

Saros can be useful in the following contexts:

* Working on complicated problems in the code
* Performing code reviews
* Debugging
* Code presentation
* Code sprints
* Introducing new developers to the project
* ...

It is my master thesis to figure out whether Saros is useful when
developing Free/Open Source Software.
```

```
We will support DEV3 after the next Release on 21.08.09 for Typo3,
but perhaps the DEV3 Project can also benefit from using Saros for
developing the Eclipse Plugin. What do you think?

If you are interested in Saros but still curious about how it
works please visit our website or feel free to contact me.

I hope you will find Saros useful and give me feedback.

Kind regards, Eike Starkmann
```

A.1.5 Samstag 20.08.09

Max Horvath veröffentlicht eine Artikel auf seinem Blog, bezugnehmend auf meine Email an TYPO3-5 Mailing-Liste: http://www.maxhorvath.com/2009/08/saros-distributed-pair-programming-plugin-for-eclipse.html

A.1.6 Freitag 26.08.09

08:39 Uhr

Nachfrage von TY1:

```
Sorry, I still need some help: Where on
http://www.eclipse.org/pdt/downloads/
can I download the all-in-one build for that friday?
```

A.1.7 Montag 31.08.09

14:16 Uhr

Erstes Testfeedback von TY1:

```
Hey Eike,

We just tried version 9.8.21.r1660 with PDT - this is the most
recent one, right?
```

```
It seemed to work, and it looks like a really nice approach!
There is one thing which didn't seem to work (or maybe we
configured it wrongly):
Highlighting what the other person typed.

We saw the cursor of the other person, but I thought the typed
text by the person is highlighted as well, no?

Looking forward to try it again  :)

Oh, and we used in on jabber.ccc.de - although it was not supposed
to work there.

Greets,
```

Bis jetzt keinerlei Mailverkehr auf DEV3 und keine Resonanz auf typo3-dev.

A.1.8 Dienstag 03.09.09

Ich antworte TY1 und versuche Informationen über die durchgeführte Sitzung zu bekommen:

```
Hi TY1,
thank you for trying it. I think you are the first one of Typo3 ;-)
> We just tried version 9.8.21.r1660 with PDT - this is the most
> recent one, right?
Yes this is the most recent one.
>
> It seemed to work, and it looks like a really nice approach!
Thank you.
> There is one thing which didn't seem to work (or maybe we
> configured it wrongly):
> Highlighting what the other person typed.
>
> We saw the cursor of the other person, but I thought the typed
>  text by the person is highlighted as well, no?
Hmm, i tested it today and you are right. This does not happen due
wrong configuration. I submitted this as a bug on our tracker:
```

```
https://sourceforge.net/tracker/?func=detail&aid=2849925
&group_id=167540&atid=843359
>
> Looking forward to try it again :)
Stay with it! :-)

By the way how many people were in the session and how did you
find together?

Greets, Eike
```

A.1.9 Mittwoch 04.09.09

13:18 Uhr

Email verschickt an Mailingliste TYPO3-project-4-3. Diese Mailingliste ist nur für das aktuelle Release von TYPO3 4.3 zuständig.

```
Dear Typo3 4.3 Team,

i already sent this email to TYPO3-dev and TYPO3-project-5_0 and
got some good response from TYPO3-project-5_0-general but on
TYPO3-dev nobody answered so I'm trying this list. Sorry if some
of you will get this email twice, i don't want to spam.
Here is my concern:

My name is Eike Starkmann and I'm working with Typo3 since 3 years
for asmall Internet agency called undkonsorten.com. During this
time I wrote a couple of extensions. At the moment I'm working as
part of the Saros Team at the Freie University in Berlin.

Saros is an Eclipse plugin for collaborative text editing and
distributed pair programming, i.e. it allows two or more
developers to work together in real-time on the same files. It is
similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.

In my opinion both  the Typo3 core team (v.4/v.5)  and the
extension community can benefit from Saros because I think it
```

```
brings many advantages to Open Source Software development:

* Distributed Pair Programming is like a live peer review. This
should help with finding good design, get rid of bugs, increase
readability, etc.

* Transferring knowledge should be easier to do when more than one
person look at and work with the same code. This should also help
to give new developers an introduction to the code.

* In contrast to screen sharing, Saros only shares your actions
inside of Eclipse with regards to the project you are both working
on (think privacy) and you are still independent to explore the
project on your own.

Saros can be useful in the following contexts:

* Working on complicated problems in the code
* Performing code reviews
* Debugging
* Code presentation
* Code sprints
* Introducing new developers to the project
* ...

It is my master thesis to figure out whether Saros is useful when
developing Free/Open Source Software.

What do you think? Could you and Typo3 benefit from doing pair
programming using Saros?

If you are interested in Saros but still curious about how it
works please visit our website or feel free to contact me.

I hope you will find Saros useful and give me feedback.

Kind regards, Eike Starkmann
```

Neue Projekte ausgesucht:

- https://sourceforge.net/projects/jboss/
  Im Repository liegt eine .projekt Datei.
- https://sourceforge.net/projects/hibernate/
  Arbeiten vermutlich mit Eclipse
- https://sourceforge.net/projects/eclipse-cs/
  Eclipse Plugin
- ttps://sourceforge.net/projects/ganttproject/
  Im Repository liegt eine .projekt Datei.
- https://sourceforge.net/projects/springframework/
  Arbeiten vermutlich mit Eclipse

21.29 Uhr

Antwort erhalten von TY6:

```
I think there are people who can do it but it is rare. I tried
pair programming several times but found that two developers
must be really on the same level. If one of them is higher,
than it becomes master/slave programming, which takes more time
than a simple "master" programming.Therefore I do not really
believe into pair programming. The concept sounds nice but in
practice I never saw satisfying result. May be I am
simply unlucky.

I believe two programmers can work on the same module if the
clearly separate tasks and agree on API that they will provide to
each other.

I do not say that my beliefs are the ony correct ones. I am sure
other people may have different experience.
```

A.1.10 Donnerstag 05.09.09

18:30 Uhr

TY7 antwortet auf TY6:

```
> I do not say that my beliefs are the ony correct ones. I am sure
> other people may have different experience.
... for example, me.  :-)

I've found "real" pair programming (two coders sitting in front of
the same physical computer) a lot of fun an really productive
because ...

1. the coding was generally faster: we could help each other, move
the keyboard over when someone was stuck, and the
"social pressure" kept our focus on the task

2. it saved us all the time-consuming code review/new version/code
review etc. cycles

3. it helps knowledge spreading between the coders

I haven't tried distributes pair programming (DPP) yet, though. I
wonder how the communication in DPP works if you cannot talk to
each other.
```

A.1.11 Dienstag 08.09.09

13:46 Uhr

Ich antworte TY6:

```
Hi TY6,
thanks for your answer.
TY6 wrote:
>
> Eike Starkmann wrote:
>> What do you think? Could you and Typo3 benefit from doing pair
>> programming using Saros?
```

```
>
> I think there are people who can do it but it is rare. I tried
> pair programming several times but found that two developers
> must be really on the same level. If one of them is higher, than
> it becomes master/slave programming, which takes more time than
> a simple "master" programming.
>
> Therefore I do not really believe into pair programming. The
> concept sounds nice but in practice I neve saw satisfying result.
> May be I am simply unlucky.
>
> I believe two programmers can work on the same module if the
> clearly separate tasks and agree on API that they will provide
> to each other.
>
> I do not say that my beliefs are the ony correct ones. I am sure
> other people may have different experience.
>
That is one point in would like to find out in my research. Is
this your opinion in general about pair programming or just in the
context of Open Source Software Development? Did you ever try pair
programming ordistributed pair programming in Open Source Software
development?

Greets, Eike
```

14:03 Uhr

Ich antworte auf die Mail von TY7 (A.1.10):

```
>> I do not say that my beliefs are the ony correct ones. I am
>> sure other people may have different experience.
>
> ... for example, me. :-)
And me ;-)
>
> I've found "real" pair programming (two coders sitting in front
> of the same physical computer) a lot of fun an really productive
> because ...
```

```
>
> 1. the coding was generally faster: we could help each other,
> move the keyboard over when someone was stuck, and the "social
> pressure"  kept ourfocus on the task
>
> 2. it saved us all the time-consuming code review/new
> version/code review etc. cycles
>
> 3. it helps knowledge spreading between the coders
>
I totally agree with this the points and have made the same
experienceswith pair programming, although if there is a big
difference of knowledge its sometimes hard to do. As Dmitry
mentioned.

> I haven't tried distributes pair programming (DPP) yet, though.
> I wonder how the communication in DPP works if you cannot talk
> to each other.
>
I have experiences with pair programming and DPP as well. One big
advantage in DPP against PP is that both (or even more) can type
in the same document. I know people using Saros for DPP and
sitting in the same room, just for this concurrent editing.
Normally we suggest to use VoIP for communication. Mumble is the
recommended but tools like Skype can do as well.
I think this would be the normal case in Open Source Projects,
since you don't have the possibility to sit in the same room,
right?


Greets Eike
```

14:58 Uhr

TY6 antwortet auf meine Mail (A.1.11):

```
Eike Starkmann wrote:
> That is one point in would like to find out in my research. Is
> this your opinion in general about pair programming or just in
```

```
> the context of Open Source Software Development? Did you ever
> try pair programming or distributed pair programming in Open
> Source Software development?

It was a very large commercial product. But I do not see any
difference.
Developers are developers everywhere. To me pair programming is
more a distraction than creativity. I am ok to ~discuss~ how to
create something but when it comes to coding I found that one
person coding produces better results. But I also prefer to think
first, discuss next and code last, not all at the same time  :)
```

A.1.12 Mittwoch 23.09.09

16:18 Uhr

Nach 20 Tagen antwortet TY4 auf meine Nachfrage an TY1 (A.1.8):

```
We were two, if TY1 was talking about the session he had with me.
:)
We simply logged in with our Jabber accounts and could see the
other in the peer list.

Regards,
TY4

PS: Since Netbeans is *awesome* when it comes to supporting PHP
5.3, I am no longer using Eclipse/PDT. At least for now...
```

A.1.13 Donnerstag 24.09.09

09:43 Uhr

TY5 befragt TY5

```
Just as an aside, have you seen any realtime editing plugins for
Netbeans? Something similar to Gobby? I've not come across
anything, but find them rather useful indeed!
```

```
Regards,
Denyer
```

15:55 Uhr

Ich antworte ihm:

```
>>Just as an aside, have you seen any realtime editing plugins for
>>Netbeans? Something similar to Gobby? I've not come across
>>anything, but find them rather useful indeed!
Sorry, i never tried Netbeans.
Are there many Netbeans users in TYPO3?

Greets, Eike
```

18:09 Uhr

TY5 antwortet mir:

```
If there are not, they should definitely give the latest version a
try, it really is quite superb for a free IDE, the sql server
integration is great if you're working with databases (not a
problemfor anyone using an ORM, but you can't always win!) and I
managed to get it working with XDebug without too much trouble,
which means it can't be hard  :)   The improvements they've got
lined up for V7 are impressive too.

Regards,
TY5
```

A.1.14 Dienstag 29.09.09

16:58 Uhr

Email verschickt an die 4 Projekt Admins von Azureus:

```
Dear ArronM,

My name is Eike Starkmann and I'm  working as part of the Saros
Team at the Freie University in Berlin.
```

```
Saros is an Eclipse plugin for collaborative text editing and
distributed pair programming, i.e. it allows two or more
developers to work together in real-time on the same files. It is
similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.

In my opinion your  can benefit from Saros because I think it
brings many advantages to Open Source Software development:

* Distributed Pair Programming is like a live peer review. This
should help with finding good design, get rid of bugs, increase
readability, etc.

* Transferring knowledge should be easier to do when more than one
person look at and work with the same code. This should also help
to give new developers an introduction to the code.

* In contrast to screen sharing, Saros only shares your actions
inside of Eclipse with regards to the project you are both working
on (think privacy) and you are still independent to explore the
project on your own.

Saros can be useful in the following contexts:

* Working on complicated problems in the code
* Performing code reviews
* Debugging
* Code presentation
* Code sprints
* Introducing new developers to the project
* ...

It is my master thesis to figure out whether Saros is useful when
developing Free/Open Source Software.

What do you think? Could you and Azureus benefit from doing pair
programming using Saros? If you think so I would like to join our
```

```
project and help you with using Saros.

If you are interested in Saros but still curious about how it
works please visit our website or feel free to contact me.

I hope you will find Saros useful and give me feedback.

Kind regards, Eike Starkmann

P.S.: Sorry for contacting all of you but i didn't know whom of
you to ask.

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

17:10 Uhr

Auf der freenet liste eingetragen[1]. Diese Liste ist für generelle technische Diskussionen.

A.1.15 Mittwoch 30.09.09

13:00 Uhr

Bis jetzt ist keine Email über die Tech-Liste von freenet gekommen, zur Beobachtung habe ich mich noch auf die Dev-Liste eingetragen.

1 http://emu.freenetproject.org/cgi-bin/mailman/listinfo/tech/

A.1.16 Montag 05.10.09

15:15 Uhr

Über die Dev-Liste (freenet) sind die ersten Emails gekommen. Meine Anfrage scheint in den Kontext der Mailingliste zu passen, daher habe ich eine Anfrage abgeschickt:

```
Dear freenet project,

My name is Eike Starkmann and I'm working as part of the Saros
Team at the Freie University in Berlin.

Saros is an Eclipse plugin for collaborative text editing and
distributed pair programming, i.e. it allows two or more
developers to work together in real-time on the same files. It is
similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.

In my opinion your  can benefit from Saros because I think it
brings many advantages to Open Source Software development:

* Distributed Pair Programming is like a live peer review. This
should help with finding good design, get rid of bugs, increase
readability, etc.

* Transferring knowledge should be easier to do when more than one
person look at and work with the same code. This should also help
to give new developers an introduction to the code.

* In contrast to screen sharing, Saros only shares your actions
inside of Eclipse with regards to the project you are both working
on (think privacy) and you are still independent to explore the
project on your own.

Saros can be useful in the following contexts:

* Working on complicated problems in the code
* Performing code reviews
```

```
* Debugging
* Code presentation
* Code sprints
* Introducing new developers to the project
* ...

It is my master thesis to figure out whether Saros is useful when
developing Free/Open Source Software.

What do you think? Could you and freenet benefit from doing pair
programming using Saros? If you think so I would like to join our
project and help you with using Saros.

If you are interested in Saros but still curious about how it
works please visit our website or feel free to contact me.

I hope you will find Saros useful and give me feedback.

Kind regards, Eike Starkmann

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

Immer noch keine Reaktion von den Vuze Admins.

A.1.17 Freitag 09.10.09

12:48 Uhr

Auf development@drupal.org eingetragen.

16:00 Uhr

Nach genauerer Betrachtung habe ich das Projekt JBoss verworfen, da sie als eine Firma organisiert sind.

A.1.18 Mittwoch 14.10.09

13:33 Uhr

Email an Drupal verschickt:

```
Dear Drupal Team,

My name is Eike Starkmann and I'm working working as part of the
Saros Team at the Freie University in Berlin.

Saros is an Eclipse plugin for collaborative text editing and
distributed pair programming, i.e. it allows two or more
developers to work together in real-time on the same files. It is
similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.

It is my master thesis to figure out whether Saros is useful when
developing Free/Open Source Software. I already was in contact
with to other projects, for example Typo3 and got some good
response.

In my opinion Drupal can benefit from Saros because I think it
brings many advantages to Open Source Software development:

* Distributed Pair Programming is like a live peer review. This
should help with finding good design, get rid of bugs, increase
readability, etc.

* Transferring knowledge should be easier to do when more than one
person look at and work with the same code. This should also help
to give new developers an introduction to the code.

* In contrast to screen sharing, Saros only shares your actions
inside of Eclipse with regards to the project you are both working
on (think privacy) and you are still independent to explore
the project on your own.

Saros can be useful in the following contexts:
```

```
* Working on complicated problems in the code
* Performing code reviews
* Debugging
* Code presentation
* Code sprints
* Introducing new developers to the project
* ...

What do you think? Could you and Drupal benefit from doing pair
programming using Saros?

If you are interested in Saros but still curious about how it
works please visit our website or feel free to contact me.

I hope you will find Saros useful and give me feedback.

Kind regards, Eike Starkmann

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

14:00 Uhr

2. Email an freenet:

```
Hi,

Don't you think that collaborative development can work in your
project? Does anyone have some experience with this?

Greets, Eike
```

14:16 Uhr

2. Iteration TYPO3: Neue Release-Email verschickt an typo3-project-5_0-general:

```
Hi,
we finally did our new release it took more time than expected but now
it is ready.
> There is one thing which didn't seem to work (or maybe we
> configured it wrongly):
> Highlighting what the other person ttped.
>
> We saw the cursor of the other person, but I thought the typed
> text by the person is highlighted as well, no?
>
We fixed that bug. And PDT now works as it should work.
We also put some new nice features in it:
https://sourceforge.net/news/?group_id=167540

Have fun checking it out  ;-)

Greets, Eike
```

14:20 Uhr

Release Ankündigung auf twitter und identi.ca gepostet[1].

14:44 Uhr

Email an wp-hackers (Wordpress) verschickt:

```
Dear Wordpress Team,

My name is Eike Starkmann and I'm working working as part of the
Saros Team at the Freie University in Berlin.

Saros is an Eclipse plugin for collaborative text editing and
distributed pair programming, i.e. it allows two or more
developers to work together in real-time on the same files. It is
similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.
```

1 http://twitter.com/saros_plugin

```
It is my master thesis to figure out whether Saros is useful when
developing Free/Open Source Software. I already was in contact
with  to other projects, for example Typo3 and got some good
response.

In my opinion Wordpress can benefit from Saros because I think it
brings many advantages to Open Source Software development:

* Distributed Pair Programming is like a live peer review. This
should help with finding good design, get rid of bugs, increase
readability, etc.

* Transferring knowledge should be easier to do when more than one
person look at and work with the same code. This should also help
to give new developers an introduction to the code.

* In contrast to screen sharing, Saros only shares your actions
inside of Eclipse with regards to the project you are both working
on (think privacy) and you are still independent to explore the
project on your own.

Saros can be useful in the following contexts:

* Working on complicated problems in the code
* Performing code reviews
* Debugging
* Code presentation
* Code sprints
* Introducing new developers to the project
* ...

What do you think? Could you and Wordpress benefit from doing pair
programming using Saros? If you think so I would like to join our
project and help you with using Saros.

If you are interested in Saros but still curious about how it
works please visit our website or feel free to contact me.
```

```
I hope you will find Saros useful and give me feedback.

Kind regards, Eike Starkmann

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

14:59 Uhr

Email an security@typo3.org:

```
Hi,

My name is Eike Starkmann and I'm working with Typo3 since 3 years
for a small Internet agency called undkonsorten.com. During this
time I wrote a couple of extensions. At the moment I'm working as
part of the Saros Team at the Freie University in Berlin.

Saros is an Eclipse plugin for collaborative text editing and
distributed pair programming, i.e. it allows two or more
developers to work together in real-time on the same files. It is
similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.

It is my master thesis to figure out whether Saros is useful when
developing Free/Open Source Software. I already sent this email to
other Typo3 mailing lists and got some good response.

As Security Team your are doing a lot of reviews, as I read the
pages Extension Security Policy and Incident Handling, I can see
at least two possible applications for Saros:
1. Making a review together with an extension developer, after he
fixed his extension.
2. Making an internal review with the help of Saros.
```

```
I also would offer you to make a test review of one TYPO3 extension
to get to know Saros.

What do you think? Could you benefit from doing distributed pair
programming using Saros? If you think so I would like to join your
project and help you with using Saros.
If you are interested in Saros but still curious about how it
works please visit our website or feel free to contact me.

Kind regards, Eike Starkmann

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

15:02 Uhr

2. Iteration TYPO3, neue Release-Email verschickt an typo3-dev:

```
Hi,
since i got some good response from other typo3 mailinglist,
i try it here again:
Saros is now supporting PDT, Aptana and DEV3(more or less).
It would be great if people check it out and tell me if
collaborative work can function in OSS development.
If you find it useful i would like to join the project and
support you with using Saros.
Greets, Eike
```

15.12 Uhr

Erste Antwort von drupal-Mailingliste von DR1:

```
Sounds very interesting  :)
What's the overhead like? Would it be smaller than that of VNC, as
you are essentially only sending text back and forth?
```

```
When both parties save an edited file, does it get stored in the
local workspace as usual? Wouldn't it cause conflicts if you are
using version control?
```

15:42 Uhr

DR2 antwortet auf DR1:

```
>Sounds very interesting :)
>What's the overhead like? Would it be smaller than that of VNC, as
>you are essentially only sending text back and forth?
>
>When both parties save an edited file, does it get stored in the
>local workspace as usual? Wouldn't it cause conflicts if you are
>using version control?
>

Not to rain on someone else's parade but you can do the same in
Emacs:

http://technomancy.us/129

Has you can read commiting with a version control system can be
done only by one person at a time, otherwise you get conflicts.
```

15:53 Uhr

3. Reaktion auf drupal-Mailingliste von DR3:

```
This sounds very similar to
http://codesurgeonblog.com/2008/06/
cola-real-time-shared-editing.html
which was demonstrated last year. How is Saros related to that
effort? A quick search didn't reveal much.
```

16:00 Uhr

Ich antworte DR1:

```
>Sounds very interesting :)
Thanks, would be great if it can improve your work.
>What's the overhead like? Would it be smaller than that of VNC,
>as you are essentially only sending text back and forth?
It is much smaller than VNC, because only the diff of the text is
send via XMPP. So only (compressed)text is send.

>When both parties save an edited file, does it get stored in the
>local workspace as usual? Wouldn't it cause conflicts if you are
> usingversion control?
Yes, the files are stored in the local workspace and are kept sync
between users. The normal way you work is that the host shares
his/the project with others. But everybody can also
just check out the project via svn/git/etc. before they start a
session. In don't know if i got your question right. Derived files
aren't send via Saros, e.g. .svn isn't synchronized.

Greets, Eike
```

16:06 Uhr

Ich antworte DR2:

```
>Not to rain on someone else's parade but you can do the same in
> Emacs:

>http://technomancy.us/129
That's true, but i'm not using emacs ;-)
>Has you can read commiting with a version control system can be
>done only by one person at a time, otherwise you get conflicts.

This isn't a problem with Saros because after a session everybody
has the same project. So i makes no difference who is committing.

Greets, Eike
```

16:16 Uhr

Ich antworte DR3:

```
>This sounds very similar to
>http://codesurgeonblog.com/2008/06/
>cola->real-time-shared-editing.html
>which was demonstrated last year. How is Saros related to that
>effort?  A quick search didn't reveal much.
It is similar, but Saros is not related in any way to Cola. Saros
is an own project done by Freie University Berlin.
Have you ever worked with Cola?
```

17:10 Uhr

1.Reaktion von WP1:

```
Speaking as somebody who's been coding professionally for 10 years,
I couldn't let this go by...

I've had to do this sort of thing before (not distributed, but
pair programming). It is, by far, the worst idea ever conceived by
man(agers). I would rip my own arm off and beat myself to death
with it before I'd ever do this sort of thing again.

The theory of pair programming is that you have two developers
working on the same piece of code, able to notice and correct
mistakes.

The reality of pair programming is that one guy is better than the
other guy, and spends half his time explaining the problem and
solution to the other guy, instead of simply coding the damn thing.
The lesser of the two programmers only drags the other guy down to
his level, and wastes both of their time.

The only way that pair programming works is if both programmers
are a) equally skilled and b) not particularly skilled. In other
words, for newbs fresh out of college, it works fine. For a
seasoned programmer, the very notion is a valid reason to look for
a job elsewhere.

In general, it's a horrible practice, and anybody actively using
it is an idiot. IMO, of course.
```

```
Also, peer review is highly overrated as well, for other reasons.
People who actively write good code don't need it, and people who
don't write good code shouldn't be put in the position of writing
code in the first place.
```

18:08 Uhr

4.Reaktion von DR4:

```
Just one point to clarify, there is no "Drupal Team" that could
decide en masse to use a tool like this.  Drupal is so distributed
that we all have our own development workflows using a variety of
tools, often in our basements rather than our offices (although
there, too).  So "Could use and Drupal benefit" is really the
wrong question to ask.

That said, I'm sure such a tool would prove useful to certain
developers if they choose to use it.

Is it language-dependent?  Vis, most Eclipse devs work on Java, not
PHP, so I am always wary of tools that may end up being centric to
one particular language's development idiosyncrasies.
```

18:17 Uhr

DR5 antwortet auf DR4:

```
I was thinking along the same lines.  Most of the Drupal
development that I've seen is either:

A) Written by one person, then reviewed / improved asynchronously
in the issue tracker rather than working at the same time
B) Written by a team of people for a project in-house, in which
case it would be the same as basically any other in-house PHP
development

I use Eclipse for all of my PHP development (and know some others
who do as well), but I probably won't ever use something like
```

```
Saros unless I got a tele-working position somewhere.

Some other module / core contributors might find it useful for
those tasks, but it doesn't seem like a common need.
```

18:36 Uhr

DR6 antwortet auf DR5:

```
I was just looking for something like this! I'll be using this
very soon on another project of mine =D

As far as Drupal goes, I think that something like this would be
particularly useful for core in a time like...oh,say....now. With
code freeze ~38 hours away, efficiency is key to getting the
patchesfrom the exception list into core. If more than one person
is working on a patch, it really makes sense to use something like
this and have everyone looking at the same code at the same time
so that it's not a constant string of upload to the issue queue,
have a minor bug pointed out, reroll with bug fixes. Two sets of
eyes on any of those patches at any one time would be awesome,
and I think it would help get them ready to be committed.
```

18:43 Uhr

FR1 antwortet über die Freenet Dev Liste:

```
My first thoughts are that this sort of thing is probably more
useful in environments where people are in close physical and
temporal proximity to each other (eg. an office).

The freenet project is made up of people scattered all over the
world; most of the development does not occur in parallel by
multiple developers in real time.
```

A.1.19 Donnerstag 15.10.09

14:39 Uhr

Ich antworte auf DR4:

```
>Just one point to clarify, there is no "Drupal Team" that could
>decide en masse to use a tool like this.
>Drupal is so distributed that we all have our own
>development workflows using a variety of tools, often in our
>basements rather than our offices (although there, too).  So
>"Could use and Drupal benefit" is really the wrong question to
>ask.

I know that there is nobody like an manager in an office to tell
the people what tool to use while developing, my question is if
there are workflows in which Saros might improve the development,
for example if there are a couple of people how can "meet" within
Saros and do pair reviews or just working at the module together.
At least this is also something I would like to find out, if you
say "No, we cannot use this because we are all working too
distributed and in different time zones" this is a result for me
but I strongly believe that it can be done and improve development,
but it also depends in the workflows you have.
>That said, I'm sure such a tool would prove useful to certain
>developers if they choose to use it.

Of course the developer himself has to choose it.
>Is it language-dependent?  Vis, most Eclipse devs work on Java,
>not PHP, so I am always wary of tools that may end up being
>centric to one particular language's development idiosyncrasies.

Saros is not language dependent, as far as other plugins are
following the eclipse guidelines. For PHP we have tested PDT,
Aptana, Zent and PHPeclipse. We have a full overview on this page:
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
If you would like to give Saros a try and something is missing,
please let me know and we will try to but it in the next release.

Greets, Eike
```

14:53 Uhr

DR7 antwortet darauf:

```
Eike, this sounds pretty neat. Thanks for brining it to our
attention. It would also be a good tool for training and
consulting,where you could save a trip to the customer's office
and work with them on code directly.
```

14:55 Uhr

DR8 antwortet mir:

```
The tool is usable, in some limited fashion - project teams etc...
It is not a Drupal tool - there is no such beast - drupal is the
tool itself.

Having said that the irc paste combo is used exactly in the manner
you suggest. Not just by the drupal crowd, but by most projects
I've touched.

Is there space for this tool - yes. Is it big? No. Until you can
collaborate with others - eclipse, emacs, textmate, put your
favourite environment, it is not going to be attractive. People
don't like to be dependent on a specific platform. Everyone has
their own.
```

15:05 Uhr

Ich antworte DR6:

```
>I was just looking for something like this! I'll be using this
>very soon on another project of mine =D
>
Great, is it also an Open Source Project? Would be nice if I can
help you with Saros, so that I can find out how you work with it
and what has to be improved in Saros
> As far as Drupal goes, I think that something like this
>would be particularly useful for core in a time
>like...oh,say....now. With code freeze ~38 hours away,
>efficiency is key to getting the patches from the exception
>list into core. If more than one person is working on a
>patch, it really makes sense to use something like this and
```

```
>have everyone looking at the same code at the same time so
>that it's not a constant string of upload to the issue
>queue, have a minor bug pointed out, reroll with bug fixes.
>Two sets of eyes on any of those patches at any one time
>would be awesome, and I think it would help get them ready
>to be committed.

Now, that sounds like good scope. Would be great if you give this
a try. Normally people work together with Saros and using Skype or
Mumlbe for
voice communication. I would offer you to join this session, to
give you realtime support if something doesn't work out.
What do you think about this?
Greets, Eike
```

15:14 Uhr

Ich antworte DR7:

```
>Eike, this sounds pretty neat. Thanks for brining it to our
> attention. It would also be a good tool for training and
> consulting, where you could save a trip to the customer's office
> and work with them on code directly.
>
Thank you. I also thought of the scope that you can introduce
newcomers to an Open Source Project more easily, by giving him/her
a short code tour with Saros. But this is just my thinking, since
I don't know if somebody has done this. What do you think?

\subsubsection*{15:17 Uhr}
Ich antworte auf DR5:
\begin{footnotesize}\begin{verbatim}
>I was thinking along the same lines.  Most of the Drupal
>development that I've seen is either:

> A) Written by one person, then reviewed / improved
>asynchronously in the issue tracker rather than working at
>the same time
> B) Written by a team of people for a project in-house, in
```

```
>which case it would be the same as basically any other in-
>house PHP development

If you are working just like this perhaps Saros really doesn't
make sense. But what is about the issue Cameron was thinking of?
Working together with Saros and getting the patches in the core?
>I use Eclipse for all of my PHP development (and know some
>others who doas well), but I probably won't ever use
>something like Saros unless I got a tele-working position
>somewhere.

>Some other module / core contributors might find it useful
>for those tasks, but it doesn't seem like a common need.

Would be great if there are some people, how would like to find
this out.

Greets, Eike
```

15:43 Uhr

DR5 antwortet mir:

```
>If you are working just like this perhaps Saros really doesn't
>make sense. But what is about the issue Cameron was thinking of?
>Working together with Saros and getting the patches in the core?

It might work in some cases, but I suspect it won't reach wide
adoption. Time is very tight, so any delays messing with
connection problems, etc. aren't really acceptable. So, both
parties would need to have already used it in the past and
frequently for it to be usable during crunch time.
```

18:05 Uhr

DR9 antwortet mir:

```
> Thank you. I also thought of the scope that you can introduce
> newcomers   to an Open Source Project more easily, by giving
```

```
> him/her a short code tour with Saros. But this is just my
> thinking, since I don't know if
> somebody has done this. What do you think?
>
> Greets, Eike
This is exactly what i find myself doing with my developers that
are new to the "drupal way" I plan on installing this today and
integrating it into my eclipse workflow for a test run.
Thanks for bringing it to my attention. Its a novel solution to an
old problem of communicating via line numbers :)
. -mf
```

A.1.20 Samstag 17.10.09

21:43 Uhr

FR2 antwortet auf FR1:

```
I second that.

Though it would be useful if we had the capability to mark code as
reviewed line-by-line in the git repository. Imagine some
color-coded display of the source code: The more people have
reviewed a certain line of code, the more it becomes dark green.
Not reviewed code would be red.

But anything which requires live interaction between developers
won't work, we all work at different times of day.
```

A.1.21 Sonntag 18.10.09

16:32 Uhr

FR3 antwortet FR2

```
In the past I've suggested looking into this tool:
http://smartbear.com/codecollab.php  As I recall, we even set it
up, but nobody used it.
```

A.1.22 Montag 19.10.09

12:51 Uhr

Ich antworte DR5:

```
>It might work in some cases, but I suspect it won't reach wide
>adoption.  Time is very tight, so any delays messing with
>connection problems, etc. aren't really acceptable.  So,
>both parties would need to have already used it in the past
>and frequently for it to be usable during crunch time.

That's true, you don't want people playing around with new tools
while crunch time, but at some point you have to start, perhaps
after the crunch time  ;-)

Greets, Eike
I second that.
```

13:04 Uhr

Ich antworte auf FR2:

```
>> >> My first thoughts are that this sort of thing is probably
>> >> more useful in environments where people are in close
>> >> physical and temporal proximity to each other (eg an office).
>> >>
>> >> The freenet project is made up of people scattered all over
>> >> the world; most of the development does not occur in
>> >> parallel by multiple developers in real time.
>> >>
That is always a problem for real time collaboration, especially
if the people strongly distributed and too few to find together
at the same time. I don't know how many you are and how
distributed, but sometimes people know each other or live near
each other and than it can be done.
> >
> > I second that.
> >
> > Though it would be useful if we had the capability to mark code
```

```
> > as reviewed line-by-line in the git repository. Imagine some
> > color-coded display of the source code: The more people have
> > reviewed a certain line of code, the more it becomes dark
> > green. Not reviewed code would be red.
> >
Ok, that would be a nice feature, but I think this can't be done
with Saros. That means it can be done but only in realtime, e.g. 4
people in a session can see which line is review by whom.
> > But anything which requires live interaction between developers
> > won't work, we all work at different times of day.
> >
Ok, now that is clear. Saros can just be used in real time.

Greets, Eike
```

16:06

Ich antworte WP1:

```
>> >> Saros is an Eclipse plugin for collaborative text editing and
>> >> distributed pair programming, i.e. it allows two or more
>> >> developers to work together in real-time on the same files.
> >
> > Speaking as somebody who's been coding professionally for 10
> > years, I couldn't let this go by...
> >
> > I've had to do this sort of thing before (not distributed, but
> > pair programming). It is, by far, the worst idea ever conceived
> > by man(agers). I would rip my own arm off and beat myself to
> > death withit before I'd ever do this sort of thing again.
> >
Perhaps you made some bad experiences, with pair programming and it
is good that you share it, but there are a lot of people who have
made different experiences and find it very useful.
> > The theory of pair programming is that you have two developers
> > working on the same piece of code, able to notice and correct
> > mistakes.
> >
> > The reality of pair programming is that one guy is better than
```

> > the other guy, and spends half his time explaining the problem
> > and solution to the other guy, instead of simply coding the
> > damn thing. The lesser of the two programmers only drags the
> > other guy down to his level, and wastes both of their time.
> >
Yes that can happen, but did you ever thought of the knowledge
transfer that happens in this situation. I think there is no better
way for training a newcomer than putting him together with a
skilled developer explaining him while working how everything works.
By the way are there such big differences in the wordpress core
team?
> > The only way that pair programming works is if both programmers
> > are a) equally skilled and b) not particularly skilled. In
> > other words, for newbs fresh out of college, it works fine. For
> > a seasoned programmer, the very notion is a valid reason to
> > look for a job elsewhere.
> >
> > In general, it's a horrible practice, and anybody actively
> > using it is an idiot. IMO, of course.
> >
Of course a programmer who doesn't want to explain what he does and
prefers work alone will not like pair programming.
> > Also, peer review is highly overrated as well, for other reasons.
> > People who actively write good code don't need it, and people
> > who don't write good code shouldn't be put in the position of
> > writing code in the first place.
> >
That's also one think you can learn while working together with a
skilled programmer, although I don't believe that good code is
always bugfree, everybody makes mistakes.

Al least, I just wanted to find out, if such a tool like Saros can
improve/be useful for open source software development. There are
lost of scopes behind just doing pair programming, for example
reviewing security patches.

Greets, Eike

19:22 Uhr

DR4 antwortet mir:

```
I've bounced between a number of different IDEs and dislike most
of them. At the moment I'm using Eclipse.  Really, it is NOT a
casual-use tool. Especially for Drupal, it's not worth it unless
you adopt it and take the time to customize it.  I can't see
anyone using Eclipse just for Saros and "other" for everything
else.

If adopted by a couple of people it could be enough to get someone
to switch to Eclipse, perhaps, especially if some other Eclipse
tools like Mylin had good Drupal-specific integration, but I can't
see Eclipse itself as a casual-use IDE.

(I say this having not tried it yet myself.)
```

A.1.23 Donnerstag 21.10.09

14:08 Uhr

WP2 antwortet mir:

```
>>
>> Perhaps you made some bad experiences, with pair programming
>> and it is good that you share it, but there are a lot of people
>> who have made different experiences and find it very useful.
>>
>>
Which is all well and good, if the point of the exercise is
training up newcomers.  However, if the intent is to produce clean,
bug-free code in a fast and efficient manner, what Otto pointed out
is true in many/most situations: you're going to handcuff a gifted
programmer and produce worse code, slower.

It's fine and dandy to do pair programming if you want to initiate
knowledge transfer (what we used to call "teaching"), but as I
said, it's not exactly the best when it comes to clean code,
quickly developed.
```

A.1.24 Mittwoch 03.11.09

22:46 Uhr

Antwort vom TYPO3 Security Team TYS1:

```
Thanks a lot for your offer. Your project sounds really
interesting.
Nevertheless we do not think it would be of great use for our
workflows.

However I personally will have a look at the plugin.
```

A.1.25 Donnerstag 04.11.09

18:01 Uhr

Ich schreibe eine Email an JUnit und eclipse-cs.

```
Dear [JUnit/Eclispe-cs Admin],

My name is Eike Starkmann and I'm  working as part of the Saros Team at
the Freie University in Berlin.

Saros is an Eclipse plugin for collaborative text editing and
distributed pair programming, i.e. it allows two or more developers to
work together in real-time on the same files. It is similar to Gobby,
SubEthaEdit or Google Docs but focuses on programming in Eclipse.

In my opinion your  can benefit from Saros because I think it brings
many advantages to Open Source Software development:

* Distributed Pair Programming is like a live peer review. This should
help with finding good design, get rid of bugs, increase readability, etc.

* Transferring knowledge should be easier to do when more than one
person look at and work with the same code. This should also help to
give new developers an introduction to the code.

* In contrast to screen sharing, Saros only shares your actions inside
```

```
of Eclipse with regards to the project you are both working on (think
privacy) and you are still independent to explore the project on your own.

Saros can be useful in the following contexts:

* Working on complicated problems in the code
* Performing code reviews
* Debugging
* Code presentation
* Code sprints
* Introducing new developers to the project
* ...

It is my master thesis to figure out whether Saros is useful when
developing Free/Open Source Software.

What do you think? Could you and [JUnit/eclipse-cs] benefit from doing pair
programming using Saros? If you think so I would like to join our
project and help you with using Saros.

If you are interested in Saros but still curious about how it works
please visit our website or feel free to contact me.

I hope you will find Saros useful and give me feedback.

Kind regards, Eike Starkmann

[eclipse-cs:P.S.: Sorry for contacting all of you but i didn't know whom of you to ask.]

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

21:45 Uhr

Antwort von JU1:

```
Eike,

That definitely sounds interesting.  Is the plug-in ready to go,
or is there something I should know first?
```

22:04 Uhr

Reaktion von JU2:

```
Just thought I'd forward this interesting research project to you;
I thought you guys might be interested!

Hope things are going well; its been a while. Say hi to everyone
for me!

-----Original Message-----
From: JU1
Sent: Wednesday, November 04, 2009 3:46 PM
To: Eike Starkmann
Cc: junit-devel@lists.sourceforge.net
Subject: Re: [Junit-devel] distributed pair programming for JUnit?
Eike,

That definitely sounds interesting.
in ready to go,
or is there something I should know first?
```

A.1.26 Freitag 05.11.09

12:14 Uhr

Nochmalige Nachfrage auf der Drupal-Liste:

```
Hi,
i would like to know if anyone tried Saros and might want to give
me a feedback. There were a couple of people how said that Saros
```

```
might be useful for Drupal development. Perhaps we could try it
for the next release.

Greets, Eike
```

12:17 Uhr

Direkte Nachfrage bei DR6:

```
>I was just looking for something like this! I'll be using this
>very soon on another project of mine =D
Did you tried Saros? Can you perhaps give me a small feedback, so
that we know what we have to improve or what is working well ;-)

Thanks, Eike
```

A.1.27 Donnerstag 19.11.09

14:37 Uhr

```
Hi,

I am working in the Institute of Computer Science at the
Free University of Berlin where we are creating ways to improve
the process of writing software, especially for Free/Open Source
Software developers.

I read on your website that the last Bug Day was on 29 October,
will the next one be 27th November?. I understand that on this
day
you work together at the same time, communicating over IRC and
fixing bugs. Have you ever thought of doing this in a more
collaborative way?

Our institute develops Saros, an open source Eclipse plug-in
that allows two or more developers to work together in
real-time developing code, training newcomers, or performing
peer reviews. It is part of my masters thesis to assess how
useful Saros is when developing Free/Open Source Software.
```

```
It is my hope that you would be able to make use of Saros and
that I
could work with you to observe how successful it is. If you
are interested, I will offer to join your project (especially
during the next Bug Day) to give you support while working
with Saros.

Of course I know that nobody wants to deal with new tools
during crunch time, but i believe there should be enough
time before the next Bug Day to discuss my proposal and to
get in touch with Saros.

What do you think? Could you benefit from using Saros?

Kind regards, Eike Starkmann

Website: https://www.inf.fu-berlin.de/w/SE/DPP
```

14:54 Uhr

red1 antwortet mir.

```
Halo Eike,
I am really glad, i mean more than happy to learn this.
Even though i be engaged in training class in Thailand on
that day, i will try to clock in and at least show the class
if there is action going on with you around.

So, let's stick around more :)
```

16:20 Uhr

Ausgewählte Passagen aus dem IRC Mitschnitt.

```
[16:23] <eike_> so will there be a bug day on 26th, because i
didn't find any informations on the website about that
[16:25] <red1> eike_: is it done from scratch or built upon an
earlier idea?
```

```
[16:25] <red1> wonder if this is good for 'pair programming' ?
[16:25] <red1> any social tool can be revolutionary :D
[16:26] <eike_> its based on jupiter algorithm, but the rest is
 from scratch
[16:27] <eike_> pair programming is one scope but not the only
 one
[16:27] <red1> what more exciting stuff this brings?
```

16:28 Uhr

```
[16:28] <red1> clearly Saros looks revolutionary
[16:28] <red1> having witeboard on your own eclipse with other
team members
[16:30] <red1> if it can be stored for later replay.. awesome
possibilities
[16:31] <eike_> there is nothing like saving a session right
now but its a nice idea
[16:31] <red1> a GoogleWave capabiity right in Eclipse :>
[16:31] <eike_> right
[16:32] <eike_> with a focus on programming
```

16:35 Uhr

```
[16:35] <eike_> but if people are intrested we can also make a
Saros session in the next days and i guide you.
[16:47] <vosscom_nasir> red1 when next bug day coming ?
[16:48] <eike_> i would like to too
[16:48] <red1> i like to see
[16:48] <red1> its just exciting to know
[16:49] <eike_> i would like to know too :D
```

16:50 Uhr

Sandor (spawndyd)loggt sich in den Chat ein.

```
[16:50] <red1> have you been using it for real projects?
[16:50] <red1> is there some trial report?
[16:50] <eike_> yes, some comercial firms use it.
[16:50] <red1> heh.. thats fantastic
```

```
[16:51] <red1> commercial use of a university project
[16:51] <red1> are you part of the development tram?
[16:51] <red1> team
[16:51] <spawnyd> red1: I am.
[16:52] <vosscom_nasir> eike thinking ans :)
[16:52] <eike_> :)
[16:53] <red1> spawnyd: its great to see your work :D
[16:53] <vosscom_nasir> eike some above question from red1 was
for you ?
[16:53] <eike_> im asking OSS Projects if they want to try
Saros and then improve it
[16:54] <red1> i can imagine a whole OSS team working with multiple java artifacts at the same
[16:54] <red1> and if it has offline update it be better
[16:54] <red1> than the present diff
[16:54] <eike_> and give them support then.  so that there is
not only commercial use.
[16:55] <red1> the present one using diff files during compare
is not this visible
[16:55] <red1> eike_ spawnyd: what are you guys plans for this?
[16:56] <red1> will it remain in the unviersity?
[16:56] <eike_> its OSS
[16:56] <spawnyd> red1: I have to improve the network module
within my thesis
[16:57] <eike_> And i have to find out if distributed pair
programming can work or improce OSS development
[16:57] <spawnyd> red1: for now i have to implement a
testframework, what is a hard thing to do with our architecture.
[16:57] <red1> well.. i m now in the middle of an International
Software Testing Quality Board exam
[16:57] <red1> tomorrow morning that is
[16:58] <eike_> but a the moment only students or peopel for the
university are developing Saros
```

16:59 Uhr

```
[16:59] <red1> we have to look for a 'pair' from our project
[16:59] <red1> maybe croo  is my match :D
[16:59] <eike_> can also be more than a pair
```

```
[16:59] <red1> i know trifon wont want to pair with a dumb one
like me
[16:59] <red1> yes u are right
[17:00] <eike_> i can join you too, but wont do much programming
work ;-)

[17:00] <vosscom_nasir> red1 who is this one eike ?
[17:02] <vosscom_nasir> lookig busy ....:)
[17:02] <eike_> i introduced myself in someway here:
https://sourceforge.net/projects/adempiere/forums/forum/610548
/topic/3463747
[17:02] <vosscom_nasir> thanks eike
[17:02] <red1> vosscom_nasir: you should ask more in forums :D
[17:03] <eike_> np
```

17:03 Uhr

```
[17:03] <red1> eike_: are there constraints of scalability?
[17:03] <red1> what can be the max users?
[17:04] <eike_> i think we teste up to 5 people
[17:04] <eike_> tested

[17:04] <red1> eike_: how did it go with 5? Any issues?
[17:05] <spawnyd> red1: try to use it with 2. Last we had
problems with more then 2. :/
[17:05] <eike_> but in dont know if that really makes sence: 5
people working at the same document.

[17:05] <red1> can there be a read only mode spawnyd ?
[17:05] <red1> yeah u got a point eike_
[17:05] <eike_> it can male sence as kind of a code guid
through or presentation
[17:06] <spawnyd> red1: It depends on XMPP and sometimes it has
a delay that really suck.
[17:06] <red1> but if lets say 5 can work on it and the rest
can just view it, it can be a great teaching tool
[17:06] <spawnyd> red1: yes
[17:06] <eike_> yes
```

```
[17:06] <red1> to get more ppl on board such project
[17:06] <eike_> the host defines who else can write
[17:06] <red1> and process of distributing knowlege increases
[17:07] <red1> seems u have thought of it quite well
[17:07] <eike_> thanks
```

17:07 Uhr

```
[17:07] <red1> what are you other motivation for such an addon?
[17:07] <spawnyd> red1: we think teaching is a use case of a
tool like saros.
[17:07] <red1> ah .. i just found a local java guru now
[17:07] <spawnyd> red1: I used it for homeworks.
[17:07] <red1> was inducting him into our codebase
[17:08] <red1> now i know what to say to him tomorrow
[17:08] <red1> how spawnyd ... is it used in your homework?
[17:08] <eike_> my main motivation is to improve oss development
[17:08] <eike_> but homework is an issu since we have to do it
in pairs
```

17:08 Uhr

```
[17:08] <red1> how did that idea first appear?
[17:09] <red1> who is the original idea guy?
[17:09] <red1> :D
[17:09] <eike_> questions
[17:09] <red1> curious
[17:09]  * spawnyd was DOSed by red1
[17:09] <red1> well.. i am pitching on OSS to the govt here
[17:09] <red1> and i need such cases to tell them
[17:10] <red1> on how ideas are spawned :D
[17:10] <spawnyd> The initial thesis was written by a guy
 called Riad.
[17:10] <red1> govt here is willing to throw money but at wrong
targets
[17:10] <red1> now the buzz is Bio technology
[17:10] <eike_> i think the original research was about pair
programming and then how the benefits of pair programming can
```

be achived situation where the developers can not meet in real
life
[17:11] <eike_> like in oss projects

17:16 Uhr

[17:16] <red1> reading about Mumble part
[17:16] <red1> hmm it has voice?
[17:16] <eike_> yes
[17:17] <red1> wow
[17:17] <red1> voice conferencing while using Saros?
[17:17] <eike_> its optional but i makes it more easy to
coordinate
[17:17] <red1> u got to be kidding me
[17:17] <red1> now i cant sleep
[17:17] <spawnyd> red1: you should use an external tool for
voice.
[17:18] <red1> ah ok.. i can use skype
[17:18] <eike_> yes
[17:18] <eike_> its not build in
[17:19] <spawnyd> or mumble.

17:19 Uhr

[17:19] <red1> hows the bandwidth requirement?
[17:20] <spawnyd> if you can use voice, you can use saros
[17:20] <red1> we can try it to get past some sticky issues
on peer review
[17:20] <spawnyd> it's less then voice.
[17:21] <eike_> but also dends on what are you doing, if yu
paste 3mb text in a session it will take a while and need more
bandwith
[17:22] <eike_> depends
[17:22] <spawnyd> the initial project sharing, too.
[17:22] <eike_> the normal case is to sync via git or svn
[17:22] <spawnyd> If you sync the hole project and not use an
existing one.
[17:22] <eike_> before

```
[17:22] <red1> so u know GIT :D
[17:23] <eike_> yes a bit
```

17:23 Uhr

```
[17:23] <red1> we do have a serious peer review constraint
[17:24] <red1> our trunk has to be well guarded and yet we want
faster evolution
[17:24] <red1> with some Saros, i can imagine more closer devs
insight into how each other work can help the goodwill
[17:25] <eike_> yes ineed
[17:25] <eike_> indeed
```

17:26 Uhr

```
[17:26] <red1> did u try calculating the impact of using such
tool?
[17:26] <red1> i mean positive rate of productivity?
[17:26] <eike_> thats hard to do because its hard to measure
[17:27] <spawnyd> red1: If you work together you won't IRC the
whole time. That result in more productivity or?
[17:27] <eike_> but researches about pair programming showed
that they are producing less bugs
[17:27] <eike_> and you have this knowledge transfer
[17:27] <eike_> they also find better designs or solutions for
hard problems
[17:28] <red1> it can give a new dimension to the dev experience
[17:28] <eike_> and of course many developers find it enjoyable
working like this
[17:28] <eike_> yes
```

17:29 Uhr

AD1 installiert Saros.

```
[17:29] <red1> so how many devs u have working on this?
[17:30] <eike_> 5 at the moment
[17:30] <eike_> but the are all doing  thesis
[17:31] <vosscom_nasir> eike what is title of your thesis ?
```

```
[17:31] <eike_> its in german i have to translate
[17:31] <red1> updating my eclipse with it now
[17:32] <eike_> Distributed Pair Programming in Open Souce
Software Projects
```

17:32 Uhr

Einladung zu Session und Installation von Saros.

```
[17:32] <eike_> want to try a session with me?
[17:32] <red1> of course
[17:33] <eike_> wait i come online
[17:33] <red1> its still pending..
[17:33] <red1> must be searching
[17:33] <red1> ok found..
[17:33] <red1> installing
[17:34] <eike_> ok i need to update too
[17:34] <red1> bar moving slowly...
[17:35] <red1> i guess i need to connect to your jabber server
```

17:35 Uhr

AD2 hat antwortet und hat Saros ebenfalls installiert.

```
[17:35] <croo> when you share a project ... you get a user
list ... where does that list come from? because mine is empty
[17:35] <croo> also once you share a project how do stop
sharing?
[17:35] <red1> hmm croo beats me to it
[17:35] <red1> martians are faster
[17:36] <croo> we come in peace
[17:36] <spawnyd> croo: if you have friends in your roster and
they have saros installed then you see someone..
[17:37] <croo> ah friends in my roster .. let me look at that
[17:37] <red1> done
[17:37] <spawnyd> croo: to unshare the project you have to use
a button on the share project view
[17:37] <eike_> eike.starkmann@jabber.org
[17:37] <eike_> thats my jabber accound
[17:37] <eike_> i recomand using jabber.org
```

17:37 Uhr

AD1 hat Probleme mit Saros.

```
[17:37] <red1> so it can work on Mac?
[17:38] <red1> i guess i need to relaunch my eclipse
[17:38] <eike_> croo: its comes from the host which is opening
the session
[17:38] <spawnyd> yes I use it on a mac
```

17:39 Uhr

Hinzufügen von AD2 war erfolgreich.

```
[17:39] <eike_> hi croo
[17:39] <croo> hi eike :)
(Bezieht sich darauf, das wir uns gegenseitig im Roster sehen,
Anm. d. Verf.)
[17:39] <croo> so must you have the same project open as me to
share?
[17:39] <red1> cant see saros in preferences here
[17:40] <croo> window preferences red1 .. not project
[17:40] <eike_> not yet i invite you now
[17:40] <croo> at least that was my error
[17:40] <spawnyd> croo: no you right click on a project you
want to share
[17:40] <red1> croo: from my mac.. i did go to preferences
[17:40] <eike_> ok you have to invite me since i cant invite
you because you already opend a session
[17:40] <croo> I meant for red1 ... he looks for saros in
preferences ... I looked first in project prreferences
[17:41] <croo> but actually it's window/preferences
[17:41] <spawnyd> red1: do you restarted after installing it?
[17:41] <croo> inviting you
[17:41] <spawnyd> croo: ok
[17:41] <eike_> i hope its not that big ;-)
[17:41] <red1> yes i relaunch my eclipse
[17:41] <red1> then i go to Eclipse top bar and select
Properties in the pull down
```

17:42 Uhr

AD3 beteiligt sich an der Diskussion.

```
[17:42] <MCalderon> elke_: hi. I am reading your SAROS info

[17:42] <croo> eike_, I invoted you to share a project
[17:42] <spawnyd> red1: it's Eclipse->Preferences
[17:43] <eike_> croo: yes, i accepted, but is it big? seems to
take ages
[17:43] <croo> yes it's adempiere
[17:43] <croo> :)
[17:43] <croo> massive
[17:43] <eike_> lol
[17:43] <croo> let me open a hello world
[17:43] <eike_> thats too big
[17:44] <eike_> since i have to get all
[17:44] <croo> IF both already have the project is it possible
then?
[17:44] <eike_> but im half done
[17:44] <eike_> shure
[17:44] <spawnyd> croo: yes you can choose an existing project.
[17:44] <eike_> because only the diff is shared the if there is
a diff in the projects
[17:44] <croo> yes I have an hello world I use to play with
syntax

[17:45] <MCalderon> elke_: the idea seems great. One "problem"
for the Adempiere community would be that we don't have the same sources
```

17:45 Uhr

AD1 hat immer noch Problem Saros zu installieren. Sandor hilft ihm dabei.

```
[17:45] <eike_> 90% done
[17:45] <croo> but first I must unshare adempiere I guess?
[17:45] <eike_> croo: right
[17:45] <croo> oh I just shut adempiere thinking it was too big
[17:45] <eike_> croo: only ons session at one time
[17:45] <spawnyd> croo: yes
```

[17:45] <spawnyd> croo: it's the most right button on session view
[17:46] <red1> wait how do i check if that plugin exist?
[17:46] <red1> in mine

[17:46] <eike_> MCalderon: hmm ok and the cannot sync before working together?

[17:47] <spawnyd> red1: in your Eclipse directory is a plugins folder.
[17:47] <croo> ok so I cannot invite you now
[17:47] <red1> spawnyd: what is the full qualified name?
[17:47] <spawnyd> red1: If you have de.fu_berlin.inf.dpp_9.10.30.r1833.jar there it's installed.
[17:47] <red1> ok looking
[17:48] <eike_> red1: or just check if there is the Roster under Windows Views
[17:48] <eike_> croo: try now
[17:48] <red1> found de.fu.. under features
[17:48] <red1> not plugins toh
[17:49] <eike_> red1: Window -Show view - Other - Saros
[17:49] <spawnyd> red1: it's not installed
[17:49] <spawnyd> try to remove the feature and install it from the updatesite
[17:49] <croo> eike_, you did appear and I selected you again but I still see an error
[17:49] <croo> roster window
[17:49] <croo> Out: IBB [ERROR]
[17:50] <eike_> try to reconnect then
[17:50] <red1> i am installing again
[17:50] <croo> ok
[17:50] <eike_> in the roster view
[17:51] <croo> hmm same ...
[17:51] <croo> might be my connection though
[17:51] <eike_> but im getting the project
[17:52] <eike_> here we are
(Die erste Session wurde aufgebaut und wir können Saros benutzten, Anm d. Verf.)

```
[17:52] <croo> ahok
[17:54] <croo> eike_ very useful I think
[17:54] <eike_> croo: thank
[17:54] <eike_> you
```

17:52 Uhr

```
[17:52] <MCalderon> elke_: yes. perhaps it would be a way of
using the same code
[17:53] <MCalderon> elke_: we usually have the sources in
different revisions with different changes of our own
[17:54] <MCalderon> with Saros we would download the latest
trunk and work on it
[17:54] <MCalderon> more secure
[17:55] <eike_> MCalderon: yes thats is the normal way of
working with it. I think
[17:55] <spawnyd> MCalderon: yes, but if you want to share a
different project we calculate diffs to send them. It's more
efficient then sending the  whole project.
[17:55] <MCalderon> last bug day I worked with a version a
couple of days old
[17:55] <MCalderon> elke_: I see
```

17:55 Uhr

Wir testen den Follow-Mode.

```
[17:55] <croo> can you browse around the classes just to see
how quick it works here
[17:55] <eike_> croo: try the follow mode
[17:56] <croo> ok following
[17:57] <eike_> croo: you need to revoce or driver role before
[17:57] <croo> done
[17:57] <croo> do I bneed to give you exlcuive driver or is
driver enough?
[17:58] <eike_> exclusive
[17:58] <eike_> so that you can follow me
[17:58] <croo> ok
[17:58] <croo> I think that just removes all other drivers...
```

```
right?
[17:58] <eike_> yes
[17:58] <croo> ok you are driver
[17:59] <croo> and I am following you
[18:00] <eike_> you see waht i do?
[18:00] <croo> I see no changes
[18:00] <spawnyd> croo: is the follow button enabled?
[18:00] <croo> yes I will do again trhough because first time
I was also driver
[18:01] <croo> ok disabled follow and following again
[18:01] <croo> still no screen changes though
[18:01] <spawnyd> your view should follow eike_'s view.
[18:02] <croo> must I open a specific perspective first?
[18:02] <croo> ah now
[18:02] <croo> I see
[18:02] <spawnyd> :)
[18:03] <croo> ok - work pretty snappy too
[18:03] <croo> a great teaching tool!
[18:03] <croo> never mind pair programming!
[18:04] <spawnyd> croo: we hope you enjoy :)
[18:04] <croo> very good guys ... well done
[18:04] <eike_> thanks
[18:04] <spawnyd> ty
```

18:04 Uhr

```
[18:04] <croo> so did I see you say it works on diff ... so
with something as big as adempiere
[18:05] <croo> once both sync with the trunk first then
[18:05] <croo> the size should not be such an issue?
[18:05] <spawnyd> croo: it should be ok.
[18:05] <croo> it is possible to develope in adempiere in two
fashions ... many small projects or one big one.  I think most
people prefer the one big one
[18:06] <eike_> croo: yes it works on diffs. its just the sync
at the begining if you have to sync everything
[18:06] <spawnyd> croo: Saros does not support multiple
projects, just one.
```

```
[18:06] <croo> ok well my workspace/adempiere folder is 2GB ..
that's just standard trunk!
[18:07] <croo> spawnyd, yes I meant that the project is smaller
then to sync initially
[18:08] <eike_> but you could sync in a faster way before like
via git or something, or?
[18:09] <croo> well yes I think we'd usually want to sync with
the SVN trunkn first
[18:09] <croo> so we both had the same revision
[18:09] <croo> but how then does it compare? must it compare
the two workspace project folders
[18:09] <croo> ?
[18:09] <croo> the source too is much less
[18:09] <croo> that is the binary with databases etc
[18:10] <croo> we wouldn't edit those but they are in the
directory
[18:10] <spawnyd> croo: hmm I dont really know, it's based on
files from the workspace. Is your binary db in your workspace?
[18:11] <croo> yes
[18:11] <croo> it's actually there a few times
[18:11] <croo> !!
[18:11] <croo> when we run the build it creates folders within
the workspace folder
[18:11] <spawnyd> croo: we also have a "share project
partially", try this.
[18:12] <croo> and puts in their a binary version .. plus a
zipped binary, a tar.gz binary and lots of other stuff .. hence
the 2gb
[18:12] <croo> partial share ok that's an idea
[18:12] <eike_> perhaps we should try it with the whole
adempiere project
[18:12] <MCalderon> can the synch be reduced to .java files?
[18:12] <spawnyd> croo: We only get Eclipse events not events
on the fs.
[18:12] <MCalderon> and xml files?
[18:12] <spawnyd> MCalderon: not now, but we can add a feature
request
[18:13] <MCalderon> wethese file types are actually he ones we
```

```
change
[18:13] <croo> yes well I do not believe adempiere follows any
known standard in that respect :)
[18:13] <croo> so I can understand it's form is not expected
[18:13] <spawnyd> MCalderon: xml files are not the problem,
the xml-editor should work.
```

18:14 Uhr

Partielles Sharing wird getestet.

```
[18:14] <croo> le me try a partial share
[18:14] <eike_> yes xml works but just code based
[18:14] <MCalderon> if you see our history changes, all mostly
happens is modifying/replacing .java, .sql, .xml  files
[18:15] <MCalderon> at building we create the .jars and .wars
[18:15] <croo> that's possible mario... there is a filter
option in the partial share (experimental)
[18:16] <croo> plus you can choose folders - much like you do
say when you commit
[18:16] <MCalderon> thes types do not not have necessarily to
be synchronized
[18:16] <MCalderon> croo: that's great
[18:16] <spawnyd> MCalderon: if you have trouble with SQL-view
or something drop us an email to dpp-user@lists.sourceforge.net
[18:16] <spawnyd> or write a bugreport on the sf-tracker.
[18:17] <croo> now if you could safe that selection it would
perfect! :)
[18:17] <eike_> or just contact me
[18:17] <MCalderon> spanwnyd: thanks
[18:17] <spawnyd> np
```

18:18 Uhr

```
[18:18] <MCalderon> now the question is how to proceed?
[18:18] <eike_> actually if you really are interested in using
it, i would like to give you support. e.g. introducing it like
i do now or joining a session, or just hang in irc for ansering
question
```

```
[18:19]  * spawnyd is off
[18:19] <eike_> MCalderon: yes that is a little bit up to you,
im open for everything
[18:20] <eike_> MCalderon: my proposal was to give you support
at the upcoming Bug Day if there is one
[18:20] <croo> ok I cut adempiere down to just source ina
shared folder
[18:21] <MCalderon> I ignore whether there will be a bug day
this month
[18:21] <MCalderon> I missed the last one
[18:21] <croo> but in retrospect I will cut it down to the
source of just one module
[18:21] <MCalderon> we'd had 10 days to try out and decide
[18:22] <croo> eike_, might be an idea to do that before hte
bug day otherwise no adempiere bugs will get fixed :)
[18:22] <eike_> MCalderon:  was just a proposal, if ther isnt
one i still can give support
[18:22] <croo> I think the plan is the last thursday of every
month is bug day??
[18:22] <MCalderon> elke_: thanks. I appreciate it
[18:22] <eike_> croo: shure.
[18:23] <eike_> croo: yes, thats what i read too.
[18:23] <eike_> croo: the files are comming
[18:23] <croo> but I can confirm to all it works well and
pretty nippy
[18:23] <croo> my connection is a little better than it was but
it's still just a 2mb connection which it not that great these days
```

18:24 Uhr

```
[18:24] <croo> eike_, oh I thought you cancelled so did myself
[18:24] <croo> I think I still selected too much
[18:24] <MCalderon> croo: could you sum up your experience in
Elke's post?
[18:24] <croo> I selected all hte src
[18:24] <croo> best probably is to take the source of just one
module
[18:25] <croo> MCalderon, will do
```

```
[18:25] <MCalderon> so other potential Bug Day attendees can
see them
[18:25] <MCalderon> cool!
[18:26] <eike_> croo: yes that would be great to have in the
forum
[18:26] <eike_> croo: ups still receving files
[18:26] <croo> typing right now
[18:31] <eike_> croo: no i wasent canceling
[18:32] <eike_> croo: i think if we want to try the adempiere
project i have to get the trunk before
[18:32] <eike_> croo: and im still getting data from you
[18:32] <croo> yes well if red1 gets sorted or mario was can
test that... or I can share a smaller part of the project
[18:32] <croo> as I said I shared all source
[18:33] <croo> which is a lot smaller than the 2gb but it
probably still a n awful lot
[18:33] <croo> usually issues are localised to a couple of
modules
[18:33] <croo> so feel free to cancel the current download
[18:34] <eike_> croo: ok then i cancel 2 gb is a bit much
[18:36] <eike_> croo: so i have to get the trunk i think, yes?
[18:39] <croo> eike_ :) now it wasn't 2gb ... I had selected
only source which was perhaps 150mb ... but still too much I
think
[18:39] <MCalderon> gentlemen: I'll leave the IRC because of
lunch.
[18:39] <MCalderon> Thanks to Elke (vielen Dank f?r die Info)
```

18:39 Uhr

Ich lade den aktuellen Trunk herunter.

```
[18:39] <croo> I think in practise it will be fine if people
share just he base folder or perhaps base + client
*[18:40] <croo> that's all that is required
[18:40] <MCalderon> elke_ auch in Mittelamerika wird Deutsch
gesprochen und verstanden!!!
[18:40] <MCalderon> I will keep in touch
[18:40] <MCalderon> c u!
```

```
[18:40] == MCalderon has changed nick to MCalderon_away
[18:44] <eike_> croo: ok. 150mb is still quiet a lot
[18:44] <croo> yes
[18:44] <croo> I understand
[18:44] <eike_> croo: normally i would hace most of this
[18:44] <croo> but the important thing is it's possible to
make it smaller
[18:44] <eike_> yeah
[18:44] <croo> with the partial share
[18:44] <croo> and it's filter
[18:45] <croo> if you are looking for enhancement ideas :)
(or maybe it's possible) it woudl be nice if you could store &
retrieve the partial project selected so that each time you
would not hav eto manually unselect bunches of the project
[18:46] <eike_> yes good point i will make a feature request
out of it
[18:47] <eike_> puhh 89% svn checkout
[18:47] <eike_> that really takes ages
[18:49] <croo> yes... three  things make it very big
[18:49] <croo> 1) it's a big project :)
[18:50] <croo> 2) it includes all the binaries of other
projects needed to run
[18:50] <croo> 3) it includes a number of seed databases (at
least oracle & postgres)
[18:51] <eike_> ok i see 2 and 3 are the biggest points,
right? ;-)
[18:51] <croo> I'm not sure if it's still there but once upon
a time there were also very large ER diagrams too
[18:51] <croo> yes
[18:52] <croo> I think tough there are a few million lines of
source code too
[18:53] <eike_> i think we have to implement a chat or irc
client in saros this switching in a session is nerves
[18:53] <eike_> ok thats a lot
[18:53] <eike_> finished
```

18:54 Uhr

Zweiter Versuch: Partial sharing von einem Modul führt zu dem Problem, dass auf meiner Seite der Rest von ADempiere gelöscht wird.

```
[18:53] <eike_> give it a secound try?
[18:54] <croo> you want me to share adempiere Trunk with you
again - or partially share one module?
[18:54] <eike_> yes
[18:55] <eike_> see if it works now
[18:55] <croo> ok lets start with one module .. base
[18:55] <croo> I'm just goin to diconnetc & reconnect first
[18:55] <eike_> ko
[18:55] <eike_> ok
[18:56] <croo> ok sharing the source of base module
[18:56] <croo> not sure how big that is
[18:57] <croo> still 22mb
[18:57] <eike_> ok i should allready have the most of it
[18:57] <croo> ok lets see
[18:59] <eike_> ok i now have to scan my local version of
adempiere
[18:59] <eike_> that takes a bit my computer isnt so fast
[18:59] <croo> ok I see ... Inviting Eike.... (3%)
[18:59] <croo> no problem
[19:00] <eike_> ups there was a problem
[19:00] <croo> ok it stopped but I don't see you in the session
[19:00] <croo> alright
[19:01] <croo> do I need to do something this end?
[19:01] <eike_> on  my side Saros wanted to delete all
packages that you didnt share, that is not an normal behaviour
[19:02] <croo> ah .. no :)  I can see what's it's thinking
though
```

19:03 Uhr

3. Versuch: Das ganze, jetzt synchrone Projekt wird benutzt.

```
[19:03] <eike_> try to share the whole project, it should work
fast i think
[19:03] <croo> ok
```

```
[19:04] <croo> I will disconnect again to start a fresh
[19:04] <croo> ok - it's off
[19:05] <croo> 2% mark
[19:05] <croo> not much network activity .. how doe sit compare?
[19:05] <croo> but lots of disk activity
[19:05] <croo> revision numbers?
[19:05] <eike_> building the filelist
[19:06] <croo> ok [INIT} displayed beside you name ... 3%
[19:06] <croo> [ERROR]
[19:06] <croo> OUT: IBB [ERROR]
[19:07] <eike_> yes but isnt ture
[19:07] <croo> ok
[19:07] <eike_> i get data from you
[19:07] <croo> ok and I see a little network activity now
[19:08] <croo> just a little mid .. might be irc .
[19:08] <croo> 3/4Ks
```

19:12 Uhr

Das Projekt benötigt partielles sharen, da sie mit großen Binärdateien arbeiten, die während des Kompilieren erstellt werden. Diese Binärdateien haben Zeitstempel, was dazu führt, dass sie bei jedem Entwickler anders sind, wodurch Saros diese immer synchronisieren müsste.

```
[19:12] <eike_> croo: i think something is worng on my side
the trunk isnt checked out correct or the partial share
distroyed somethink
[19:13] <croo> eike_, I am just thinking .. I am sharing the
whole project ... you sync with SVN but I have also built the
project so I in addition to the SVN all the build folders ..
PLUS the a folder with a Binary version ready for deployment
(which in testing I have also deployed), plus some packages ...
some zipped some tarred ready for to dtp to some remote servers
[19:13] <eike_> croo: finnaly i got it
[19:14] <eike_> croo: ok so we are not working at the same
trunk with the same files?
[19:14] <croo> well ... it's the same trunk
[19:14] <croo> but I have then built the system from those
files
```

```
[19:15] <croo> and so I had a lot of additional binary files
now too
[19:15] <eike_> ok, that is the point.
[19:15] <eike_> these files i don't have so they have to be
send
[19:15] <croo> yes
[19:15] <eike_> and that will take while
[19:16] <croo> with another developer he would of course have
these binary files too
[19:16] <eike_> yes, thats true
[19:16] <croo> but if we built at different times they woudl
have different timestamps etc
[19:16] <croo> so how would the sync decide if they are the
same file?
[19:16] <eike_> will that be much?
[19:17] <croo> yes
[19:17] <eike_> ok.
[19:17] <croo> one folder alone (the binary distributions) is
950MB
[19:17] <eike_> so you have to use partial share
[19:18] <croo> yes
[19:18] <eike_> there is rather no other option, or have a very
fast internet connection and no firewall
[19:18] <croo> I have just 2mb
[19:18] <croo> :)
[19:18] <eike_> the p2p connection is much faster, now we are
sending over jabber.org
[19:19] <croo> I've not used jabber before
[19:19] <Kai_> So croo, what's your first impression of Saros?
[19:19] <croo> it's works very well ... but the size of
adempiere is causing a problem
[19:19] <eike_> croo: im canceling
[19:20] <croo> was proposed as a pair development tool
[19:20] <croo> but would also be very good for training
[19:20] <Kai_> The size of Adempiere is always causing
problems. :-)
[19:20] <croo> eike_, ok
[19:20] <croo> :)
```

```
[19:21] <eike_> and it structure, we didnt thought of haven
big binary databases in the workspace
[19:21] <croo> there is an option to partially share but we
have found an unwanted side effect
[19:21] <croo> yes and the built for deployment versions
[19:21] <eike_> yeah we will go through the logs tomorrow and
see what that was.
```

19:22 Uhr

```
[19:22] <croo> if you have built adempiere too and had the
same binaries ...
[19:22] <croo> how would it now if they were the same as mine?
[19:22] <eike_> yes
[19:23] <croo> does it look at timestamps or revision numbers
or something like that?
[19:23] <croo> how would it now -> how would it KNOW
[19:23] <croo> I must improve my English :)
[19:23] <eike_> then only the filelist would be send saros
would recocnize the diffs and transmitt th diffs. which would
be zero then.
[19:24] <eike_> its using file hashes
[19:24] <croo> ah ok
[19:24] <croo> that's the bit I was missing
[19:24] <croo> so then if two people synced with trunk and then
built the system
[19:25] <croo> THEN there would nothing to sync and it would
be quicker
[19:25] <croo> still maybe 5-10 initially
[19:25] <croo> but then you are in sync for the day
[19:25] <eike_> yes exactly
[19:25] <croo> 5-10 mins that is
[19:25] <croo> ok
[19:26] <croo> thank you for all your help
[19:26] <eike_> but i will be here toworrow
[19:26] <croo> I will post our adventures in the forum
[19:26] <eike_> if you want to test or somethink like that
[19:26] <eike_> he he yes that would be nice
```

```
[19:26] <eike_> cu
[19:27] <croo> ok fiereabend!
[19:27] <croo> bye
[19:27] <eike_> genau
```

19:56 Uhr

Colin erstellt eine Zusammenfassung unserer Saros Sitzung.

```
Eike kindly talked me through a little demo from IRC so I
thought I would share my experiences here.

To be honest I was a little hesitant about install Saros
because it requires eclipse is version 3.4 or 3.5 and only my
main work system had this version installed so I was afraid I
might cause some instability. But installation worked without
a hitch and my worries turned out to be unfounded.

With Saros installed next I created a jabber account to work
with and luckily my trusty "croo" handle was still available :)

I must say the instructions on the link provided
(and again here
https://www.inf.fu-berlin.de/w/SE/DPP#Via_Update_Site) were
very clear and easy to follow.

Once all was installed I did an update with Adempiere and
shared it with Eike.

The developer you share with does not need to have the eclipse
project already on their system, and in this case Eike didn't
have Admepiere. However this meant he needed to download my
entire adempiereTrunk eclipse workspace which because I have
build & deployed it within eclipse was just over 2GB.... so
once we realised this we instead decided to begin with a small
hello world project. :)

The connect & sharing was straight forward. And the response
quite fast. For the record my internet connection these days
```

has improved since my 3g days .. now I have a very stable 2mb connection.

If you read the Saros website you will learn that 1 or more user may edit the files as once. The person with edit ability is called a "Driver" and more than one developer man be a "Driver". When only 1 developer has the Driver ability they are said to be the "Exclusive Driver".

We tested first the Exclusive Driver where one person leads and does the typing while the other watches. Next we tried Dual Driver where two people can each make edits. In theory two people can edit the same file at the same time but in reality this can cause conflicts so you wouldn't really want two actually typing at the same time but this mode saves time switching between who is the Exclusive Driver - a little bit of co-ordination is all that is required and this mode works quite well.

Next we tested the "Follow the user" functionality. In the "Shared Projects Session" tab, we right click on a user and give them "Exclusive Driver" control then again right click and select "Follow this User". Now where ever that Driver goes my eclipse window changes to follow, and of course all text entered appears too. And it's not just text, if the Driver creates new classes or methods my eclipse perspective changes to reflect this just as if I was typing in the changes myself - so my outline is updated for example.

So back to Adempiere then. As I mentioned, when you build and deploy Adempiere from with the eclipse you end up with a pretty huge workspace folder .. mine is 2GB. When you share a project, the two projects are diff'd first (a filelist with a hash code per file is created and this is what is compared). You can share a project the other user didn't have and the whole thing is copied - or a project both have in which case just the different files need be transfered). So sharing Adempiere in all it's 2GB with Eike who had not installed it as an eclipse

project was not really practical.

But all is not lost Saros does have a still experimental Partial Share. In this mode, when you share, you will be prompted for the folders to include or exclude (it works much like the selection when you use Team->Commit on the project level), and there is an addition filter to auto select/deselect based on filetype which could be very useful.

Eike, downloaded the trunk from SVN. And we began with a "Share Partial project" that included all the source from Base. Just once module but perhaps the biggest at 22mb which took perhaps 5 mins to sync up our systems which is much more practical. Alas, we ran into a issue... to sync the project it wanted to delete all the files I didn't share from Eike's project. We can see the logic - but that's now what we require.

Now, both Eike & I have sync'd trunk versions we decide to test the whole project sync again.

Initially all seemed well .. it took somewhere between 5 and 10 mins to diff the file structure and decide what needed to be sync'd alas since i had built the project using our build.xml and even deployed that under the adempiere/Adempiere folder the resulting diff was still over 1GB.

The secret we decided is that both developers should sync and then run the build first... then both systems would be in sync (or nearly if there were some small local changes) ... in which case the sync part would be small and would proceed faster.

Also worth noting that once the initial sync is made the pair could could work all day without further lengthy waits for synchronization.

So in conclusion;
The actual editing & follow me works very well and seems quite nippy, but some strategy to get around the fact that an

```
adempiere workspace is potentially up to 2gb in size.

The partial project share is one potential .. when the deletion
issue is resolved. And another might be to work with adempiere
in the "many projects" approach favoured by some.

Many thanks to Eike, and spawnyd too, for their patience and
help in IRC.

I hope this is helpful to others.

colin
```

22:32 Uhr

red1 antwortet croo.

```
Colin,
I have to retire to my inner chamber while all this is
happening and did not let the doorman sound the trumpets as not
to disturb the good work.

Upon awakening dawn here, i tried to install again, and it went
thru like magic. Particularly in the end it allows me to see
properties in all its glory.

However on pushing the connect icon in Roster, it connects to
this url bernas.com.my which is known to be a local company
doing rice mills for the government (lol, but since when did i
plunder the folks rice?) and supply me as an error.

Going off now to my worldly rounds, and shall attend to this
probably off and on whenever i can. Like to say this is amazing
and wondrous of what it can do to normal dummies like me.
Imagine what my friends will now think when they see me code at
the 'speed' of Carlos Ruiz, Heng Sin and Teo Sarca combined!

Real thanks to the Saros team of Eike and Spawnyd last nite for
been online in IRC and really efficient in teaching the Irish
```

```
peasantry here. I am sure in due time the whole world shall
take this on as the next best thing.

red1
```

A.1.28 Montag 23.11.09

12:13 Uhr

Ich antworte croo.

```
Thanks croo for this detailed summary of the session we had on
Thursday.If anyone else would like to try it out, i will be in
IRC channel tomorrow (Tuesday) and Wednesday from 15:00 h GMT.
If somebody has no time on this days just post your proposal
here then we can make an meeting, i'm flexible.
I'll also be there on 26th, when the Bug Day takes place. If
somebody would like to work with Saros on that day it would be
nice to invite me too, so that i can give support.
```

A.1.29 Mittwoch 25.11.09

13:56 Uhr

```
Yesterday i was working on how to partial share a big project
like ADempiere. We found out that there are some bugs in the
partial sharing (still experimental) and it is not working yet.
But we found a workaround in how to do partial sharing:

The idea is that Saros doesn't share files marked as derived.
So you can mark everything you don't want to share with others
as derived in eclipse. If you mark a folder the whole folder
will be ignored.
I know this project has lots of files and folders, but you only
have to do it once.

Marking folders/files as derived can done this way:
Open your project with the "Navigator" then right-click on the
folder/file you want to mark as derived (just one at a time).
```

```
Then click "Properties"
and there you find the check-box "Derived".

This will solve the problems croo and me had when trying to
share ADempiere.

I was just testing it locally, so if anyone would like to try a
more realistic test you can find me in the IRC channel or just
send me an email.

eike
```

A.1.30 Donnerstag 26.11.09

12:38 Uhr

Openbravo angeschrieben.

```
Hi,

I am working in the Institute of Computer Science at the Free
University of Berlin where we are creating ways to improve the
process of writing software, especially for Free/Open Source
Software developers.

Our institute develops Saros, an open source Eclipse plug-in that
allows two or more developers to work together in real-time
developing code, training newcomers, or performing peer reviews.
It is similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse. As part of my masters thesis I would like
to assess how useful Saros is when developing Free/Open Source
Software. I already was in contact with  to other projects and
got some good response.

It is believed that Saros is particularily well suited to the
following development scenarios:

* an arbitrary number of developers working together over a
  network on a common project
```

```
* distributed pair programming, with an author and an observer
  who are geographically dispersed
* transferring knowledge between developers and introducing new
  people to the project
* joint code reviews
* collaborative development where developers are aware of each other
and wish to switch easily between working together or independently

If you are interested, I would offer you to join your project for a
while to give you support while working with Saros. What do you think?

Kind regards, Eike Starkmann

For more information:

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

A.1.31 Sonntag 29.11.09

Gekürzter ADempiere Bug Day Mitschnitt.

11:40 Uhr

```
[11:40] <eike_> hi all
[11:41] <croo> hi eike
[11:41] <trifon> hello
[11:41] <red1> hi eike_  the pair programmer
[11:41] <eike_> he he
[11:43] <eike_> croo: Did you read my workaround for partial sharing in the forum?
[11:43] <croo> eike_, not yet Ike .. been away from my desk for the last few days
[11:44] <croo> will check it out
```

11:44 Uhr

AD4 spricht mich auf ein Problem mit Saros an.

```
[11:44] <genied> hi eike
[11:44] <eike_> hi genied
[11:45] <genied> eike: I installed your plugin but didn't managed to connect, I'm
behind a proxy that requires authentication
[11:48] <eike_> genied: Dit you enter the proxy date in the eclipse "network
connections"?
[11:55] <eike_> genied: so you weren't able to connect to the jabber server or to
you partner?
[12:05] <genied> eike: I can't connect to the server (talk.google.com)
[12:18] <eike_> genied: i think that we dont support proxy with auth yet. So there
is no way to connect to talk.google.com out of the box
[12:23] <genied> eike: thanks, neat feature anyway
```

15:08 Uhr

Nachfrage an AD2, ob er Saros benutzt

```
[15:08] <eike_> i wonder if somebody is actual working with Saros, i just see croo
online using Saros.
[15:09] <eike_> croo: are you in a session with someone, or alone?
[15:09] <croo> no eike... actually I am looking for one of my own bugs right now
[15:09] <croo> I trie dto setup with red1 to test
[15:09] <croo> but today he is busy too
[15:10] <croo> sorry ... will get there eventually though
[15:10] <croo> fear not
[15:10] <croo> and I will let you know how it works for us
[15:17] <eike_> croo: thanks. if you see me online when you meet with red1 just
send me a message, i would love to see you both work.
[17:31] <eike_> i gtg may all bugs be caugth
```

A.1.32 Montag 30.11.09

16:09 Uhr

AD1 kann hat Saros installiert, kann sich aber nicht mit einem Jabber-Server verbinden. (gekürzter Mitschnitt)

```
(16:09:25) eike1: back again
(16:09:41) red1: wb
(16:09:48) red1: ok my error msg is the same as before
(16:10:01) eike1: and that is?
(16:10:02) red1: complaining that i need to check again my login and pass
(16:10:18) eike1: hmm ok
(16:10:39) eike1: do you have another japper client?
(16:10:54) red1: how do i get that?
(16:11:06) eike1: you have mac, or?
(16:11:20) red1: mac
(16:11:22) eike1: as far as i remeber
(16:11:24) eike1: ok
(16:12:48) eike1: i just wanted to chek if you can connect to jabber.org in
generel or if this is an saros problem
(16:13:22) red1: good idea
(16:13:28) red1: so how do i go about doing that?
(16:14:19) eike1: audium
(16:14:24) eike1: is a mav jabber client
(16:14:26) eike1: http://adium.im/
(16:14:31) eike1: aduim i mean
(16:14:36) red1: do i install it in eclipse also?
(16:14:50) eike1: no ist an external program
(16:14:55) red1: ok on to that
(16:15:06) eike1: but i dont know it since im on linux
(16:15:26) red1: dl in 4 mins
(16:15:37) eike1: ok
(16:15:38) red1: a Mac dmg
(16:21:39) red1: half min
(16:28:19) red1: executing adium
(16:35:57) red1: eike1: ok adium is up
(16:36:04) red1: awaiting your orders
(16:38:30) eike1: ok you have it installed, they try to cennect with out accound
to jabber.org
(16:43:42) red1: eike1: i launched adium
(16:43:48) red1: it ask me to give account
(16:43:58) red1: ok jabber
(16:44:03) red1: understand
(16:44:41) red1: connecting
```

```
(16:44:45) eike1: yey your jabber accoun
(16:44:47) eike1: yea
(16:44:56) eike1: waht was it agein?
(16:45:00) eike1: again?
(16:45:35) eike1: you jabber account?
(16:45:39) eike1: your
(16:46:34) eike1: omg lots of typos
(16:46:35) red1: red1daniel
(16:46:43) red1: i m now online
(16:46:55) red1: its ok eike1 .. mistakes are holy trails
(16:47:01) red1: i mea ngrails
(16:47:12) eike1: he he
(16:47:21) eike1: red1: asking for auth
(16:47:49) red1: waiting by the adium
(16:48:00) eike1: hmm
```

16:48 Uhr

Ab1 installiert den Jabberclient Adium, um zu testen ob er sich dann mit dem Jabber-Server verbinden kann.

```
(16:48:32) red1: whats yours?
(16:48:40) eike1: try to at me eike.starkmann
(16:49:03) eike1: add
(16:50:02) eike1: on jabber.org
(16:50:36) red1: adding
(16:51:14) eike1: hmm nothing happens here
(16:51:23) red1: added
(16:51:25) red1: just did
(16:51:45) red1: i think physics take time
(16:51:50) red1: which law u using?
(16:52:08) eike1: which law?
(16:52:14) red1: u shuld migrate from Newton's to Planks
(16:52:19) eike1: he he
(16:52:23) eike1: now i got it
(16:52:26) red1: just joking
(16:52:40) red1: now pls send back the photons
(16:52:50) red1: i m sure they will disintegrate
(16:54:08) eike1: it cant take that long
```

```
(16:54:37) eike1: are you shure that you are connected?
(16:54:43) eike1: try to add croo
(16:54:50) eike1: perhaps its just mee
(16:54:51) eike1: me
(16:54:59) red1: u got me there?
(16:55:04) red1: can u msg me ?
(16:55:08) eike1: no
(16:55:08) red1: i cant see u yet
(16:55:25) eike1: me neighter still waiting for permission
(16:55:28) red1: is croo id?
(16:55:31) eike1: croo@jabber.org
(16:57:13) red1: installing growl
(16:58:14) eike1: ok its saiy that red1daniel doesnt exist on jabber.org
(16:58:32) red1: hmm
(16:58:39) red1: then how come i can login?
(16:58:45) red1: i am in jabber client
(16:59:56) red1: getting info about eike.starkmann
(17:00:58) eike1: ok jabber.org says that red1daniel alteady exists, so that is
correct
(17:05:16) eike1: greets
(17:05:56) eike1: hmm ok but you can connect to jabber.org so that means something
doesnt work with saros
(17:06:02) red1: eike1: i exist yet i do not talk? :D
(17:06:26) red1: the info search on eike.starkmann still awaiting
(17:06:34) red1: hmm.. there is sumtin wrong
(17:06:39) eike1: strange
(17:06:51) red1: ok lets leave physics alone for now :D
(17:07:24) eike1: its not physics, physics work!
(17:09:32) eike1: my client sais you are offline
(17:10:16) red1: ok i changed from available to away
(17:10:20) red1: just to hack it up
(17:10:38) red1: ok i got u
(17:10:43) red1: u know why i couldnt before?
(17:10:53) red1: and there is a chime to prove it eike1
(17:11:19) red1: it was hidden behind some window!!!
(17:11:31) red1: careless physician
(17:11:39) eike1: i got you too
(17:12:01) red1: ok u get the Novel Prize
```

17:17 Uhr

Unterhaltung wird über Jabber fortgesetzt.

(17:17:54) eike.starkmann@jabber.org/31504052011259596721967422: what is you poxy settings in saros?
(17:18:10) red1daniel@jabber.org: I remove the proxy
(17:18:14) red1daniel@jabber.org: i check the normal one
(17:18:25) red1daniel@jabber.org: maybe i shuld restart?
(17:18:49) eike.starkmann@jabber.org/31504052011259596721967422: yeah that is always a good idea in eclipse
(17:19:07) red1daniel@jabber.org: i select 'direction connection to the internet'
(17:19:34) red1daniel@jabber.org: ok restarting
(17:22:26) red1daniel@jabber.org: restarted and connecting..
(17:24:15) red1daniel@jabber.org: same error
(17:24:21) red1daniel@jabber.org: let me paste the snapshot
(17:24:29) eike.starkmann@jabber.org/31504052011259596721967422: yeah
(17:25:08) red1daniel@jabber.org: getting it?
(17:25:19) red1daniel@jabber.org: it ask which resource.. i select saros
(17:25:32) eike.starkmann@jabber.org/31504052011259596721967422: yes thats ok
(17:25:39) eike.starkmann@jabber.org/31504052011259596721967422: no i dont get
(17:25:45) eike.starkmann@jabber.org/31504052011259596721967422: perhpas per mail
(17:25:54) eike.starkmann@jabber.org/31504052011259596721967422: eike.starkmann@fu-berlin.de
(17:25:58) red1daniel@jabber.org: offering to send picture.png
(17:26:29) red1daniel@jabber.org: ok sending
(17:26:44) eike.starkmann@jabber.org/31504052011259596721967422: that always doesnt work, becaue im behind nat and firewall
(17:28:33) red1daniel@jabber.org: mail went thru
(17:28:41) red1daniel@jabber.org: gee its past midnite
(17:29:09) red1daniel@jabber.org: i got to retire.. to redeem my cells..
(17:29:15) eike.starkmann@jabber.org/31504052011259596721967422: ok
(17:29:17) red1daniel@jabber.org: u can mail me back any tots
(17:29:23) red1daniel@jabber.org: thanks for your effort
(17:29:29) red1daniel@jabber.org hat das Gespräch verlassen.
(17:29:49) eike.starkmann@jabber.org/31504052011259596721967422: ok wil discuss about this with the networ guys tomorrow
(17:29:53) eike.starkmann@jabber.org/31504052011259596721967422: sleep well
(17:30:01) red1daniel@jabber.org hat das Gespräch verlassen.

A.1.33 Dienstag 08.12.09

13:56 Uhr

Emails verschickt an die Maintainer von OpenProj.

```
Hi,

I am working in the Institute of Computer Science at the Free
University of Berlin where we are creating ways to improve the
process of writing software, especially for Free/Open Source
Software developers.

Our institute develops Saros, an open source Eclipse plug-in that
allows two or more developers to work together in real-time
developing code, training newcomers, or performing peer reviews.
It is similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.
As part of my masters thesis I would like to assess how useful
Saros is when developing Free/Open Source Software. I already was
in contact with other projects and got some good response.

As i see you are using agile methods, it is believed that Saros
is particularly well suited for agile development and to the
following development scenarios:

* an arbitrary number of developers working together over a
  network on a common project
* distributed pair programming, with an author and an observer
  who are geographically dispersed
* transferring knowledge between developers and introducing new
  people to the project
* joint code reviews
* collaborative development where developers are aware of each
  other and wish to switch easily between working together or
  independently

If you are interested, I would offer you to join your project
OpenProj for a while to give you support while working with Saros.
```

```
What do you think?

Kind regards, Eike Starkmann

For more information:

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
 https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

A.1.34 Montag 14.12.09

17:07 Uhr

Tomcat kontaktiert.

```
Hi,

I am working in the Institute of Computer Science at the Free
University of Berlin where we are creating ways to improve the
process of writing software, especially for Free/Open Source
Software developers.

Our institute develops Saros, an open source Eclipse plug-in that
allows two or more developers to work together in real-time
developing code, training newcomers, or performing peer reviews.
It is similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.
As part of my masters thesis I would like to assess how useful
Saros is when developing Free/Open Source Software. I already was
in contact with other projects and got some good response.

As i see you are using agile methods, it is believed that Saros
is particularly well suited for agile development and to the
following development scenarios:

* an arbitrary number of developers working together over a
```

```
  network on a common project
* distributed pair programming, with an author and an observer
  who are geographically dispersed
* transferring knowledge between developers and introducing new
  people to the project
* joint code reviews
* collaborative development where developers are aware of each
  other and wish to switch easily between working together or
  independently

If you are interested, I would offer you to join your project
Tomcat for a while to give you support while working with Saros.
What do you think?

Kind regards, Eike Starkmann

For more information:

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
 https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

A.1.35 Dienstag 12.01.10

16:39 Uhr

Eine Email an die Entwickler-Mailingliste von Hibernate.

```
Hi,

I am working in the Institute of Computer Science at the Free
University of Berlin where we are creating ways to improve the
process of writing software, especially for Free/Open Source
Software developers.

Our institute develops Saros, an open source Eclipse plug-in that
allows two or more developers to work together in real-time
```

```
developing code, training newcomers, or performing peer reviews.
It is similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.
As part of my masters thesis I would like to assess how useful
Saros is when developing Free/Open Source Software. I already was
in contact with other projects and got some good response.

As i see you are using agile methods, it is believed that Saros
is particularly well suited for agile development and to the
following development scenarios:

* an arbitrary number of developers working together over a
  network on a common project
* distributed pair programming, with an author and an observer
  who are geographically dispersed
* transferring knowledge between developers and introducing new
  people to the project
* joint code reviews
* collaborative development where developers are aware of each
  other and wish to switch easily between working together or
  independently

If you are interested, I would offer you to join your project
Hibernate for a while to give you support while working with Saros.
What do you think?

Kind regards, Eike Starkmann

For more information:

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
 https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

A.1.36 Mittwoch 12.01.10

10:37 Uhr

Eintrag im Spring Forum *Core Container* verfasst.

```
Hi,

I am working in the Institute of Computer Science at the Free
University of Berlin where we are creating ways to improve the
process of writing software, especially for Free/Open Source
Software developers.

Our institute develops Saros, an open source Eclipse plug-in
that allows two or more developers to work together in real-time
developing code, training newcomers, or performing peer reviews.
It is similar to Gobby, SubEthaEdit or Google Docs but focuses on
programming in Eclipse.
As part of my masters thesis I would like to assess how useful
Saros is when developing Free/Open Source Software. I already
was in contact with other projects and got some good response.

It is believed that Saros is particularly well suited to the
following development scenarios:

* an arbitrary number of developers working together over a
network on a common project
* distributed pair programming, with an author and an observer
  who are geographically dispersed
* transferring knowledge between developers and introducing new
  people to the project
* joint code reviews
* collaborative development where developers are aware of each
  other and wish to switch easily between working together or
  independently

If you are interested, I would offer you to join Spring for a
while to give you support while working with Saros.
What do you think?
```

```
Kind regards, Eike Starkmann

For more information:

Website: https://www.inf.fu-berlin.de/w/SE/DPP
Update Site: http://dpp.sf.net/update
Saros @ SF: http://sourceforge.net/projects/dpp/
Programming Languages Supported by Saros :
https://www.inf.fu-berlin.de/w/SE/DPPCompatiblePlugin
```

03:15 Uhr

Antwort erhalten von HI1.

```
Hey eike this sounds really interesting.  Pair programming is an
incredible development tool that in my exerience is not generally
used on OSS dev due to developers not being located close together.

We have been discussing using Crucible for Hibernate development for
exactly alot of the points you list.

Unfortunately I personally do not use eclipe so I myself am not a
good fit for this.
```

A.2 Kategoriensystem

1. Ablehnung
   a) Technische Ablehnung
      i. Kein sinnvoller Vorschlag
         - Es gibt schon so etwas

           „Not to rain on someone else's parade but you can do the same in Emacs:“ [DR2 A.1.18]
      ii. Mangelhaftes Produkt

- Defekt

  Sind nicht immer Ablehnungen, sondern auch Akzeptanzen, mit Defekt hinweisen, wie „Tolles Produkt, aber sollten nicht XY funktionieren?“
  „[...]I thought the typed text by the person is highlighted as well, no?“ [TY1 A.1.7]

- Abhängigkeit von einer Entwicklungsumgebung (Eclipse)

  „Since Netbeans is *awesome* when it comes to supporting PHP 5.3, I am no longer using Eclipse/PDT“ [TY4 A.1.12]
  „People don’t like to be dependent on a specific platform. Everyone has their own.“ [DR8 A.1.19]
  „At the moment I’m using Eclipse. Really, it is NOT a casual-use tool. Especially for Drupal, it’s not worth it unless you adopt it and take the time to customize it. I can’t see anyone using Eclipse just for Saros and other for everything else.“ [DR4 A.1.22]
  „Unfortunately I personally do not use eclipe so I myself am not a good fit for this.“ [HI1 A.1.36]

b) Konzeptionelle Ablehnung

   i. Keine sinnvolle Arbeitsmethode oder Prozess

   - Abneigungen gegen Pair Programming

     „[...]I do not really believe into pair programming.“ [TY6 A.1.9]
     „To me pair programming is more a distraction than creativity. I am ok to *discuss* how to create something but when it comes to coding I found that one person coding produces better results.“ [TY6 A.1.11]
     „The concept sounds nice but in practice I neve saw satisfying result.“ [TY6 A.1.9]
     „I’ve had to do this sort of thing before (not distributed,

but pair programming). It is, by far, the worst idea ever conceived by man(agers).[...]
The reality of pair programming is that one guy is better than the other guy, and spends half his time explaining the problem and solution to the other guy, instead of simply coding the damn thing. The lesser of the two programmers only drags the other guy down to his level, and wastes both of their time.“ [WP1 A.1.18]
„However, if the intent is to produce clean, bug-free code in a fast and efficient manner, what Otto pointed out is true in many/most situations: you’re going to handcuff a gifted programmer and produce worse code, slower.“ [WP2 A.1.23]

c) Umfeld bezogene Ablehnung.

   i. Produkt unpassend für das Projekt.

   - Fehlende Funktionalität

     „I guess most of us really need PDT in order to try it out properly :-)“ [TY1 A.1.1]

   - Muss jeder selber entscheiden.

     „[...]there is no Drupal Team that could decide en masse to use a tool like this.“ [DR4 A.1.18]

   - Es wird schon so etwas benutzt

     „Having said that the irc paste combo is used exactly in the manner you suggest.“ [DR8 A.1.19]

   - Zu wenige Entwickler würden das benutzen.

     „I think there are people who can do it but it is rare.“ [TY6 A.1.9]
     „Some other module / core contributors might find it useful for those tasks, but it doesn’t seem like a common need.“

[DR5 A.1.18]
„It might work in some cases, but I suspect it won't reach wide adoption.“ [DR5 A.1.19]

- Zu große Einarbeitungszeit

  „Time is very tight, so any delays messing with connection problems, etc. aren't really acceptable. So, both parties would need to have already used it in the past and frequently for it to be usable during crunch time.“ [DR5 A.1.19]

- Produkt könnte nützlich sein aber nicht für uns.

  „Your project sounds really interesting. Nevertheless we do not think it would be of great use for our workflows.“ [TYS1 A.1.24]
  „I probably won't ever use something like Saros unless I got a tele-working position somewhere.“ [DR5 A.1.18]
  „The tool is usable, in some limited fashion - project teams etc... It is not a Drupal tool“ [DR8 A.1.19]

ii. Konzept unpassend für das Projekt.

- Entwickler arbeiten zu asynchron.

  „[...]in such an open-source project the working times will be very, very different so that it might be difficult to for teams to find together.“ [TY2 A.1.1]
  „Drupal is so distributed that we all have our own development workflows using a variety of tools, often in our basements rather than our offices“ [DR4 A.1.18]
  „Most of the Drupal development that I've seen is either: A) Written by one person, then reviewed / improved asynchronously “ [DR5 A.1.18]
  „[...]most of the development does not occur in parallel by multiple developers in real time.“[FR1 A.1.18]
  „But anything which requires live interaction between devel-

opers won't work, we all work at different times of day."[FR2 A.1.20]
„[...]this sort of thing is probably more useful in environments where people are in close physical and temporal proximity to each other (eg. an office)."[FR1 A.1.18]

- Entwickler arbeiten ohnehin in einem Büro zusammen.

  „Most of the Drupal development that I've seen is either: [...] B) Written by a team of people for a project in-house, in which case it would be the same as basically any other in-house PHP development" [DR5 A.1.18]

- Konzept könnte nützlich sein aber nicht für uns/OSSP.

  „Pair programming is an incredible development tool that in my exerience is not generally used on OSS dev due to developers not being located close together." [HI1 A.1.36]

2. Akzeptanz
   a) Technische Akzeptanz
      i. Sinnvolles Produkt
         - Interesse/sinnvoller Vorschlag

           „Yep, I'll give it a try soon after my holidays!" [TY1 A.1.1]
           „I would be happy to try it with somebody[...]" [TY2 A.1.1]
           „That sounds quite cool :-)" [TY1 A.1.1]
           „Sounds like a very cool project!" [TY4 A.1.3]
           „Ok, I'll give it a try[...]" [TY4 A.1.3]
           „Sounds very interesting :)" [DR2 A.1.18]
           „Looking forward to try it again :)" [TY1 A.1.7]
           „That definitely sounds interesting." [JU1 A.1.25]
           „That said, I'm sure such a tool would prove useful to certain developers if they choose to use it." [DR4 A.1.18]

„Eike, this sounds pretty neat.“ [DR7 A.1.19]
„clearly Saros looks revolutionary“ [AD1 A.1.27]
„Hey eike this sounds really interesting.“ [HI1 A.1.36]

- Anwendungsvorschlag

  „It would also be a good tool for training and consulting,where you could save a trip to the customer's office and work with them on code directly.“[DR7 A.1.19]
  „but if lets say 5 can work on it and the rest can just view it, it can be a great teaching tool [...] and process of distributing knowlege increases“ [AD1 A.1.27]

- Vorschläge für Weiterentwicklung

  „if it can be stored for later replay.. awesome possibilities“ [AD1 A.1.27]
  „can the synch be reduced to .java files? [...] and xml files?“ [AD3 A.1.27]
  „it woudl be nice if you could store & retrieve the partial project selected so that each time you would not hav eto manually unselect bunches of the project“ [AD2 A.1.27]

- Schon nach so etwas gesucht

  „I was just looking for something like this!“ [DR6 A.1.18]
  „We have been discussing using Crucible for Hibernate development for exactly alot of the points you list.“ [HI1 A.1.36]

- Gute Lösung (Saros getetest)

  „I think it can be a quite useful tool,[...]“ [TY2 A.1.1]
  „It seemed to work, and it looks like a really nice approach!“ [TY1 TY1-31.08.09-14.16]
  „ok - work pretty snappy too a great teaching tool! never mind pair programming! very good guys ... well done“ [AD2 A.1.27]

b) Konzeptionelle Akzeptanz

i. Sinnvolle Arbeitsmethode oder Prozess

- Produktivitätssteigerung

  „I've found real" pair programming (two coders sitting in front of the same physical computer) a lot of fun an really productive because...
  1. the coding was generally faster: we could help each other, move the keyboard over when someone was stuck, and the social pressure" kept our focus on the task
  2. it saved us all the time-consuming code review/new version/code review etc. cycles
  3. it helps knowledge spreading between the coders"' [TY7 A.1.10]

- Sonstiges

  „the idea seems great.“ [AD3 A.1.27]

c) Umfeld bezogene Akzeptanz

i. Produkt passend für das Projekt

- Generell von Nutzen für das Projekt

  „As far as Drupal goes, I think that something like this would be particularly useful for core in a time like...oh,say....now.“ [DR6 A.1.18]

- Anwendungsvorschlag für das Projekt

  „With code freeze 38 hours away, efficiency is key to getting the patches from the exception list into core. If more than one person is working on a patch, it really makes sense to use something like this [...] and have everyone looking at the same code at the same time so that it's not a constant string of upload to the issue queue, have a minor bug pointed out, reroll with bug fixes. Two sets of eyes on any of those

patches at any one time would be awesome, and I think it would help get them ready to be committed.“ [DR6 A.1.18]
„This is exactly what i find myself doing with my developers that are new to the "drupal wayÏ plan on installing this today and integrating it into my eclipse workflow for a test run. [...] Its a novel solution to an old problem of communicating via line numbers :)“ [DR9 A.1.19]
„we do have a serious peer review constraint our trunk has to be well guarded and yet we want faster evolution with some Saros, i can imagine more closer devs insight into how each other work can help the goodwill“ [AD1 A.1.27]

3. Unklarheiten
   a) Technische Nachfragen
      i. Installation

         „[...]Where on http://www.eclipse.org/pdt/downloads/ can I download the all-in-one build[...]“ [TY1 A.1.6] „Is the plug-in ready to go, or is there something I should know first?“ [JU1 A.1.25]

      ii. Performance
         - Bandbreite

           „What’s the overhead like? Would it be smaller than that of VNC[...]“ [DR2 A.1.18]
           „hows the bandwidth requirement?“ [AD1 A.1.27]

      iii. Kompatibilität
         - Betriebssystem

           „so it can work on Mac?“ [AD1 A.1.27]

         - Versionsverwaltungen

„When both parties save an edited file, does it get stored in the local workspace as usual? Wouldn't it cause conflicts if you are using version control?„ [DR2 A.1.18]

- Programmiersprachen

  „Is it language-dependent?“ [DR4 A.1.18]

iv. Verwendete Technologien

„is it done from scratch or built upon an earlier idea?“ [AD1 A.1.27]

v. Produktvergleich

- Zusammenhang mit andern Produkten

  „Is it something like Bespin?“ [TY3 A.1.2]
  „How does it relate to: http://wiki.eclipse.org/DocShare_Plugin ?“ [TY1 A.1.1]
  „This sounds very similar to http://codesurgeonblog.com/2008/06/cola-real-time-shared-editing.html which was demonstrated last year. How is Saros related to that effort?“ [DR3 A.1.18]

- Sonstige

  „what can be the max users?“ [AD1 A.1.27]
  „can there be a read only mode spawnyd ?“ [AD1 A.1.27]
  „reading about Mumble part hmm it has voice?“ [AD1 A.1.27]
  „when you share a project ... you get a user list ... where does that list come from? because mine is empty“ [AD2 A.1.27]
  „lso once you share a project how do stop sharing?“ [AD2 A.1.27]
  „so must you have the same project open as me to share?“ [AD2 A.1.27]

„does it look at timestamps or revision numbers or something like that?“ [AD2 A.1.27]

b) Konzeptionelle Nachfragen

i. Verständigung

„I wonder how the communication in DPP works if you cannot talk to each other.“ [TY7 A.1.10]

ii. Effektivität

„did u try calculating the impact of using such tool? i mean positive rate of productivity?“ [AD1 A.1.27]

iii. Sonstige

„wonder if this is good for 'pair programming' ?“ [AD1 A.1.27]
„what are you guys plans for this? will it remain in the unviersity?“ [AD1 A.1.27]
„what are you other motivation for such an addon?“ [AD1 A.1.27]
„how did that idea first appear?“ [AD1 A.1.27]
„so how many devs u have working on this?“ [AD1 A.1.27]

Literaturverzeichnis

[ADE10a] ADEMPIERE.COM.: *ADempiere Bug Day.* Version: 2010. http://www.adempiere.com/index.php/ADempiere_Bug_Day. – Online–Ressource, Abruf: 27. Januar 2010

[ADE10b] ADEMPIERE.COM.: *QAStart.* Version: 2010. http://www.adempiere.com/index.php/QAStart. – Online–Ressource, Abruf: 27. Januar 2010

[ADE10c] ADEMPIERE.DE.: *Adempiere Deutschland e.V. gegründet.* Version: 2010. http://adempiere.de/ADempiere-e-V.adempiere-ev.0.html. – Online–Ressource, Abruf: 27. Januar 2010

[AGI09] AGILEMANIFESTO.ORG: *Manifesto for Agile Software Development.* Version: 2009. http://agilemanifesto.org/. – Online–Ressource, Abruf: 18. November 2009

[Ale10] Alex Fulton, Christine Black: *BBC Press Office.* Version: 2010. http://www.bbc.co.uk/pressoffice/bbcworldwide/worldwidestories/pressreleases/2007/08_august/america_theoffice.shtml. – Online–Ressource, Abruf: 13. Februar. 2010

[APA10a] APACHE.ORG: *Apache Software Foundation.* Version: 2010. http://www.apache.org/. – Online–Ressource, Abruf: 16. Januar 2010

[APA10b] APACHE.ORG: *Apache Tomcat.* Version: 2010. http://tomcat.apache.org/index.html. – Online–Ressource, Abruf: 16. Januar 2010

[APA10c] APACHE.ORG: *The Foundation.* Version: 2010. http://www.apache.org/foundation/. – Online–Ressource, Abruf: 16. Januar 2010

[APA10d] APACHE.ORG: *Heritage.* Version: 2010. http://tomcat.apache.org/whoweare.html. – Online–Ressource, Abruf: 16. Januar 2010

[APA10e] APACHE.ORG: *How it works.* Version: 2010. http://www.apache.org/foundation/how-it-works.html. – Online–Ressource, Abruf: 16. Januar 2010

[BC07] Bauer C., King G.: *Java Persistence mit Hibernate.* HANSER, München, 2007

[Bec99] Beck, K.: *Extreme Programming Explained: Embrace Change.* Addison-Wesley, 1999

[Bec02] Beck, K: *Test Driven Development by Example.* Addison-Wesley Verlag, Amsterdam, 2002

[Bra04] Brand, A.: *Fallstudie horizontales elektronisches Arbeitsnetz/ Open Source- Projekt.* Version: 2004. http://nbn-resolving.de/urn:nbn:de:0168-ssoar-65917. – Online–Ressource, Abruf: 27. Januar 2010

[Cla99] Clarke, I.: *A Distributed Decentralised Information Storage and Retrieval System*, Division of Informatics University of Edinburgh, Diplomarbeit, 1999

[CW01] Cockburn, Alistair ; Williams, Laurie: The costs and benefits of pair programming. (2001), S. 223–243

[DD06] Denning, Peter J. ; Dunham, Robert: Innovation as language action. In: *Commun. ACM* 49 (2006), Nr. 5, S. 47–52. http://dx.doi.org/http://doi.acm.org/10.1145/1125944.1125974. – DOI http://doi.acm.org/10.1145/1125944.1125974

[Dje06] DJEMILI, R.: *Entwicklung einer Eclipse-Erweiterung zur Realisierung und Protokollierung verteilter Paarprogrammierung.*, Software Engineering Freie Universität Berlin, Diplomarbeit, 2006

[dru09a] DRUPAL.ORG: *Drupal history seen by Dries.* Version: 2009. `http://drupal.org/node/297669`. – Online–Ressource, Abruf: 21. Oktober 2009

[dru09b] DRUPAL.ORG: *Features, mission, and principles.* Version: 2009. `http://drupal.org/getting-started/project-features`. – Online–Ressource, Abruf: 21. Oktober 2009

[Fab10] FABIÁN E. BUSTAMANTE: *Ono plugin for the Vuze.* Version: 2010. `http://www.aqualab.cs.northwestern.edu/projects/Ono.html`. – Online–Ressource, Abruf: 12. Februar. 2010

[fre09a] FREENETPROJECT.ORG: *People.* Version: 2009. `http://freenetproject.org/people.html`. – Online–Ressource, Abruf: 05. November 2009

[fre09b] FREENETPROJECT.ORG: *The Philosophy beerhind Freenet.* Version: 2009. `http://freenetproject.org/philosophy.html`. – Online–Ressource, Abruf: 05. November 2009

[GIT10] GITHUB.COM: *JUnit Impact.* Version: 2010. `http://github.com/KentBeck/junit/graphs/impact`. – Online–Ressource, Abruf: 12. Januar 2010

[Gun86] GUNZ, J.: *Handlungsforschung: Vom Wandel der distanzierten zur engagierten Sozialforschung.* Braunmülller, Wien, 1986

[HEI10] HEISE.DE: *VMware kauft Enterprise Java Spezialisten SpringSource.* Version: 2010. `http://www.heise.de/newsticker/meldung/VMware-kauft-Enterprise-Java-Spezialisten-SpringSource-750685.html`. – Online–Ressource, Abruf: 27. Januar 2010

[HIB10] HIBERNATE.ORG: *Relational Persistence for Java and .NET.* Version: 2010. https://www.hibernate.org/. – Online–Ressource, Abruf: 18. Januar 2010

[Hub08] HUBMANN, A.: *ThreadVis for Thunderbird*, Graz University of Technology, Diplomarbeit, 2008

[JBO10] JBOSSS.ORG: *The hibernate-dev Archives.* Version: 2010. http://lists.jboss.org/pipermail/hibernate-dev/. – Online–Ressource, Abruf: 18. Januar 2010

[Joh02] JOHNSON, R.: *Expert One-On-One J2EE Design and Development.* wrox Press, 2002

[Ker03] KERR, Bernard: THREAD ARCS: An Email Thread Visualization. In: *Information Visualization, IEEE Symposium on* 0 (2003), S. 27. http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/INFVIS.2003.1249028. – DOI http://doi.ieeecomputersociety.org/10.1109/INFVIS.2003.1249028

[Kle82] KLEINING, G: Das qualitative Experiment. In: *Kölner Zeitschrift für Soziologie und Sozialpsychologie* 34 (1982), S. 224–253

[KSL03] KROGH, Georg von ; SPAETH, Sebastian ; LAKHANI, Karim R.: Community, joining, and specialization in open source software innovation: a case study. In: *Research Policy* 32 (2003), July, Nr. 7, 1217-1241. http://ideas.repec.org/a/eee/respol/v32y2003i7p1217-1241.html

[MAT09] MATT MULLENWEG: *The blogging software dilemma.* Version: 2009. http://ma.tt/2003/01/the-blogging-software-dilemma/. – Online–Ressource, Abruf: 03. November 2009

[May02] MAYRING, P.: *Einführung in die qualitative Sozialforschung.* Beltz Verlag, Weinheim und Basel, 2002

[Nos98] Nosek, John T.: The case for collaborative programming. In: *Commun. ACM* 41 (1998), Nr. 3, S. 105–108. http://dx.doi.org/http://doi.acm.org/10.1145/272287.272333. – DOI http://doi.acm.org/10.1145/272287.272333

[OHL10a] OHLOH.NET: *Eclipse Checkstyle Plugin.* Version: 2010. http://www.ohloh.net/p/eclipse-cs. – Online–Ressource, Abruf: 12. Januar 2010

[OHL10b] OHLOH.NET: *News.* Version: 2010. http://www.ohloh.net/p/eclipse-cs/rss_articles. – Online–Ressource, Abruf: 12. Januar 2010

[OP07] Oezbek, Christopher ; Prechelt, Lutz: On Understanding How to Introduce an Innovation to an Open Source Project. In: *FLOSS '07: Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development.* Washington, DC, USA : IEEE Computer Society, 2007, S. 12

[OPE09a] OPENBRAVO.COM.: *History.* Version: 2009. http://web.archive.org/web/20080612063754/http://www.openbravo.com/about-us/our-company/history/. – Online–Ressource, Abruf: 18. November 2009

[OPE09b] OPENBRAVO.COM.: *Openbravo.* Version: 2009. http://www.openbravo.com/about-us/. – Online–Ressource, Abruf: 16. November 2009

[OPE09c] OPENBRAVO.COM.: *Openbravo ERP:Legal.* Version: 2009. http://www.openbravo.com/legal/product/erp/. – Online–Ressource, Abruf: 16. November 2009

[OPE09d] OPENBRAVO.COM.: *Openbravo POS:Legal.* Version: 2009. http://www.openbravo.com/legal/product/pos/. – Online–Ressource, Abruf: 16. November 2009

[OPE10a] OPENBRAVO.ORG: *Universities.* Version: 2010. http://wiki.openbravo.com/wiki/Universities. – Online–Ressource, Abruf: 11. Januar 2010

[OPE10b] OPENPROJ.ORG: *OpenProj .* Version: 2010. http://openproj.org/openproj. – Online–Ressource, Abruf: 15. Januar 2010

[Ray99] RAYMOND, E: *Die Kathedrale und der Basar.* Version: 1999. http://gnuwin.epfl.ch/articles/de/Kathedrale/. – Online–Ressource, Abruf: 27. Januar 2010

[Ray10] RAYMOND, E: *Goodbye, „free software"; hello, „open source".* Version: 2010. http://www.catb.org/~esr/open-source.html. – Online–Ressource, Abruf: 27. Januar 2010

[RED10] RED1.ORG.: *Debate - Has Compiere Become Closed? Do We Fork? How?* Version: 2010. http://red1.org/forum/viewtopic.php?t=931&postdays=0&postorder=asc&start=0. – Online–Ressource, Abruf: 27. Januar 2010

[Roß07] ROSSNER, A.: *Empirisch-qualitative Exploration verschiedener Kontaktstrategien am Beispiel der Einführung von Informationsmanagement in OSS-Projekten*, Software Engineering Freie Universität Berlin, Diplomarbeit, 2007

[Sch02] SCHODER, D: *Peer-to-peer: Ökonomische, technologische und juristische Perspektiven.* Springer-Verlag Berlin, Heidelberg, 2002

[SER10] SERENA.COM: *Agile on Demand.* Version: 2010. http://www.serena.com/products/agile-software/. – Online–Ressource, Abruf: 15. Januar 2010

[SOU09] SOURCEFORGE.NET: *About.* Version: 2009. http://sourceforge.net/about. – Online–Ressource, Abruf: 27. Januar 2010

[SOU10a] SOURCEFORGE.NET.: *ADempiere ERP Business Suite Mailing lists.* Version: 2010. http://sourceforge.net/mail/?group_id=176962. – Online–Ressource, Abruf: 27. Januar 2010

[SOU10b] SOURCEFORGE.NET.: *Adempiere Members.* Version: 2010. http://sourceforge.net/project/memberlist.php?group_id=176962. – Online–Ressource, Abruf: 27. Januar 2010

[Sou10c] SOURCEFORGE.NET: *Azureus / Vuze.* Version: 2010. http://sourceforge.net/projects/azureus/. – Online–Ressource, Abruf: 12. Februar. 2010

[SOU10d] SOURCEFORGE.NET: *eclipse-cs development environment.* Version: 2010. http://eclipse-cs.sourceforge.net/development.html. – Online–Ressource, Abruf: 12. Januar 2010

[SOU10e] SOURCEFORGE.NET: *Eclipse-cs Tracker.* Version: 2010. http://sourceforge.net/tracker/?group_id=80344. – Online–Ressource, Abruf: 12. Januar 2010

[SOU10f] SOURCEFORGE.NET: *Email Archive: springframework-developer.* Version: 2010. http://sourceforge.net/mailarchive/forum.php?set=custom&viewmonth=&viewday=&forum_name=springframework-developer&style=ultimate&max_rows=100&submit=Change+View. – Online–Ressource, Abruf: 27. Januar 2010

[SOU10g] SOURCEFORGE.NET: *Features.* Version: 2010. http://sourceforge.net/register-project/features.phpt. – Online–Ressource, Abruf: 27. Januar 2010

[SOU10h] SOURCEFORGE.NET: *Forum Traffic for Contributing to Openbravo ERP Statistics.* Version: 2010. http://sourceforge.net/project/stats/detail.php?group_id=162271&ugn=openbravo&type=forum&mode=year&year=2009&forum_id=870683. – Online–Ressource, Abruf: 11. Januar 2010

[SOU10i] SOURCEFORGE.NET: *Forum Traffic for Developers Statistics.* Version: 2010. http://sourceforge.net/project/stats/detail.php?group_id=162271&ugn=openbravo&type=forum&mode=year&year=2009&forum_id=549512. – Online–Ressource, Abruf: 11. Januar 2010

[SOU10j] SOURCEFORGE.NET.: *Forum Traffic Statistics.* Version: 2010. https://sourceforge.net/project/stats/detail.php?group_id=176962&ugn=adempiere&type=forum&mode=year&year=2009&forum_id=0. – Online–Ressource, Abruf: 27. Januar 2010

[SOU10k] SOURCEFORGE.NET: *Gail Badner.* Version: 2010. http://dbradby.users.sourceforge.net/. – Online–Ressource, Abruf: 18. Januar 2010

[SOU10l] SOURCEFORGE.NET: *Hibernate-Members.* Version: 2010. http://sourceforge.net/project/memberlist.php?group_id=40712. – Online–Ressource, Abruf: 18. Januar 2010

[SOU10m] SOURCEFORGE.NET: *JUnit Email Archiv.* Version: 2010. http://sourceforge.net/mailarchive/forum.php?forum_name=junit-devel. – Online–Ressource, Abruf: 12. Januar 2010

[SOU10n] SOURCEFORGE.NET: *OpenProj-Mailarchive.* Version: 2010. http://sourceforge.net/mailarchive/forum.php?forum_name=openproj-developers. – Online–Ressource, Abruf: 15. Januar 2010

[SOU10o] SOURCEFORGE.NET: *OpenProj-Members.* Version: 2010. http://sourceforge.net/project/memberlist.php?group_id=199315. – Online–Ressource, Abruf: 15. Januar 2010

[Sou10p] SOURCEFORGE.NET: *Project Admins.* Version: 2010. http://sourceforge.net/projects/azureus/develop. – Online–Ressource, Abruf: 12. Februar. 2010

[SOU10q] SOURCEFORGE.NET: *Project Members.* Version: 2010. http://eclipse-cs.sourceforge.net/developers.html. – Online–Ressource, Abruf: 12. Januar 2010

[SOU10r] SOURCEFORGE.NET: *Spring Framework Members.* Version: 2010. http://sourceforge.net/project/memberlist.php?group_id=73357. – Online–Ressource, Abruf: 27. Januar 2010

[Sou10s] SOURCEFORGE.NET: *Top Downloads.* Version: 2010. http://sourceforge.net/top/topalltime.php?type=downloads. – Online–Ressource, Abruf: 12. Februar. 2010

[SPR10a] SPRINGSOURCE.ORG: *Spring - A Managers Overview.* Version: 2010. http://www.springsource.com/files/Spring_A_Managers_OverviewVer05.pdf. – Online–Ressource, Abruf: 27. Januar 2010

[SPR10b] SPRINGSOURCE.ORG: *Spring Community Forums.* Version: 2010. http://forum.springsource.org/index.php. – Online–Ressource, Abruf: 27. Januar 2010

[SPR10c] SPRINGSOURCE.ORG: *Spring Community Forums: Core Container.* Version: 2010. http://forum.springsource.org/forumdisplay.php?f=26&order=desc&page=18. – Online–Ressource, Abruf: 27. Januar 2010

[Sta10a] STALLMAN, R: *Kategorien freier und unfreier Software.* Version: 2010. http://www.gnu.org/philosophy/categories.de.html. – Online–Ressource, Abruf: 27. Januar 2010

[Sta10b] STANGL, W.: *Handlungsforschung.* Version: 2010. http://arbeitsblaetter.stangl-taller.at/FORSCHUNGSMETHODEN/Handlungsforschung.shtml. – Online–Ressource, Abruf: 27. Januar 2010

[TYP09a] TYPO3 Association: *Core-Development Mailinglist Rules*. Version: 2009. http://typo3.org/teams/core/core-mailinglist-rules/. – Online–Ressource, Abruf: 04. September. 2009

[TYP09b] TYPO3 Association: *Developing TYPO3 v4.3*. Version: 2009. http://forge.typo3.org/repositories/stats/typo3v4-core. – Online–Ressource, Abruf: 07. Dezember. 2009

[TYP09c] TYPO3 Association: *The History of TYPO3*. Version: 2009. http://www.typo3.com/History.1268.0.html. – Online–Ressource, Abruf: 18. August. 2009

[TYP09d] TYPO3 Association: *How are Security Issues handled?* Version: 2009. http://typo3.org/teams/security/incident-handling/. – Online–Ressource, Abruf: 07. September. 2009

[TYP09e] TYPO3 Association: *Licenses*. Version: 2009. http://typo3.org/about/licenses/. – Online–Ressource, Abruf: 18. August. 2009

[TYP09f] TYPO3 Association: *Mailing list rules & guidelines*. Version: 2009. http://typo3.org/community/mailing-lists/mailing-list-rules-guidelines/. – Online–Ressource, Abruf: 04. September. 2009

[TYP09g] TYPO3 Association: *Security in TYPO3*. Version: 2009. http://typo3.org/teams/security/. – Online–Ressource, Abruf: 07. September. 2009

[TYP09h] TYPO3 Association: *TYPO3 Extension Security Policy version 1.0*. Version: 2009. http://typo3.org/teams/security/extension-security-policy/. – Online–Ressource, Abruf: 07. September. 2009

[Ull08] Ullenboom, C: *Java ist auch eine Insel*. Galileo Computing, 2008

[Vuz10] VUZE, INC.: *Management Team.* Version: 2010. http://www.vuze.com/corp/management.php. – Online–Ressource, Abruf: 12. Februar. 2010

[Wil00] WILLIAMS, L.: *The Collaborative Software Process*, University of Utah, Diss., 2000

[WO05] WEST, Joel ; O'MAHONY, Siobhán: Contrasting Community Building in Sponsored and Community Founded Open Source Projects. In: *Hawaii International Conference on System Sciences* 7 (2005), S. 196c. http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/HICSS.2005.166. – DOI http://doi.ieeecomputersociety.org/10.1109/HICSS.2005.166

[WOR09a] WORDPRESS.ORG.: *Contributing to WordPress.* Version: 2009. http://codex.wordpress.org/Contributing_to_WordPress. – Online–Ressource, Abruf: 03. November 2009

[WOR09b] WORDPRESS.ORG.: *WordPress Download Counter.* Version: 2009. http://wordpress.org/download/counter/. – Online–Ressource, Abruf: 03. November 2009