

# Masterarbeit am Institut für Informatik der Freien Universität Berlin ${\bf Arbeitsgruppe~Software~Engineering}$

# Verbesserung der Usability von Saros unter Verwendung eines User-Centered Design Ansatzes

Maria Spiering maria.spiering@fu-berlin.de

Berlin, 11. Oktober 2012

Gutachter: Prof. Dr. Lutz Prechelt

Zweitgutachterin: Prof. Dr. Claudia Müller-Birn

Betreuerin: Julia Schenk

### Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, 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.

11. Oktober 2012

Maria Spiering

### Zusammenfassung

Diese Arbeit beschreibt wie die Gebrauchstauglichkeit (Usability) eines Werkzeuges zur verteilten Programmierung (Saros) anhand eines nutzerorientierten Gestaltungsprozesses verbessert wird. Dafür wird das implizite Wissen der Benutzer von Zusammenarbeit im Allgemeinen und den Saros-Konzepten im Speziellen untersucht.

Für die Spezifikation des Nutzungskontexts von Saros werden potentielle Benutzergruppen bestimmt und Anwender aus den Bereichen "Arbeit" und "Freizeit" als Zielgruppen definiert. Zur Erhebung des impliziten Wissens (mentales Modell) werden verschiedene Methoden vorgestellt und hinsichtlich ihrer Eignung zur Erhebung des allgemeinen Verständnisses von Zusammenarbeit und der Untersuchung der Saros-Konzepte bewertet.

Das mentale Modell der Zielgruppen wird anhand von Interviews erfasst und mit den Funkionalitäten des Werkzeugs verglichen. Der Vergleich erfolgt in Form einer grafischen Darstellung des mentalen Modells<sup>1</sup>. Zudem wird mittels handlungsorientierter Fragen das Verständnis der Saros-Konzepte ermittelt, sodass ingesamt 34 Usability-Probleme identifiziert werden konnten.

Abschließend werden 9 der Probleme behoben und anhand einer Online-Umfrage evaluiert. Für die verbleibenden Probleme werden geeignete Lösungsansätze vorgeschlagen.

<sup>&</sup>lt;sup>1</sup>Beim Diagramm des mentalen Modells nach Young (2008) [1] handelt es sich um eine Darstellungsform, die das Anwenderverhalten und die Funktionalität der Software gegenübergestellt.

# Inhaltsverzeichnis

1	Einleitung	1
	1.1 Saros	1
	1.2 Motivation	2
	1.3 Zielsetzung und Fokus	3
	1.4 Aufbau der Arbeit	3
2	Grundlagen	5
	2.1 Eclipse	5
	2.2 Saros	5
	2.3 Mentale Modelle	12
	2.4 Usability	13
	2.5 User-Centered Design	16
3	Methoden zur Erhebung von Mentalen Modellen	21
	3.1 Inspektion	21
	3.2 Instance-based cognitive Mapping	22
	3.3 Struktur-Lege-Technik	24
	3.4 Interviews	24
	3.5 Auswahl der Evaluationsmethode	28
4	Erhebung	31
	4.1 Vorbereitung	31
	4.2 Durchführung	46
5	Auswertung	<b>51</b>
	5.1 Abstrakte Ebene	51
	5.2 Zielgruppe	62
	5.3 Konkrete Ebene	63
	5.4 Zusammenfassung und Priorisierung der Usability-Probleme	68
6	Behandlung und Behebung der Usability-Probleme	72
	6.1 Arbeitspaket "Getting Started"	72
	6.2 Arbeitspaket "Follow Mode"	80
	6.3 Arbeitspaket "Saros-View"	82
	6.4 Arbeitspaket "Send File"	85
	6.5 Andere Arbeitspakete	88
7	Evaluation	90
8	Fazit	95
A	Intonvious	00
A	Interviews	98
	A.1 Vorbereitung der Interviews	98
	A.2 Ergebnisse der Interviews	108
$\mathbf{B}$	Arbeitspakete	<b>139</b>

<ul><li>B.1 Arbeitspaket "Getting Started</li><li>B.2 Arbeitspaket "Saros-View" .</li></ul>													
Literaturverzeichnis													151

# Abbildungsverzeichnis

1	Die Eclipse Workbench	6
2	Das 3K-Modell für Groupware	8
3	Saros-annotierter Screenshot mit Awareness-Information	10
4	Darstellung der Awareness-Informationen im Package Explorer	11
5	Follow-Modus und Host-Konzept im Saros-View	11
6	Die drei Hauptaspekte der Usability (nach ISO 9241)	13
7	Aktivitäten des ISO 13407 Entwicklungsprozesses	17
8	UCD-Methoden dieser Arbeit	20
9	Mentales Modell einer webbasierten Such- und Orientierungsplattform	25
10	Beispiel einer Frame-basierten Struktur	26
11	Mentales Modell von Reaktionen auf Computer-Sicherheitshinweise	27
12	Mentales Modell zum Verhalten nach dem Aufstehen	29
13	Beispiel für ein Produkt, das für zu viele Anwender konstruiert wurde	32
14	Tabellarische Übersicht über die Saros-Konzepte	43
15	Überblick über die Versuchspersonen	47
16	Zeitplan der Interview-Durchführung	49
17	Task-Typen	52
18	Gruppierung von Atomic Tasks in Tasks	54
19	Gruppierung von Atomic Tasks in Tasks (Beispiel)	54
20	Gruppierung von Tasks in Tower und Mental Spaces	55
21	Gruppierung von Tasks in Tower und Mental Spaces (Beispiel)	56
22	Das mentale Modell der abstrakten Ebene (Auszug)	58
23	Beispiel einer Problembeschreibung (abstrakte Ebene)	60
24	Beispiel einer Problembeschreibung (konkrete Ebene)	67
25	Problem-Verteilung nach Ebene	68
26	Problem-Verteilung nach Fatalität	68
27	Problem-Verteilung der abstrakten Ebene nach dem 3K-Modell	69
28	Vergleich des Priorisierungsprinzips	70
29	Priorisierung nach Fatalität und Machbarkeit	70
30	Ausschnitt aus dem Storyboard des Host-Comics	74
31	Ausschnitt aus dem Host-Comic	75
32	Viewport-Annotation bei SubEthaEdit	76
33	Veränderungen seit der letzten Aktualisierung der Einführungsseite	<b>7</b> 9
34	Die neue Einführungsseite	80
35	Vergleich der dargestellten Status-Informationen des Follow-Modus	82
36	Fehlende Farbkachel bei Selektierung eines Teilnehmers	83
37	Verbesserung des Farbkontrast	84
38	Vergleich der verwendeten Betriebssysteme (nach Saros-Sitzungen)	84
39	Darstellung des Saros-Views bei Mac OS X und Windows 7	85
40	Einstellung der Saros-Lieblingsfarbe	89
41	Antworten aller Teilnehmer zum Host-Comic	93
42	Antworten der Teilnehmer zum Host-Comic nach Saros-Kentnissen kategorisiert	94

# Tabellenverzeichnis

1	Raum-Zeit-Taxonomie für Groupware
2	Zusammensetzung des Interviews
3	Ziele von Saros Anwendern
4	Benutzer- und Zielgruppen von Saros
5	Interviewfragen zu Aktivitäten (abstrakte Ebene)
6	Interviewfragen zu Fähigkeiten (abstrakte Ebene)
7	Interviewfragen zu Zielen (abstrakte Ebene)
8	Interviewfrage zur Einstellung (abstrakte Ebene)
9	Frage-Kategorien zur Zielgruppe (abstrakte Ebene)
10	Frage-Kategorien zu Saros (abstrakte Ebene)
11	Einordnung der Fatalität
12	Farbliche Kennzeichnung der Fatalitäten
13	Frage-Kategorien zur Demographie
14	Notation von Tasks (Beispiel)
15	Notation der Problembeschreibungen
16	Zusammenfassung der Usability-Probleme der Konzepte
17	Zusammenfassung der Usability-Probleme der Wünsche und Beschwerden 6
18	Zusammenfassung der verwendeten Begriffe
19	Zusammenfassung der Arbeitspakete

# Listings

1	Ausschnitt der Definition der Viewport-Annotation in der plugin.xml	77
2	Deklaration des SendFile-Commands	87
3	Deklaration des SendFile-CommandHandlers	87

### 1 Einleitung

Die Software-Entwicklung ist ein komplexer Prozess, der maßgeblich durch die involvierten Menschen beeinflusst wird. Folglich macht es einen Unterschied, ob bei der Entwicklung nur eine einzelne Person oder gar mehrere Teams beteiligt sind. Aber auch der Ort der Zusammenarbeit spielt ein Rolle: Arbeiten die Teammitglieder am selben Rechner, im gleichen Büro oder nicht einmal auf dem selben Kontinent. Und schließlich beeinflusst die zeitliche Komponente das Zusammenspiel der Software-Entwicklung. So stellt sich die Frage: Findet die Arbeit der Mitglieder gleichzeitig oder asynchron statt?

Anhand der genannten Einflussfaktoren, die keinen Anspruch auf Vollständigkeit erheben, sei verdeutlicht, dass es sich bei der Entwicklung von Software um kein leichtes Unterfangen handelt. Vor allem wenn es sich um mehrere Software-Entwickler handelt, die gemeinsam und gleichzeitig zusammenarbeiten wollen, sich jedoch nicht am selben Ort aufhalten.

Für diese Form der Zusammenarbeit stellt Saros eine Lösung dar.

### 1.1 Saros

Saros ist ein Open-Source-Plugin für die Entwicklungsumgebung Eclipse zur verteilten, gemeinsamen und gleichzeitigen Programmierung.

Diese spezielle Form der Zusammenarbeit - unter Verwendung von Eclipse und Saros - findet innerhalb einer sogenannte Saros-Sitzung statt.

Während einer solchen Sitzung besitzen alle Teilnehmer eine identische Kopie aller Projektdateien. Jede Änderung an einer Datei durch einen Teilnehmer wird in Echtzeit an der jeweilig entsprechenden Datei der anderen Teilnehmer reproduziert. Das heißt, Saros stellt zum Einen sicher, dass alle Beteiligten stets an einer konsistenten Kopie arbeiten und überträgt andererseits, woran jeder Sitzungsteilnehmer arbeitet und welches seine letzten Änderungen gewesen sind. [2, 3]

Die erste Version von Saros wurde 2006 von [4] im Rahmen einer Diplomarbeit entwickelt. Ursprünglich sah diese eine verteilte Zusammenarbeit von lediglich zwei Personen vor. Seitdem wurde die Entwicklung des Eclipse-Plugins in der Arbeitsgruppe "Software Engineering"<sup>2</sup> am Institut für Informatik der Freien Universität Berlin hauptsächlich durch wissenschaftliche Mitarbeiter und Studenten vorangetrieben. Saros wurde so um einige Funktionen reicher, sodass mittlerweile bis zu fünf Teilnehmer in einer Saros-Sitzung zusammenarbeiten können.

Für weiterführende Informationen sei an dieser Stelle nur auf die Saros-Webseite<sup>3</sup> hingewiesen. Eine aktuelle Version von Saros kann zudem stets unter http://dpp.sourceforge.net/update abgerufen werden.

Eine ausführlichere Vorstellung von Saros und den verwendeten Konzepten wird in Kapitel 2.2 vorgenommen.

<sup>2</sup>http://www.inf.fu-berlin.de/en/groups/ag-se/index.html

<sup>3</sup>http://www.saros-project.org

2 1 Einleitung

### 1.2 Motivation

Wie im vorherigen Abschnitt 1.1 angeführt, wurde Saros seit 2006 stetig von Studenten und wissenschaftlichen Mitarbeitern der Freien Universität Berlin weiterentwickelt. Dies erfolgt hauptsächlich durch Abschlussarbeiten und umfasst sowohl die Verbesserung von Saros als auch die Entwicklung neuer Funktionalitäten [5].

Diese Funktionen sollen die gleichzeitige und gemeinsame Arbeit von verteilten Beteiligten erleichtern. Damit Anwender von den in Saros integrierten Funktionen profitieren zu können, reicht die ledigliche Bereitstellung dieser aber nicht aus. Um ein System und dessen Funktionen besser verwenden und erlernen zu können, müssen Anwender in die Lage versetzt werden, dies auch verstehen zu können [6, 7, 8]. Das heißt, die Saros-Anwender müssen in die Lage versetzt die Konzepte in Saros verstehen zu können.

### 1.2.1 Worin besteht das Problem?

Vorangegangene Abschlussarbeiten haben gezeigt, dass Anwender Schwierigkeiten mit der Benutzung von Saros haben [9, 10, 11]. Zum Teil ist dies darauf zurückzuführen, dass grundlegende Konzepte und Funktionen von Saros nicht nicht wahrgenommen, falsch oder unzureichend verstanden werden [11, 12].

Ein Beispiel für ein derartiges Missverständnis ist das derzeitig in Saros verankerte *Host*-Konzept:

Derjenige Anwender, der eine Sitzung initiiert, nimmt in Saros eine Sonderrolle namens Host ein. Zu Beginn einer Sitzung bekommen alle anderen Teilnehmer die Daten vom Host übertragen. Er ist der Einzige, der anderen Sitzungsteilnehmern Lese- und Schreibrechte gewähren und entziehen kann. Außerdem wird eine Saros-Sitzung automatisch für alle Beteiligten beendet, sobald der Host diese verlässt. Damit nicht einhergehend ist die Farbe, die sich jeder Sitzungsteilnehmer selbst zuweisen kann. Diese freie Farbwahl ist unabhängig davon, ob man sich in der Rolle des Hosts befindet oder nicht. Nichtsdestotrotz wurde in einem Test von [12] aufgedeckt, dass Anwender zwischen der Host-Rolle und der Farbe rosa eine Assoziation hergestellt haben.

Obwohl dieser Zusammenhang so nicht in Saros vorhanden ist, wurde fälschlicherweise angenommen, dass derjenige mit der Farbe rosa der Host sein müsse. [12, Usability-Test 3]

### 1.2.2 Warum stellt dies ein Problem dar?

Wenn Funktionen und Konzepte in Saros nicht wahrgenommen, falsch oder unzureichend verstanden werden, zieht dies einige negative Konsequenzen nach sich. Zur Veranschaulichung seien hier einige Beispiele genannt:

- Anwender nehmen Saros als unausgereift wahr. (Quelle: Studenten der FU, Mitarbeiter von Optimal Systems)
- Anwender finden Saros ist seltsam bzw. eigenartig zu verstehen. (Quelle: Saros Fragebogen [Antwort 38] "kinda weird to figure out")

• Anwender wünschen sich Funktionalität, die bereits vorhanden ist.

Beispiel: Das "Remote Screen" Feature

(Quelle: Saros Fragebogen [Antwort 73] "missing [..] video")

• Anwender sind irritiert.

Beispiel: Änderungen innerhalb des  $Editor\ Views$  werden synchronisiert, Modifikationen in anderen Views wie dem Eclipse  $Package\ Explorers$  hingegen nicht

(Quelle: Usability-Test von [12])

### 1.3 Zielsetzung und Fokus

Das Ziel dieser Arbeit ist es die Lücke zwischen den in Saros verwendeten Konzepten und dem Verständnis der Anwender von Saros schließen. Um dies zu erreichen, gilt es zunächst selbst zu verstehen, wie die Anwender Saros verstehen. So kann auf Verständnisprobleme reagiert und Saros entsprechend angepasst und verbessert werden.

Im Rahmen dieser Arbeit wird dafür zunächst die Zielgruppe(n) von Saros bestimmt. Weiterhin soll erfasst werden, welche Kenntnisse durch diese Zielgruppe(n) mitgebracht werden und welches Wissen notwendig ist, um die in Saros verankerten Konzepte zu verstehen. Darauf aufbauend, werden Maßnahmen entwickelt, die entweder ein Änderung von Saros vorsehen, um sich den Anwendern anzupassen oder, die die Anwender unterstützt diese Konzepte leichter zu erfassen.

Dabei liegt der Fokus dieser Arbeit auf den Konzepten, die innerhalb einer aktiven also laufenden Saros-Sitzung zum Tragen kommen, so zum Beispiel das erwähnte Host-Konzept, der Verfolgermodus und andere Awareness-Informationen (siehe Kapitel 2.2.2).

Der Sitzungsauf- und -abbau selbst steht dabei ebenso wenig im Vordergrund, wie die noch nicht ausgereiften Saros-Funktionen Whiteboard und Remote Screen.

### 1.4 Aufbau der Arbeit

Diese Arbeit befasst sich mit der Verbesserung der Usability des Plugins Saros. Diese Verbesserung soll durch das Erfassen des Verständnisses der Anwender erreicht werden.

besteht neben dieser Einleitung aus dem sich direkt anschließenden Grundlagenkapitel, dem Hauptteil, sowie dem recht umfangreichen Anhang.

Der Aufbau dieser Arbeit ist deshalb eng an den Prozess des User-Centered Design geknüpft. Dieser Begriff wird neben anderen Grundlagen im nachfolgenden Kapitel näher erläutert werden.

Kapitel 2 behandelt alle grundlegenden Begriffe und Konzepte dieser Arbeit.

Das Ziel dieses Kapitels ist es, an einer zentralen Stelle alle für diese Arbeit relevanten Grundlagen zu erläutern. Zu diesen gehören genauere Informationen zu Saros 2.2, eine Begriffserklärung des mentalen Modells 2.3, der Gebrauchstauglichkeit 2.4, als auch des User-Centered Design Ansatzes 2.5.

4 1 Einleitung

Kapitel 3 stellt vier verschiedene Methoden zur Erhebung des mentalen Modellen vor. Im abschließenden Unterkapitel 3.5 werden diese gegeneinander abgewogen und entschieden, welche die geeignete Methode für diese Arbeit darstellt.

- Kapitel 4 behandelt die Vorbereitung und Durchführung der Interviews. Im Rahmen der Vorbereitung werden die Zielgruppen von Saros als auch der Umfang des Interviews ermittelt. Zudem wird das verwendete Material, die Versuchspersonen und die Versuchsdurchführen beschrieben.
- Kapitel 5 wertet die Interviews aus und stellt deren Ergebnisse vor. Eines der Ergebnisse ist das mentale Modell der Anwender. Zudem wird für die identifizierten Usability-Probleme nach Lösungsansätzen gesucht und diese zu Arbeitspaketen zusammengefasst.
- Kapitel 6 beschreibt die Behandlung und Behebung der priorisierten vier Arbeitspakete.
- Kapitel 7 evaluiert den erstellten Comic anhand eines Online-Fragebogens und diskutiert die Umsetzung der anderen Arbeitspakete.
- Kapitel 8 enthält ein zusammenfassendes Fazit und einen Ausblick über zukünftige Weiterentwicklungen.

Dieses Kapitel befasst sich mit den Grundlagen der vorliegenden Arbeit. Es werden alle relevanten Begriffe, Konzepte und Prozesse erläutert, die im weiteren Verlauf der Arbeit verwendet werden.

### 2.1 Eclipse

Eclipse ist eine Open-Source-Gemeinschaft, deren Projekte die Entwicklung einer (quell-) offenen Entwicklungsplattform als Schwerpunkt haben.

Diese Plattform umfasst erweiterbare Frameworks, Tools und Runtimes zur Entwicklung, Verteilung and Verwaltung von Software über ihren kompletten Lebenszyklus hinweg. Die Beherbergung der Projekte wird von der gemeinnützigen Organisation Eclipse Foundation übernommen.

Die von der Gemeinschaft entwickelte Eclipse-Plattform ist eine Entwicklungsumgebung, die gemeinhin als Eclipse bezeichnet wird. Die Funktionalität von Eclipse lässt sich durch sogenannte Plugins erweitern. [13] (Ausführlichere Informationen zu Eclipse liefert die Infobox auf Seite 6.)

Wenn im weiteren Verlauf der Arbeit von *Eclipse* die Rede ist, dann ist damit stets die Entwicklungsumgebung und nicht die Open-Source-Gemeinschaft gemeint.

### 2.2 Saros

Dieses Unterkapitel stellt Saros in drei Abschnitten auf drei unterschiedlichen Ebenen vor. Abschnitt 2.2.1 ordnet Saros in die vorhandene Software-Landschaft und dem zugehörigen Fachgebiet ein. Diese Einordnung soll zum besseren Verständnis von Saros selbst als auch den in Saros verwendeten Konzepten beitragen.

Anschließend wird in Abschnitt 2.2.2 der Begriff Awareness erläutert. Zudem wird veranschaulicht, wie die grafische Integration dieses Konzepts, des Verfolgermodus und des Host-Konzepts in Eclipse realisiert sind (siehe Abschnitt 2.2.3).

### 2.2.1 Einordnung in die Software-Landschaft

Saros ist ein Plugin für Eclipse, durch das "Eclipse um die Funktionalität der Programmierung mit verteilten Beteiligten erweitert" wird [9, S. 5]. Damit gehört Saros zu der Kategorie von Systemen, die die Zusammenarbeit von Gruppen unterstützten und gemeinhin als *Groupware* bezeichnet werden. [14] definieren Groupware als:

"computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment." [14, S. 40]

### Die Eclipse Oberfläche - Editoren, Ansichten und Perspektiven

Eclipse besteht aus einer Reihe von Komponenten. Eine davon ist die *Eclipse Workbench* (siehe Abbildung 1). Die Workbench bezeichnet das Hauptfenster und ist die grafische Oberfläche von Eclipse, die beim Start von Eclipse erscheint [15]. Die Oberfläche setzt sich aus verschiedenen Fenstern zusammen: Den *Ansichten* (engl. views) und einem speziellen *Editor*-Fenster.

Der Editor dient der Bearbeitung von Dokumenten. Für verschiedene Arten von Dokumenten existieren verschiedene Arten von Editoren. Eclipse wählt für jede Datei den möglichst passenden Editor. Für einfache Text-Dokumente wird der integrierte Text-Editor verwendet. Handelt es sich um eine Java-Datei, wird diese im Java-Editor geöffnet, der über zusätzliche Eigenschaften wie das Überprüfen der Syntax verfügt.

Ansichten müssen nicht selbst ausgewählt und angeordnet werden, dafür gibt es *Perspektiven*. Perspektiven sind aufgabenorientiert gestaltet. So gibt es je eine Perspektive zum Schreiben und zum Debuggen von Java-Programmen. [15]

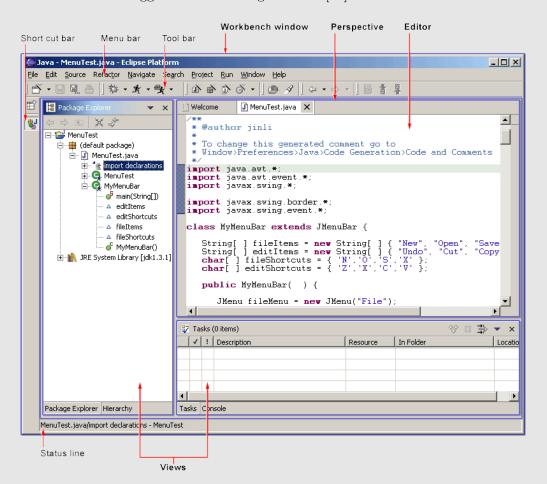


Abbildung 1: Die Eclipse Workbench [16]

2.2 Saros 7

Unter dieser relativ generischen Definition versteht man sowohl Hardware- und Software-Systeme als auch Dienste zur Unterstützung von Gruppen. In der Praxis wird der Begriff aber zumeist für Software verwendet. [17, S. 10]

Groupware ist zudem ein großer Bereich des multidisziplinären Feldes der rechnergestützten Gruppenarbeit oder Computer-supported Cooperative Work (kurz CSCW) [18, 14, 17]. Der Forschungsbereich CSCW beschäftigt sich mit

"dem Verstehen sozialer Interaktionen sowie der Gestaltung, Implementation und Evaluierung von technischen Systemen zur Unterstützung sozialer Interaktion [..]." [17, S. 10]

Die im Bereich des CSCW erforschten theoretischen Grundlagen und Methoden der Gruppenarbeit haben demnach großen Einfluss auf die entwickelten CSCW-Systeme (Groupware).

Um einen guten Überblick über die verschiedenen CSCW-Anwendungen zu bekommen, darüber wie sich Saros zu den Anwendungen verhält und welche Funktionalität diese typischerweise aufweisen, werden im Folgenden zwei unterschiedliche Ansätze zur Klassifizierung vorgestellt.

### Klassifizierung

Grundsätzlich existieren eine Reihe von Kategorisierungsansätzen für CSCW-Systeme. Die beiden, die für Saros am relevantesten sind, werden an dieser Stelle vorgestellt.

Die wohl älteste und bekannteste Groupware-Klassifikation ist die 2-dimensionale Raum-Zeit-Taxonomie, die ursprünglich auf Johansen (1991) zurückzuführen ist [14, 17]. Diese ordnet Groupware-Anwendungen danach ein, ob die Hindernisse der Zusammenarbeit zeitlicher oder örtlicher Natur sind. Das heißt, die Kollaboration kann am gleichen oder verschiedenen Orten stattfinden und asynchron oder gleichzeitig ablaufen. Dadurch ergeben sich ingesamt vier Kategorien (siehe Tabelle 1).

Saros lässt sich in die linke untere Ecke einordnen (gleiche Zeit / verschiedene Orte) [4].

	Gleiche Zeit	Verschiedene Zeit		
Gleicher Ort	Gesicht-zu-Gesicht	Asynchrone		
Gleicher Ort	Interaction	Interaktion		
Verschiedener Ort	Synchrone verteile	Asynchrone verteilte		
verschiedener Ort	Interaction	Interaction		

Tabelle 1: Raum-Zeit-Taxonomie für Groupware nach [14]

**Anmerkung:** Die Quelle für die Raum-Zeit-Taxonomie nach Johansen (1991) ist im Orginal nicht auffindbar. Aus diesem Grund sei auf [14] verwiesen, die diese Taxonomie aufgreifen und auf Johansen verweisen.

Diese Klassifikation wurde hauptsächlich deshalb herangezogen, um ermitteln zu können, welche Formen der Kommunikation für die jeweilige Kategorie sinnvoll sind [17]. So unterscheiden sich die Anforderungen und dementsprechend auch die Systeme selbst stark voneinander.

Würde Saros in die Kategorie "Asynchrone verteilte Interaktion" fallen, dann wäre die vorhandene Chat- und VoIP-Funktion vermutlich überflüssig, da diese Funktionen die zeitliche Verfügbarkeit des Kommunikationspartners voraussetzen.

Die Klassifikation nach Johansen (1991) hat aber auch seine Schwächen. Es lassen sich nicht alle CSCW-Systeme einer Kategorie eindeutig zuordnen. Zum Beispiel können Versionskontrollsysteme wie SVN auf jedem Gebiet eingesetzt werden. Andererseits gehören nach dieser Taxonomie einige Systeme derselben Kategorie an, dessen Anforderungen an eine gemeinsame Zusammenarbeit nicht unterschiedlicher sein könnten. Dieser Fall trifft beispielsweise auf Saros und Videokonferenzsysteme zu. Beide gehören zur Art "Synchrone verteilte Interaktion", haben aber andere Möglichkeiten und Anforderungen an die Kommunikation. Da sich die Teilnehmer bei Saros nicht sehen können, werden non-verbale Informationen wie Gestik mit dem Mauszeiger durch andere Konzepte ersetzt, was bei einer Videokonferenz nicht notwendig ist (siehe Abschnitt 2.2.2 und 2.2.3).

Eine zweite Variante zur Klassifizierung von CSCW-Systemen erfolgt nach [19] und wird als 3K-Modell bezeichnet. Bei dieser werden Groupware-Anwendungen nach dem Grad ihrer Kommunikations-, Kooperations- und Koordinationsunterstützung eingeordnet [18, 17]. Diese drei Bereiche spannen ein Dreieck auf, in das sich die Groupware je nach Ausprägung der Funktionen einsortieren lässt (siehe Abbildung 2).

Die Hauptfunktion von Saros entspricht der eines *Gruppeneditors*, weshalb es sich in den Kooperationsbereich einordnen lässt (rechte untere Ecke).

# Kommunikation E-Mail Chat Videokonferenzen VolP Foren Shared Whiteboard Weblogs Wikis Gruppenkalender Planungssoftware Sitzungsunterstützungssysteme Workflow-Management Gruppeneditoren Kooperation

Abbildung 2: Das 3K-Modell für Groupware nach [19]

Bildquelle: http://mussswiki.idv.edu/iv1/Datei:3K\_Modell.png

Verbindet man beide Klassifikationen, kommt man einer Eingrenzung von Saros wesentlich

 $2.2 \quad Saros$  9

näher als unter Anwendung einer einzelnen: Saros ist eine CSCW-Anwendung zur verteilten Koorperation in Echtzeit.

Auch wenn sich die Hauptfunktion von Saros auf den Kooperationssaspekt fokussiert, werden dadurch keine Funktionen in den Bereichen Kommunikation und Koordination ausgeschlossen (Vgl. Chat- und VoIP-Funktion).

### 2.2.2 Awareness

Awareness ist ein in Saros umgesetztes Konzept, das umfangreicher ist als andere Konzepte in Saros. Aus diesem Grund wird der Erläuterung des Awareness-Konzepts ein eigener Abschnitt gewidmet, bevor im nächsten Abschnitt die grafische Darstellung und die anderen für diese Arbeit relevanten Konzepte vorgestellt werden.

Der Begriff Awareness wird definiert als das Wissen um die Aktivitäten anderer Personen, welches den Kontext der eigenen Handlungen bestimmt [20, Vgl. S. 107].

Vereinfach ausgedrückt kann Awareness als "knowing what is going on" verstanden werden [21, S. 36].

Da diese Definition von Awareness sehr allgemein gehalten ist, wird das Awareness-Konzept im Bereich der Groupware durch den *Shared Workspace* genauer spezifiziert. Damit wird die Betrachtung auf die Informationen innerhalb des gemeinsamen Arbeitsplatzes eingeschränkt. [4, S. 28]

Dieses Konzept wird als Workspace Awareness (kurz WA) bezeichnet und von [22] definiert als:

"up-to-the-moment understanding of another person's interaction with a shared workspace" [22, S. 33]

Im Falle der Groupware Saros beschreibt die WA also das aktuelle Wissen um die Interaktionen anderer Personen mit der gemeinsamen Arbeitsumgebung Eclipse. Diese werden von jeder einzelnen Person benötigt, um sich mit den Anderen zu koordinieren und die eigenen Aufgaben innerhalb der Gruppe zu erfüllen [23, Vgl.].

In Bezug auf Saros liegt der Fokus auf der Darstellung von Eclipse-Interaktionen durch die anderen Sitzungsteilnehmer. Es werden Handlungen wie das Markieren in einer Datei, die zuletzt durchgeführten Änderungen oder auch der aktuelle Sichtbereich der Teilnehmer übertragen und grafisch dargestellt.

Wie diese Darstellung konkret für das Awareness- und die anderen Konzepte aussieht, wird im nachfolgenden Abschnitt gezeigt.

### 2.2.3 Konzepte

In diesem Abschnitt werden für diese Arbeit relevanten Konzepte von Saros erläutert und verdeutlicht, wie sich diese grafisch in Eclipse integrieren. Dabei wird der Begriff Konzept stellvertretend für die Grundlage oder Idee einer Umsetzung verwendet.

### Die Awareness-Informationen

Als erstes wird die grafische Umsetzung des im vorherigen Abschnitt erwähnten Awareness-Konzepts vorgestellt. Die in der Abbildung 3 dargestellten Ziffern markieren die Positionen der sichtbaren Interaktionen und Präsenzinformationen der anderen Sitzungsteilnehmer:

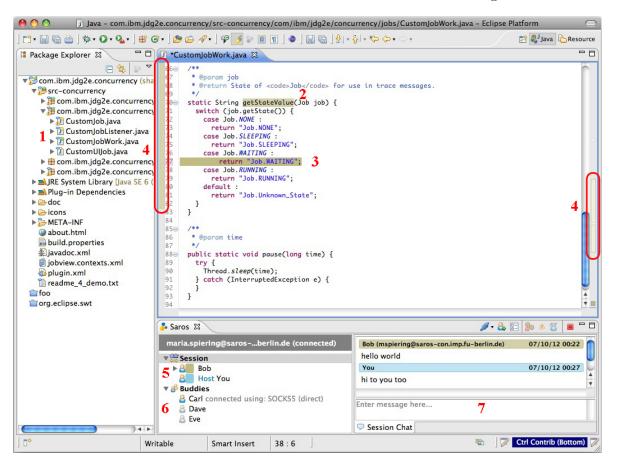


Abbildung 3: Saros-annotierter Screenshot mit Awareness-Information

- Package Explorer View
  - 1 Zeigt an, welche Dateien mit anderen Teilnehmer geteilt werden (blau), offen (gelb) oder sichtbar sind (grün).
- Editor
  - 2 Der Markierte Text eines Teilnehmers
  - 3 Die Cursorposition eines Teilnehmers
  - 4 Durch einen Teilnehmer veränderter Text (standardmäßig die letzten zwanzig Zeichen).
  - 5 Der aktuelle Sichtbereich der anderen Teilnehmer
    -links: der sichtbare Ausschnitt der Anderen in Bezug auf den eigenen Sichbereich
    -rechts: der Sichtbereich der anderen Teilnehmer in Bezug auf die Datei

 $2.2 \quad Saros$  11



Abbildung 4: Vergrößerte Darstellung der verwendeten Symbole im Package Explorer für geteilte, offene und sichtbare Dateien (von links nach rechts)

Die drei Hauptkomponenten der Saros-Oberfläche:

- Saros-View
  - 6 Die Teilnehmer der aktuellen Sitzung (Die Farbe der Teilnehmer entspricht der Annotationen im Editor.)
  - [7] Die Buddy-Liste: Sie beinhaltet alle Kontakte, die der Anwender kennt.
  - 8 Das Chat-Fenster: Hier können Nachrichten mit den Teilnehmern der aktuellen Sitzung ausgetauscht werden.

Stößt ein Benutzer zu einer Sitzung hinzu, bekommt dieser eine Farbe zugewiesen, welche für alle Annotationen verwendet wird. Die Farbe kann über die Eclipse-Einstellungen und das Saros-Kontextmenü verändert werden.

### Der Verfolgermodus

Beim Follow Mode handelt es sich um eine Funktion, durch die zusätzliche Awareness-Informationen eines Teilnehmers übertragen werden.

Folgt man einem anderen Situngsteilnehmer, so werden alle Navigationsbewegungen dieser Person an den Verfolger übertragen. Dabei werden die Aktionen nicht nur übertragen, sondern auch automatisch beim Verfolger durchgeführt. Folgende Aktionen sind davon betroffen:

- Öffnen einer Datei
- Wechseln zu einer anderen Datei
- Ändern des sichtbaren Bildschirmausschnitts in einer Datei (scrollen)

Ist der Follow-Modus aktiv, so wird dies für alle Parteien im Saros-View angezeigt und ist durch die Information "(following)" gekennzeichnet (siehe Abbildung 5).



Abbildung 5: Darstellung der Informationen zum Verfolgermodus und dem Host-Konzept im Saros-View

### Das Host-Konzept

Bevor das Konzept selbst erläutert wird, ist es wichtig zu verstehen, dass dieses Konzept nicht eingeführt wurde, um den Anwender bei seiner Arbeit zu unterstützen. Es existiert vielmehr aus historisch Gründen und wurde bereits mit der ersten Version von Saros durch [4] eingeführt.

Durch die folgenden sechs Punkten werden alle Aspekte des Host-Konzepts zusammengefasst:

- 1. Als *Host* wird der Benutzer bezeichnet, der die Sitzung eröffnet hat.
- 2. Die Daten des Hosts stellen die Ausgangsbasis für eine Sitzung dar. Bei den Daten handelt es sich entweder um ein ganzes Projekt oder einzelne Dateien, die an alle Sitzungsteilnehmer übertragen werden.
- 3. Der Host verteilt auf technischer Ebene die Nachrichten eines Sitzungsteilnehmers an die übrigen Teilnehmer.
- 4. Nur der Host kann Lese- und Schreibrechte vergeben.
- 5. Die Rolle des Hosts ist nicht übertragbar.
- 6. Nur der Host kann die gesamte Sitzung beenden. Das heißt, verlässt er die Sitzung, dann wird diese für alle Beteiligten geschlossen.

Welcher Sitzungsteilnehmer sich in dieser Sonderrolle befindet, ist für alle Beteiligten im Saros-View ersichtlich. Der Name des Teilnehmers ist mit dem Präfix "Host" versehen (siehe Abbildung 5).

Um zu Erheben wie Anwender Saros-Konzepte wie Awareness wahrnehmen und verstehen, muss der Begriff *Mentales Modell* eingeführt werden.

### 2.3 Mentale Modelle

Im Bereich der Kognitiven Psychologie gibt es eine Fülle an Begriffen, um den Prozess zu erklären, wie Indiviuen ihre Umwelt wahrnehmen und verstehen. Dabei beziehen sich alle Konzepte auf eine Art Repräsentation von Wissen. [24, S. 3]

Im Kontext der Software-Ergonomie hat sich dafür der Begriff *Mental Model* (kurz MM) durchgesetzt und kann wie folgt verstanden werden [7, 25, 26, 27, 28, Vgl.].

"Mental models are internal representations of systems in particular knowledge domain. These internal representations are formed through knowledge (instruction) or experience or a combination of the two." [8, S. 601]

Demnach handelt es sich bei mentalen Modellen um implizites Wissen über ein System in einem bestimmten Kontext. Dieses Verständnis wird durch eine Kombination aus Wissen und Erfahrungen aufgebaut. Diesen Aufbau beschreibt [29] etwas präziser und benennt die *Interaktion* als die treibende Kraft hinter dem Enstehungsprozess. Aus diesem Grund sei es mithilfe eines vorhanden mentalen Modells auch möglich Interaktionen und Verhaltensweisen zu erklären und vorherzusagen [29, S. 7]. Nichtsdestrotz bleibt unklar, wie der Entstehungsprozess genau abläuft und Erkenntnisse verinnerlicht wird [8].

2.4 Usability 13

Um also MM zu erfassen, muss diese interne Repräsentation zunächst wieder externalisiert werden [30, Vgl.]. [7] stellte jedoch fest, dass es Anwendern zumeist schwer fällt ihre mentalen Modelle zu artikulieren. Aber es gab auch einige erfolgreiche Ansätze, um MM zu erfassen. Welche Methoden zur Erhebung von mentalen Modellen existieren und im Rahmen dieser Arbeit angewendet werden, wird in Kapitel 3 erläutert.

Unter der Annahme, dass es möglich ist, ein mentales Modell zu erheben, folgt die zentrale Frage: Wie können mentale Modelle dabei helfen die Usability zu verbessern?

Zur Beantwortung dieser Frage, wird zunächst der Begriff der Usability definiert, um abschließend in 2.4.2 die Frage erneut aufzugreifen und zu beantworten.

### 2.4 Usability

Der englische Begriff *Usability* wird im deutschen als *Gebrauchstauglichkeit* bezeichnet - und wurde ehemals mit *Benutzerfreundlichkeit* übersetzt [9].

Die Internationalen Organisation für Normung (ISO) gilt als die Hauptreferenz [31, 32, Vgl.] und definiert Gebrauchstauglichkeit als

"das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen." [DIN EN ISO 9241-11, 1998]

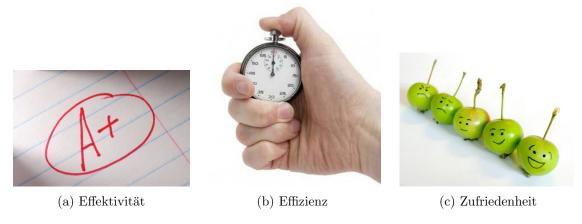


Abbildung 6: Die drei Hauptaspekte der Usability (nach ISO 9241)

Die einzelnen Usability-Aspekte werden zudem wie folgt differenziert:

Effektivität Die Genauigkeit und Vollständigkeit mit der Anwender das festgelegte Ziel erreichen.

**Effizienz** Die verbrauchten Ressourcen, in Relation zur Genauigkeit und Vollständigkeit, die der Benutzer benötigt um sein Ziel zu erreichen.

**Zufriedenheit** Freiheit von Unbehagen und positive Einstellung zur Benutzung des Produkts.

Die Definition nennt zudem mit den Anwendern, dessen Zielen und den Benutzungskontext des Produkts drei Rahmenbedingungen für die Aspekte. Diese drei Bedingungen stecken auch den Rahmen für den ersten Iterationsschritt beim *User-Centered Design* (kurz UCD). Dieser Schritt wird in Abschnitt 2.5 als *Context Of Use* bezeichnet. In diesem ersten Schritt des UCD gilt es die drei Rahmenbedingungen zu identifizieren.

So relevant dieser Teil der Definition für das User-Centered Design ist, umso mangelhaft hilfreich sind die beschriebenen Aspekte in Bezug auf mentale Modelle. Die Definition beschreibt einen Maßstab, der bei der Benutzung des Produkts eingehalten werden muss, jedoch nicht die Ursachen, die zu der gewünschten Effektivität, Effizienz und Zufriedenheit führen. [33]

Aus diesem Grund wird zusätzlich die wohl bekannteste Definition von Nielsen (1993) herangezogen [32]. Diese beschreibt die Gebrauchstauglichkeit ebenfalls als eine mehrdimensionale Eigenschaft eines Systems, die sich aus den folgenden Aspekten zusammensetzt [34, S. 26ff.]:

**Learnability** Das System sollte einfach zu erlernen sein, damit Benutzer schnell das System bedienen können.

Efficiency Das System sollte effizient zu benutzen sein, damit Anwender - nach erlernen des Systems - ein hohes Maß an Produktivität erreichen können.

Memorability Das System sollte einprägsam und leicht wiederzuerkennen sein, sodass auch gelegentliche Nutzer zu dem System zurückkehren können, ohne alles neu erlernen zu müssen.

Errors Das System sollte ein niedrige Fehlerrate besitzen, damit Anwender wenig Fehler machen können. Falls sie Fehler machen, dann sollten diese leicht rückgängig gemacht werden können. Zudem dürfen keine schwerwiegenden Fehler auftreten.

Satisfaction Das System sollte angenehm zu bedienen sein, sodass Anwender während der Benutzung subjektiv zufrieden sind.

Durch Aufnahme der Aspekte "Learnability" und "Memorability" lässt sich die Rolle der mentalen Modellen bei der Gebrauchstauglichkeit besser verstehen. Die Gebrauchstauglichkeit eines Produkt wird durch das Ausmaß bestimmt, in dem ein korrektes mentales Modell erlernt, behalten und wiedererkannt werden kann. So ist es Anwendern möglich ihre Ziele effizient, effektiv und zufriedenstellend erreichen zu können. [33, Vgl.]

Das heißt, je leichter es den Anwendern beispielsweise fällt das Host-Konzept zu erlernen und richtig zu verstehen, umso höher ist das Maß an Usability von Saros.

Da zur Verbesserung der Usability von Saros die mentalen Modelle der Anwender erhoben werden und das User-Centered Design zum Einsatz kommt, wird für diese Arbeit die Definition der ISO-Norm durch die Aspekte der "Erlernbarkeit" und "Einprägsamkeit" nach [34] erweitert.

### 2.4.1 Usability-Probleme

Laut [35] handelt es sich bei Usability-Problemen um alles, was mit der Fähigkeit des Anwenders interferiert, seine Aufgaben effizient und effektiv zu bewältigen<sup>4</sup>. Durch die Erweiterung

<sup>&</sup>lt;sup>4</sup>[9] zitiert bei Karat (1992) [35].

2.4 Usability 15

Usability-Definition, zählen auch Probleme bei der Erlernbarkeit und Einprägsamkeit zu den Usability-Problemen.

Wie können also solche Usability-Probleme erhoben werden?

Eine Möglichkeit stellen klassische Evaluationsmethoden wie Usability-Tests, die Heuristische Evaluation oder auch sogenannte Cognitive Walkthroughs dar. Jedoch konnte gezeigt werden, dass trotz der Durchführung mehrerer dieser klassischer Evaluationen das Erheben von mentalen Modellen eine Reihe neuer Usability-Probleme aufgedeckt werden konnten [27]. Die Autoren führen dies darauf zurück dass, "Designer oder Entwickler etwas 'Falsches' antizipieren, [und] die Entwicklung in suboptimale Bahnen lenken" [27, S. 281].

Ein Grund für dieses falsche Antizipieren sind vermutlich die klassischen Evaluationsmethoden selbst. Diese stellen zumeist nur fest, dass es ein Usability-Problem gibt, erheben aber nicht die Ursache für das Problem.

Dieser Umstand gleicht dem Problem der Usability-Definition, die durch die Effizienz, Effektivität und Zufriedenheit nur attestiert, dass ein Problem vorzuliegen scheint, aber ebenfalls nicht auf die Ursache dafür eingeht.

### 2.4.2 Mentale Modelle und Usability

Abschließend für soll die in Abschnitts 2.3 gestellte Frage "Wie können mentale Modelle dabei helfen die Usability zu verbessern?" hier erneut aufgegriffen werden.

Dafür wurde zuvor der Begriff "Gebrauchstauglichkeit" vorgestellt und anhand einer Studie dargelegt, dass mithilfe von mentalen Modellen die Usability verbessert werden kann.

Um zu beantworten wie dies möglich ist, muss jedoch zunächst der Begriff Conceptual Model näher zu erläutern werden.

"A conceptual model is invented to provide an appropriate representation of the target system, appropriate in the sense of being accurate, consistent, and complete. Conceptual models are invented by [..] engineers." [29, S. 7]

Das Target System Saros wird demnach durch ein konzeptuelles Modell repräsentiert. Dieses Modell ist eine vereinfachte Repräsentation der in Saros umgesetzten Konzepte. Die Workspace Awareness ist eines dieser Konzepte und wurde bereits in Abschnitt 2.2.2 vorgestellt. Eine Erklärung des WA-Konzepts wäre demnach ein konzeptuelles Modell auf dessen Grundlage der Anwender sein mentales Modell aufbauen könnte [7, S. 48].

Wird dem Benutzer kein konzeptuelles Modell präsentiert, dann stellt das Target System (Saros) und dessen Oberfläche (*System Image*) den Ausgangspunkt für sein mentales Modell dar [8, S. 588].

In Bezug auf die Benutzeroberfläche bedeutet dies, das im Idealfall das mentale Modell des Anwenders identisch mit dem Conceptual Model des Designers ist [27, S. 274]. Weicht das mentale Modell ab, so deutet dies auf ein Usability-Problem hin und sowohl das konzeptuelle Modell als auch das System Image überdacht werden.

Um diese Abweichung erkennen zu können, muss jedoch zunächst das mentale Modelle der Anwender erhoben werden. Hier steht im Vordergrund zu verstehen, welches Verständnis die

Anwender von Saros haben und welche Ziele sie verfolgen, um so eine benutzerfreundliche, effiziente und leicht erlernbare Software zu entwickeln. [27, Vgl. S. 274]

Das heißt, mithilfe dieses Wissens können zunächst Probleme identifiziert und anschließend behoben werden, indem die Lücke zwischen den Modellen geschlossen wird. Das Schließen einer Lücken kann auf zwei Wegen erfolgen [36]:

### (1) Anpassen des Systems

Hierbei wird Saros an das mentale Modell der Anwender angepasst.

Kann bei der Erhebung des Modells zum Beispiel festgestellt werden, dass die verschiedenen Aspekte des Host-Konzept den Anwender daran hindern effektiv und effizient zu arbeiten, dann sollte dieses Konzept abgeschafft werden.

### (2) Verbessern des mentalen Modells

Dabei soll das mentalen Modell der Anwender verbessert werden, damit dies eher dem zugrunde liegenden System entspricht.

Ist es beispielsweise nicht möglich, dass Host-Konzept abzuschaffen, so sollte es Anwender erleichtert werden, das Konzept zu erlernen und sich zu merken. (Das zugrunde liegende System und dessen Konzepte bleiben hierbei unverändert.)

In diesem Abschnitt wurde der Begriff Usability eingeführt eingeführt und erläutert, welche Rolle das mentale Modell der Anwender bei der Verbesserung der Gebrauchstauglichkeit von Saros spielt.

Der nächste Abschnitt stellt das User-Centered Design vor und wird veranschaulichen wie sich die Gebrauchstauglichkeit und die Erhebung des mentalen Modells in diesen Prozess eingliedern.

### 2.5 User-Centered Design

Das Ziel der benutzerorientierten Gestaltung ist es eine möglichste hohe Gebrauchstauglichkeit zu erreichen, indem die Anforderungen, Bedürfnisse und Präferenzen der Anwender in den Mittelpunkt gerückt werden und nicht das technisch Machbare [37, Vgl.].

Die Methode selbst kann wie folgt zusammengefasst werden:

"User-Centered Design (UCD) is a multidisciplinary design approach based on the active involvement of users to improve the understanding of user and task requirement, and the iteration of design and evaluation." [38, S. 105]

[38] nennt in der ausformulierten Definition der ISO 13407 Norm bereits die Grundprinzipien des UCD. Eines dieser Prinzipien besagt, dass Benutzer früh und aktiv in den Design-Prozess involviert werden sollten, da sie den Kontext, in dem die Software eingesetzt wird, am Besten kennen [39, S. 588f.]. Ein weiteres Prinzip ist der Prozess selbst, bei dem es sich um einen wiederholenden Ablauf von vier activities handelt. Diese Aktivitäten werden im Rahmen des Software-Entwicklungsprozesses durchlaufen und im Folgenden näher beschrieben.

### 2.5.1 Ein internationaler Standard

Der internatioale Standard ISO 13407 (Human-centred design process for interactive systems) bildet die Grundlage für viele UCD-Methoden. Der Standard beschreibt einen Gestaltungsprozesses bei dem vier Aktivitäten durchlaufen werden, die notwendig für das UCD sind [31, S. 547]. Dabei werden keine konkreten UCD-Methoden für die Aktivitäten vorgegeben. Passende Methoden werden für ein Projekt oder Produkt je nach verfügbaren Ressourcen selbst bestimmt. So unterscheidet sich die Wahl der Methoden von den Fähigkeiten der involvierten Entwickler und Designer, der verfügbaren Zeit und dem Zugang zu realen Benutzern. [40, S. 137] Die Aktivitäten und der Ablauf sind unabhängig von den Methoden und bleiben deshalb in ihrem Kern gleich.

Wurde der Bedarf für die Notwendigkeit eines solchen Prozesses erkannt, Formen vier Aktivitäten den iterativen Human-Centred Design Process. Am Ende eines jeden Durchlaufs wird entschieden, ob alle Anforderungen erfüllt worden sind oder eine weitere Iteration notwendig ist (siehe Abbildung 7).

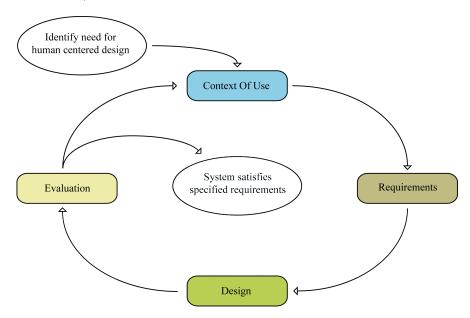


Abbildung 7: Aktivitäten des Human-Centred Design Prozesses nach ISO 13407 [41, Vgl.]

- Understand and specify the **context of use**Bei dieser Aktivität wird identifiziert, wer das Produkt verwendet (oder verwenden wird), wofür es benutzt wird und in welchem Kontext das Produkt zum Einsatz kommt.
- Specify the user and organizational **requirements**In diesem Schritt werden die Anforderungen seitens des Unternehmens (hier das SarosTeam der FU-Berlin) und der Anwender spezifiziert, die erfüllt werden müssen um ein
  zufriedenstellendes Produkt zu entwickeln.
- Produce **design** solutions
  Dieser Teil des Prozesses kann in mehreren Etappen durchgeführt werden, indem zum
  Beispiel Design-Konzepte mittels *Paper* oder *Software Prototyping* iterativ entwickelt
  werden (siehe [39, S. 609f.].

### • Evaluate designs against requirements

Am Ende eines jeden Iterationsschritts werden die umgesetzten Lösungen durch Anwender evaluiert. Anhand dieser Evaluation wird entschieden, ob die Anforderungen erfüllt wurden oder sich eine weitere Iteration anschließt.

Wie bereits erwähnt, bildet dieser Ablauf den Rahmen für das User-Centered Design ohne dabei genaue Vorgaben bezüglich der anzuwenden Methoden zu machen. Der nächste Abschnitt stellt die UCD-Methoden vor, die in dieser Arbeit zum Einsatz kamen und welche Aktivität sie zuzuordnen sind.

### 2.5.2 Die Vorgehensweise

Im Rahmen des UCD existieren eine Reihe von Methoden, die zur Verbesserung der Gebrauchstauglichkeit angewendet werden können. Zu den bekannteren zählen die Card Sorting, Personas, Paper Prototyping und die Heuristische Evaluation, um nur einige zu nennen. Auf eine Vorstellung aller Methoden sei an dieser Stelle verzichtet und auf die ausführliche Methoden-Übersicht von [39] verwiesen.

Um einen Überblick über diese Arbeit zu bekommen, werden deshalb an dieser Stelle "nur" die verwendeten Methoden kurz vorgestellt. Dabei wird lediglich beschrieben "wofür" die Methode angewendet wurde aber nicht "warum". Eine Begründung für die Wahl der Methode ist in den genannten Kapitel zu finden.

### • Saros-Fragebogen

Um eine bessere Vorstellung von den unterschiedlichen Benutzergruppen von Saros zu bekommen, werden die bereits vorhanden Daten des Saros-Fragebogens ausgewertet (siehe Kapitel 4.1.1).

### • UT-Auswertung

Mithilfe der Video-Aufzeichnungen von Usability-Tests wird der Umfang der Interviews bestimmt. Die Aufzeichnungen der Nutzertests enstanden im Rahmen der Abschlussarbeit von [12] (siehe Kapitel 4.1.2).

### • Interviews

Die Interviews stellen den Kern dieser Arbeit dar und dienen der Erhebung des mentalen Modells. Handelt es sich bei den Befragten um potentielle Anwender, soll anschließend zusätzlich die Struktur-Lege-Technik angewendet werden (siehe Kapitel 3 für die Abwägung mit anderen Methoden und Kapitel 4.2 für die eigentliche Durchführung).

### • Competitor Analysis

Saros wird mit ähnlichen Werkzeugen wie SubEthaEdit verglichen, um Lösungsansätze für die identifizierten Usability-Probleme zu finden (siehe Kapitel 5.1 und 5.3).

### • Richtlinien

Bei der Umsetzung der Lösungen werden die User Interface Guidelines von Eclipse zu Hilfe genommen (siehe Kapitel 6).

### • Umfrage

Zur Evaluation einer umgesetzten Lösung wurde ein Online-Fragebogen an eine gemischte Gruppe von Anwendern und potentiellen Anwendern versendet (siehe Kapitel 7).

Im Rahmen dieser Arbeit wird eine Iteration des vorgestellten Prozesses durchlaufen, so dass sich die sechs Methoden, wie in Abbildung 8 veranschaulicht, den einzelnen Aktivitäten zuordnen lassen.

### 2.5.3 Zusammenfassung

In diesem Abschnitt wurde der Begriff des User-Centered Design erläutert und die zugrunde liegenden Aktivitäten des Prozesses veranschaulicht. Darüber hinaus wurden die für diese Arbeit relevanten UCD-Methoden vorgestellt und den einzelnen Aktivitäten zugeordnet. Dabei wurden die zuvor eingeführten Begriffe "Usability" (Gebrauchstauglichkeit) und "Mentale Modelle" erneut aufgegriffen und aufgezeigt, wie sich diese in den Prozess eingliedern.

In den folgenden Kapiteln wird nun beschrieben, wie diese Vorgehensweise umgesetzt wird. Dafür wird im nachstehenden Kapitel 3 zunächst bestimmt, welche Methode sich im Rahmen dieser Arbeit für die Erhebung von MM eignet. Anschließend beginnt mit Kapitel 4 der erste Teilschritt der Iteration.

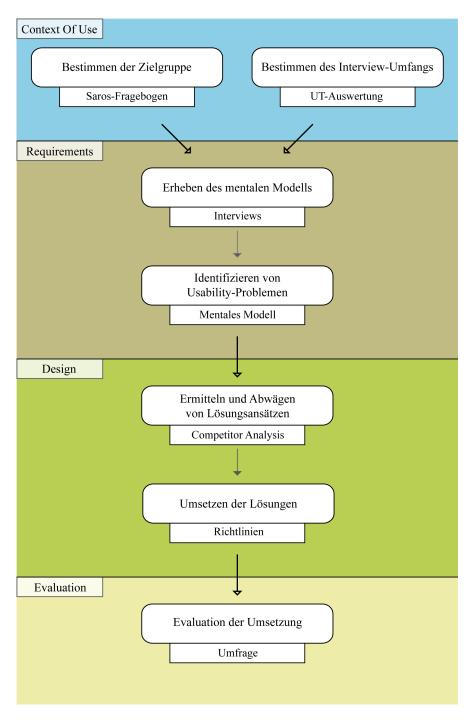


Abbildung 8: Eingesetzte UCD-Methoden dieser Arbeit

### 3 Methoden zur Erhebung von Mentalen Modellen

Die Methoden zur Erhebung von mentalen Modellen stellen eine Untergruppe der Evaluationsmethoden dar [27]. Dabei wird unter Evaluation im Allgemeinen die objektive Bewertung eines geplanten, bearbeiteten oder abgeschlossenen Projektes verstanden. Die Evaluationsdaten werden zum Zweck der Beantwortung spezifischer Fragestellungen oder zur Feststellung eines Zielerreichungsgrades erhoben. [9]

Im Rahmen dieser Arbeit wird das mentale Modell der Anwender von Saros erhoben. Dadurch sollen vermeintliche Annahmen und Verständnislücken aufgedeckt werden. Durch die Schließung der bestehenden Verständnislücken wird das umfassendere Ziel der Verbesserung der Gebrauchstauglichkeit erreicht.

Die Begegnung mit dem bestehenden Verständnisproblem soll auf zwei Ebenen stattfinden. Zum Einen auf der abstrakten Ebene (1), die Saros lediglich als Werkzeug zur lokalen oder verteilten Zusammenarbeit begreift und zum Anderen auf der konkreten Ebene (2), die untersucht wie gut oder schlecht einzelne Funktionen und Konzepte von Saros wahrgenommen und verstanden werden.

Zusammengenommen soll diese Herangehensweise Erkenntnisse liefern, an welchen Stellen Saros mit den Erwartungen und dem Vorwissen der Anwender übereinstimmt (und an welchen nicht).

Die zentrale Frage ist, welche Möglichkeiten gibt es, um das Verständnis der Anwender von Saros und der Zusammenarbeit zu erfassen, oder genauer gefragt: Wie kann man ein vorherrschendes mentales Modell erheben?

Im Folgenden werden verschiedene Ansätze für die Erhebung eines mentalen Modells vorgestellt.

### 3.1 Inspektion

Eine Inspektion ist ein manuell statistisches Verfahren, welches zur analytischen Qualitätssicherung herangezogen wird. Bei diesem sehr aufwändigem Teamprozess gilt es möglichst viele Schwächen in einem Dokument zu finden. Innerhalb des Prozesses gibt es typischerweise eine Phase, bei der die Gutachter das Produkt durchsehen und sich damit vertraut machen. Anschließend gibt es eine Gruppenphase in der das ganze Team nach Mängeln sucht. [42]

[43] verwenden eine Variante dieses Verfahren zur Erhebung von mentalen Modellen. Das Inspektionsverfahren wurde abgewandelt und durch Checklisten, eigenständige Zeitangaben, Fragebögen und Notizen von Beobachtern ergänzt.

In Phase eins beantworten Probanden einzeln Fragen zur Software mit Ja oder Nein (Checkliste). Zur Beantwortung der Frage kann und sollte die Software benutzt werden, da zusätzlich notiert werden soll, welche Handlungen zu der Antwort geführt haben und wieviel Zeit die Durchführung in Anspruch genommen hat.

Anschließend werden in einer 40 minütigen Phase zwei, die Antworten in zweier oder dreier Teams verglichen, ebenfalls unter Verwendung der Software. Unterschiedliche Antworten werden aufgelöst und Ergebnisse zusammengetragen. Diese umfasst ebenfalls eine Beschreibung der durchgeführten Aktionen und eine Begründung, warum einer oder mehrere der Probanden

falsche Antwort gegeben hat. Während dieser Zeit können Beobachtungen notiert werden. Abschließend soll innerhalb von 5 Minuten ein Fragebogen für eine Art Nachbesprechung beantwortet werden. Jedoch sind weder Inhalt noch genaues Ziel des Fragebogens erläutert.

Bei diesem Verfahren sind durch die Auswertung der Einzelantworten (inklusiver der Durchgeführten Handlungen) und deren Veränderung in der Gruppenphase Rückschlüsse auf die verwendeten mentalen Modelle möglich.

### 3.1.1 Anwendbarkeit

Das Verfahren, wenn auch sehr effizient und effektiv, ist zur Erhebung von konkreten mentalen Modellen von Saros nur bedingt geeignet. Grund dafür ist a) der Mangel an Validität der Studie [43, S. 131] und b) eine fehlende konkrete Beschreibung, wie aus den gewonnen Daten das mentale Modell konstruiert werden kann.

Das größte Problem bei dieser Methode besteht jedoch darin, dass bei nicht gelösten Aufgaben der Checkliste durch mehrere Team-Mitglieder nicht die Ursache für das Problem gefunden werden kann, sondern lediglich klar wird, dass an dieser Stelle anscheinend eines besteht. Sollten aus diesem Grund mehrere Durchläufe des Verfahrens notwendig sein, so ist das Argument der Effizienz und Effektivität hinfällig.

### 3.2 Instance-based cognitive Mapping

Cognitive Mapping ist ein Prozess, mit dem wichtige Eigenheiten eines mentalen Modells in einer spezifischen Domäne beschrieben und festhalten werden können. Zu diesen Eigenheiten zählen Annahmen, bestimmte Schlüsselfaktoren (konzeptuelle, logische, physische) und deren Beziehung zueinander (zeitlich, kausal, räumlich). [30]

[30] präsentieren eine Technik namens instance-based cognitive mapping (ICM), zu deutsch Fall-basierte Kognitive Karten. Diese Technik basiert auf den folgenden vier Annahmen:

- Personen können zumeist ihr mentales Modell nicht beschreiben, es kann lediglich durch Beoabachtungen von Entscheidungen und Handlungen hypothetisiert und getestet werden.
- 2. Innherhalb einer speziellen Domäne werden Entscheidungensalternativen auf der Grundlage von bestimmten Schlüsselfaktoren evaluiert. Dabei führen unterschiedliche Werte von einen oder mehrerer dieser Faktoren zu unterscheidlichen Entscheidungen.
- 3. Wird eine Person mit einer domänenspezifischen Situation (einem konkreten Entscheidungsfall) konfrontiert, dann bestimmt sie zunächst die entsprechenden Werte der einzelnen Schlüsselfaktoren. Anschließend werden anhand des internen mentalen Modells die Alternativen evaluiert, die möglichen Auswirkungen abgeschätzt und letztendlich die beste Entscheidung getroffen.
- 4. Mittels einer induktiven Analyse von mehreren Entscheidungen durch eine einzelne Person innerhalb einer spezifischen Anwendungsdomäne kann ein mathematisches Modell abgeleitet werden. Dieses mathematische Modell repräsentiert das mentale Modell für

diese Domäne. Das heißt, das abgeleitete Modell ist die entsprechende explizite Form der implizit vorgenommenen Abwägungen von Alternativen.

Zusammengefasst schlagen [30] hier also ein Methode vor, die mentale Modelle durch eine mathematische Analyse der getroffenen Entscheidungen aufdecken soll.

Der eigentliche Prozess wird von [30] wie folgt beschrieben:

Die oben erwähnten Faktoren können anhand von firmeninternen Experten, externen Beratern, Fachzeitschriften und/ oder Artikeln generiert werden. Danach werden diese Faktoren in 10-20 Fälle gruppiert und mit sinnvollen Werten belegt. Diese Fälle werden Einzelpersonen oder Gruppen präsentiert, die je nach Fall (und unterschiedlicher Werte der Schlüsselfaktoren) eine Entscheidung treffen. Die Entscheidungen werden gespeichert und den der anderen Fälle auf inkonsistenzen überprüft. Wurde eine Entscheidung getroffen, die mit einer anderen im Konflikt steht, wird der Befragte (oder die Gruppe) nochmals frequentiert und gebeten, die Entscheidung zu überdenken, um so die Inkonsistenz zu beheben.

Die Werte der Schlüsselfaktoren als auch die getroffenen Entscheidungen werden dann als Datengrundlage für den *Group Method of Data Handling (GMDH)* Algorithmus verwendet. Anhand dessen wird berechnet wir konistent das mentale Modell einer Person ist.

Inkosistente Entscheidung geben wiederum Rückschlüsse über die Lücken im mentalen Modell.

## 3.2.1 Anwendbarkeit

Die instance-based Methode bringt viele gute Ansätze und Vorteile mit sich. So zum Beispiel der geringere Einfluss durch voreingenommene Personen und damit einhergehende subjektive Ergebnisse. Außerdem verspricht das Vorgehen durch die Automatisierung eine effizientere Durchführung und verarbeitet, durch das einbeziehen getroffener Entscheidungen, echte Verhaltensweisen.

Andererseits sind die Beschreibungen in [30] und [44] nicht ausreichend, um die Methode praktisch anzuwenden. Das liegt unter anderem daran, dass der angeführte GMDH-Algorithmus nicht wie angedeutet ein einzelner Algorithmus, sondern ein ganzes Spektrum von Algorithmen ist, genau genommen zwölf an der Zahl<sup>5</sup>. Je nach Art der verwendeten Variablen für die Schlüsselfaktoren (stetig/ diskret oder parametrisch/ nicht parametrisch), ist ein anderer Algorithmus anzuwenden. Die Wahl des entsprechendens Algorithmus benötigt also ein tieferes mathematisches Verständnis.

Hier lauert auch das nächste Problem: Der vorgestellte Algorithmus basiert auf konkreten Zahlenwerten der Schlüsselfaktoren. Das heißt, die Einflussfaktoren müssen mit konkreten numerischen Werten belegt werden, um die Konsistenz der mentalen Modelle berechnen zu können. Das dies für kompliziertere Fälle als den angeführten Beispielen funktionieren, aber etwas Kreativität erfordern soll, wird nur als Hypothese in den Raum gestellt, aber nicht belegt.

 $<sup>^5</sup>$ http://www.gmdh.net/GMDH\_alg.htm

# 3.3 Struktur-Lege-Technik

"Die [Struktur-Lege-Technik] SLT basiert auf [..] einer Konzeptualisierung des Interviews durch Begriffs- und Relationskarten, einer mit diesen Kärtchen durchgeführten Legung der Struktur (Mentales Modell) und einer abschliessenden Verifizierung des Modells." [27, S. 277]

Das heißt, bei der SLT werden zentrale Begriffe assoziativ miteinander verbunden und auf eine inhaltslogische Struktur abgebildet [45].

Die SLT läuft in mehreren Schritten ab [46]:

- 1. Dem Teilnehmer wird erläutert, was erarbeitet werden soll und wie.
- 2. Dann werden wichtige Begriffe gesammelt und auf Karten geschrieben.
- 3. Anschließend werden die Karten in übergeordnete Konzepte gruppiert.
- 4. Danach wird versucht diese Begriffe/ Konzepte in Beziehung zueinander zu setzen, bis alle in einem Netz verbunden sind.
- 5. Abschließend werden noch einmal alle Begriffe durchgegangen und lässt sich durch den Teilnehmer bestätigen, ob alles stimmig ist.

Dabei sollte zu jeder Zeit der Teilnehmer möglichst selbstbestimmt und unbeeinflusst Änderungen vornehmen können. Alle Begriffe und deren Beziehungen zueinander dürfen bis zum Schluss überarbeitet und anders angeorndet werden.

# 3.3.1 Anwendbarkeit

Der Vorteil der SLT liegt darin, dass während der Erhebungsphase sofort eine grafische Darstellung des mentalen Modells erzeugt wird und diese zudem direkt von den Anwendern kommt. Die ursprünglich aus der Pädagogischen Psychologie kommende Methode ist auf dem softwareergonomischen Gebiet neu und wurde bisher nur anhand von potentiellen Anwendern getestet [27].

### 3.4 Interviews

Das Interview ist die am häufigsten genannte und verwendete Methode zur Erhebung von mentalen Modellen [7, 47, 30, 1, 48].

Die durchgeführten Interviews der genannten Autoren unterscheiden sich kaum in ihrer Befragungsform. Es wurden immer persönliche Interviews mit offenen Fragestellungen geführt [7, 47, 48, 1]. Sollte dies nicht möglich sein, schlagen [1] eine telefonische Befragung vor. Bis auf [1] verwenden alle strukturierte Interviews mit einer Reihe von festen Anzahl von Fragen. [47] und [48] verpackten diese Fragen in je ein Szenario, welches den Kontext der Frage um eine Situationsbeschreibung und eine grafische Dartellung erweitert.

3.4 Interviews 25

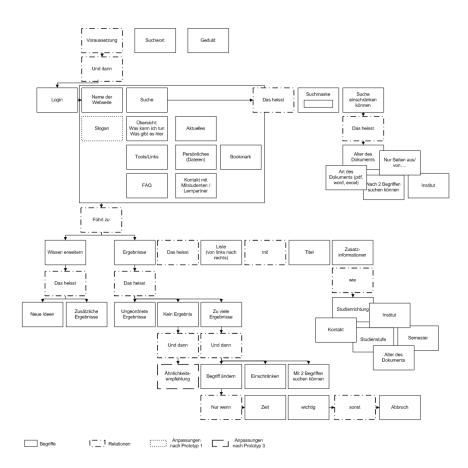


Abbildung 9: Strukturlegemodell (mentales Modell) eines Anwenders von einer webbasierten Such- und Orientierungsplattform für E-Learning-Ressourcen

## 3.4.1 Cognitive Mapping

Im Abschnitt 3.2 wurde dieser Prozess bereits vorgestellt. Neben der von [30] selbst entwickelten Methode, gibt es auch drei andere Varianten des Cognitive Mapping. Zwei davon unterscheiden sich zwar durch ihre Vorannahmen und Herangehensweisen, verwenden aber jeweils die Befragung als Erhebungsmethode.

# Die diagramm-basierte Technik

Diese Methode nimmt an, dass kausale und konzeptuelle Zusammenhänge und / oder Argumente einen Hinweis darauf bieten, wie das mentale Modell einer Person organisiert ist. Zudem sind sie Grundlage für Entscheidungen zwischen Alternativen. Bei dieser Technik werden Interviews mit offenen Fragen durchgeführt und transkribiert. Anschließend werden daraus relevante Konzepte, Argumente und Beziehungen extrahiert und in Diagrammen veranschaulicht. Laut [30] gehören zu dieser Kategorie unter anderem Kausal-Diagramme<sup>6</sup>, freies Card Sorting<sup>7</sup>, und Entscheidungsbäume.

### Die frame-based Technik

Dieser methodische Ansatz basiert auf der Annahme, dass die Wahrnehmung eines bestimmten Ereignisses stark durch vorherige Erfahrungen beinflusst wird. Diese vorangegangen Erfahrungen bilden eine hierarchische Struktur, auf deren Grundlage Entscheidungen getroffen werden. [30] Als Beispiel zeigt Abbildung 10 wie ein Unternehmens-Vizepräsident konkurrierende Kapitalprojekte in Erwartung (verschachtelt) eines bevorstehenden Capital Budget Meeting analysieren könnte. Solche Schemata werden mittels umfangreicher Interviews aufgedeckt und von "Experten" interpretiert.

```
MEETING
                                - Place
                                - Day
                                - Time
        CAPITAL BUDGET MEETING
                                        - CEO
                                        - Executive VPs
                                        - Other attendees
                PROJECTS
                                                Capital ($MM) Available
                                                Project rankings by IRR
                                                Project earnings risks
                        PROJECT CHAMPIONS
                                                        Forecasting tendency
                                                          (pessimistic/optimistic)
                                                        Forecasting technique
```

Abbildung 10: Frame-basierte Struktur eines Capital Budget Meeting

<sup>&</sup>lt;sup>6</sup>Bei Kausal-Diagrammen werden Knoten durch gerichtete Kanten verbunden, die die Richtung der kausalen Zusammenhängen darstellen.

<sup>&</sup>lt;sup>7</sup>Bei Card Sorting ordnen Anwender mit Begriffen versehene Karten in hierarchische Strukturen ein [39].

3.4 Interviews 27

#### 3.4.2 Strukturierte Interviews mit Szenarien

Strukturierte Interviews verwenden ein festes Set von Fragen, die in der Reihenfolge ungeachtet der Antworten gestellt werden [49, S. 97f.].

[47] und [48] haben die Fragen der Interviews durch eine Beschreibung der Situations und Screenshots oder Mock-ups<sup>8</sup> zu einem Szenario erweitert. Zu jedem Szenario wurden mehrere offene Fragen gestellt. Dabei stellten [47] Fragen der Form "What do you think the web site must do to...", um so herauszufinden, wieviel die Teilnehmer über die innere Funktionsweise der hypothetischen Applikation verstehen. [48] wählen einen anderen Ansatz und stellten Fragen nach dem eigenen Verhalten der Studienteilnehmer.

Bei der Analyse der transkribierten Interviews, sortierten [47] die Antworten in untersuchten, festgelegten Kategorien. Anschließen wurden die Antworten ausgewertet und das mentale Modell der jeweiligen Kategorie in textueller Form festgehalten. [48] untersuchten das abgefragte Verhalten nach gemeinsamen Ideen und diese kodiert. Ähnliche Ideen wurden zu einer Kodierungen zusammengefasst, so dass am Ende nur eine Liste mit eindeutigen Kodierungen existiert. Der Kodierungsprozess wurde erst beendet als keine neuen Ideen mehr gefunden wurden. Anschließend wurden die transkribierten Interviews nochmals durchgelesen und auf semantische Zusammenhänge der kodierten Ideen hin überprüft. Aus den Zusammenhängen der gruppierten Verhaltensweisen entstand das mentale Modell (siehe Abbildung 11).

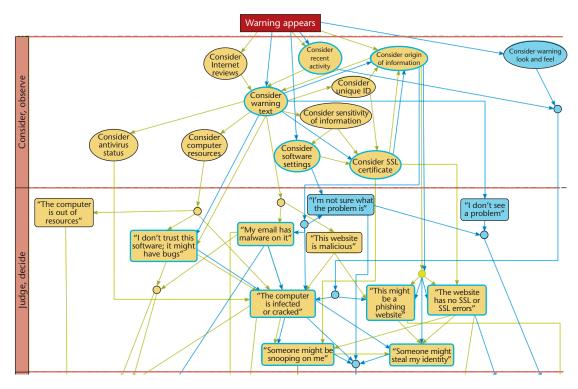


Abbildung 11: Ausschnitt des mentalen Modells von [48] zur Wahrnehmung und Reaktion auf Computer-Sicherheitshinweise

<sup>&</sup>lt;sup>8</sup>Ein Mock-up ist eine vereinfachte skizzenhafte Darstellung eines Benutzeroberfläche.

#### 3.4.3 Unstrukturierte Interviews

Im Gegensatz zu strukturierten Interviews folgen unstrukturierte Interviews keinem System von vordefinierten Fragen. Je nach Verlauf des Interviews können Fragen in einer anderen Reihenfolge gestellt, ausgelassen oder ergänzende Fragen hinzugefügt werden. [49, S. 98]

Dabei sind die Interviews von [1] nicht komplett ohne Struktur. Auch hier gilt es schließlich ein mentales Modell für eine Software oder einen Teil dieser zu erheben.

Zunächst werden dafür aufgabenbasierten Zielgruppen und der Umfang des Interviews bestimmt. Der Umfang bestimmt grob die Themen, die in den Interviews zum Tragen kommen sollen, also wovon ein mentales Modell erfasst werden soll. Interviews werden nicht mit konkreten Fragen und einer vorgegebenen Reihenfolge vorbereitet, sondern durch Stichworte oder Stichpunkte zu den einzelnen Themen. Nach erfolgreicher Transkribierung der Interviews, werden diese nach Äußerungen durchsucht, die konkretes Verhalten beschreiben. Die Verhaltensweisen werden - ähnlich wie bei [48] - zusammengefasst und in semantisch zusammenhängende Gruppen sortiert.

Zusätzlich können die gruppierten Verhaltensweisen mit den Funktionen eines System verglichen werden (siehe Abbildung 12). Dabei bezeichnet [1] beide Formen als mentales Modell, egal ob mit oder ohne Abgleich.

#### 3.4.4 Anwendbarkeit

Interviews sind sehr zeitaufwändig [30], bei der Analyse umso mehr, je unstrukturierte sie sind [49, S. 98]. Zudem können Ergebnisse von der Person abhängen, die die Befragung durchführt [30]. Andererseits scheinen sie ein populäres und probates Mittel zu sein, welches zudem zu vielversprechende Ergebnissen führt.

# 3.5 Auswahl der Evaluationsmethode

Es gibt einige Evaluationsmethoden zur Erfassung von mentalen Modellen, jedoch eignen sich nicht alle zur Erfassung eines mentalen Modells von Saros als Werkzeug zur Zusammenarbeit oder zur Erhebung von einzelnen Saros-Konzepten.

Inspektion eignet sich auf Grund seiner mangelnden Validität und methodischen Lücken nicht zur Erhebung von mentalen Modellen von Saros und wird deshalb nicht weiter betrachtet.

Instance-based Cognitive Mapping würde für den weiteren Verlauf der Arbeit ein sehr hohes Risiko darstellen, da es bisher keine konkreten Hinweise für eine allgemeine Umsetzung der Methode gibt. Zudem beinhaltet die Beschreibung der Methode zu wenig Hinweise, um sie auf die Saros Konzepte zu übertragen.

Die zentrale Idee, das Verhalten von Anwendern durch Entscheidungen abzufragen, kann festgehalten und könnte auch innerhalb von Interviews angewendet werden.

Struktur-Lege-Technik ist sehr vielversprechend, jedoch gibt es für diese recht junge Methode bisher wenig empirische Studien. Aus diesem Grund soll diese Methode nur in einem kleinen Rahmen und für potentielle Nutzer zur Anwendung kommen.

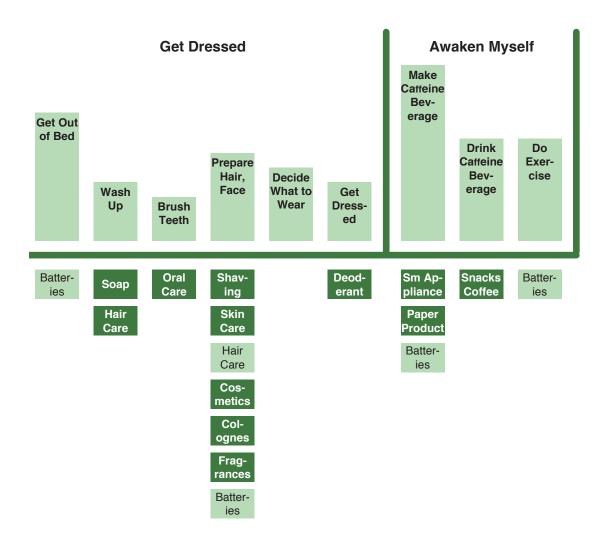


Abbildung 12: Ausschnitt eines mentalen Modells von [1] zum Verhalten nach dem Aufstehen verglichen mit der Produktliste von Procter & Gambles Webseite

**Interviews** sind trotz ihres enormen zeitaufwandes ein probates Mittel zu Erfassung von mentalen Modellen.

Um den möglichen Ergebnis-Radius nicht einzuschränken, aber auch die Auswertung im Rahmen dieser Arbeit durchführen zu können, scheint ein Mittelweg über semistrukturierten Interviews der richtige.

Unter Betrachtung der Evaluationsmethoden zur Erhebung von mentalen Modellen, wird das semi-strukturierte Interview die zentrale Methode dieser Arbeit darstellen.

Vor allem vorstrukturierte Interviews weisen den Makel auf, dass sie sich nur Ausschnitte eines Gesamtkonzepts oder nur auf die vorgegebenen Kriterien beschränken. Aus diesem Grund wird diese Interview-Form hauptsächlich zur Beleuchtung der konkreten vorausgewählte Saros-Konzepte verwendet werden (Ebene 2). Dabei wird ein szenario-/ handlungsbasierte Ansatz nach [48] zum Tragen kommen, der die Anwender nach ihrem eigenen Verhalten in bestimmten Situation fragt.

Zusätzlich fließt die Methode der unstrukturierten Interviews von [1] mit ein, um das allgemeine Verständnis von Zusammenarbeit (Ebene 1) zu erheben. Die Umsetzung soll jedoch nur in einer abgeschwächten Form durchgeführt werden, da ich auf keine Interview-Erfahrungen zurückgreifen kann. Deshalb werden hier Fragen vorbereitet und nicht nur Stichtworte zu einzelnen Themen.

Bei der Durchführung der Interviews gilt es besonders darauf zu achten, den Studienteilnehmern den nötigen Freiraum für thematische Ausführungen in andere Richtungen als den vorbereiteten Fragen zu geben.

# 4 Erhebung

In Kapitel 3 wurde das semi-strukturiere Interview als Methode zur Erhebung der mentalen Modelle von Saros bestimmt. Dieses Kapitel stellt die Vorbereitung, Durchführung und Auswertung dieser Interviews vor.

Nachdem festgelegt wurde "wie" das mentale Modell erfasst werden soll, beschäftigt sich Kapitel 4.1 mit den Fragen nach der Zielgruppe von Saros ("wer") und dem genauen Inhalt der Interviews ("was"). Kapitel 4.2 beschreibt das verwendete Material, die Versuchspersonen und die Versuchsdurchführung der Interviews.

# 4.1 Vorbereitung

Die Vorbereitung der Interviews besteht aus zwei Teilen.

Zunächst wird die relevante Zielgruppe von Saros bestimmt (siehe Kapitel 4.1.1). Im Anschluss daran wird in Kapitel 4.1.2 der eigentliche Inhalt der Interviews festgelegt.

Dabei zählt zum Inhalt nicht nur "was" gefragt wird, sondern auch die Form der Fragestellung. Zudem werden die herausgearbeiteten Fragen die zuvor angesprochenen zwei Ebenen (abstrakt und konkret) adressieren.

Das vollendete Interview ist in vier Teile gegliedert, welche in Tabelle 2 dargestellt sind. Der vollständige Leitfaden mit den ausformulierten Fragen kann auf Seite 101 unter Anhang A.1.2 nachgeschlagen werden.

Zielgruppe - Allgemein
 Saros - Allgemein
 Saros - Konzepte
 Demographie

Tabelle 2: Übersicht über die einzelnen Teile des Interviews

### 4.1.1 Bestimmung der Zielgruppe

Im Allgemeinen möchte man mit einem Produkt möglichst viele an Menschen erreichen. Um aber ein zufriedenstellendes Produkt für all diese potentiellen Kunden zu entwickeln, sollte man nicht versuchen es allen recht zu machen. Das Ergebnis könnte ein Produkt sein, was im schlechtesten Fall keinem hilft (siehe Abbildung 13). Der beste Weg um also eine Vielzahl von Abnehmern anzusprechen ist, das Produkt für eine explizite Gruppe mit ganz bestimmten Bedürfnissen zu entwickeln. [50, S. 77]

[1] und [50] setzen bei der Bestimmung von Zielgruppen an recht ähnlichen Stellen an. Im Gegensatz zu traditionellen Vorgehen legen sie den Fokus nicht auf demographische Faktoren oder Persönlichkeitstypen, sondern rücken das Verhalten von Anwendern in den Vordergrund.

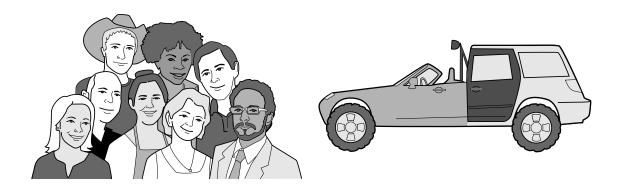


Abbildung 13: Beispiel für ein Auto, welches möglichst vielen Fahrern gefallen soll, eine Vielzahl von Funktionen vereint, aber letztendlich keinen Fahrer anspricht [50].

Da vor den ersten Interviews meist noch nicht sicher ist, welche genauen Verhaltensweisen Anwender aufweisen und wie diese sie unterscheiden, setzt [1] zunächst auf *Brainstorming*<sup>9</sup> und schlägt folgende Schritte vor: [1, S. 45ff.]

#### 1. Auflisten von unterschiedlichem Verhalten

Es gilt in groben Zügen alle Möglichkeiten aufzuzeigen, wie sich unterschiedliche Menschen verhalten könnten.

# 2. Gruppieren des Verhaltens

Verhalten wird zu Verhaltensweisen zusammengefasst und gruppiert.

## 3. Benennen der Gruppen

Für die Gruppen werden provisorische Namen vergeben.

Dieses aufwendige Vorgehen - so simpel es auf den ersten Blick scheint - wird als Prozess beschrieben, der möglichst im Team durchgeführt werden soll. Das erforderliche Brainstorming unter 1. sollte viele und wenn möglich alle denkbaren Verhaltensweisen abdecken, was von einer Einzelperson nur schwer zu bewältigen ist.

[50] hingegen betont, dass selbst bei der Bestimmung der Interview-Kandidaten schon auf vorhandene Daten zurückgegriffen werden sollte. Zudem dürfe weder bei den Interviews noch bei der Erfassung der Kandidaten nur das Verhalten betrachtet werden. Vielmehr gelte es die Ursachen für das Verhalten zu verstehen: Die Ziele.

Ziele sind die Ursache und der Motivator für unser Verhalten und sollten deshalb noch vor den Verhaltensweisen selbst erhoben werden. [50, S. 88f.]

# Auswertung des Saros Fragebogens

Saros erhebt auf zwei unterschiedlichen Wegen Daten von Nutzern. Auf dem indirekten Weg, indem die Sitzungsdaten der Anwender unter Voraussetzung von deren Einverständnis gesammelt werden und auf dem direkten Weg, der es Anwendern erlaubt, einen Fragebogen auszufüllen<sup>10</sup>.

<sup>&</sup>lt;sup>9</sup>Brainstorming ist eine Technik, die der Risiken-, Ideen- und Lösungs-Findung dient [51, S. 368].

<sup>&</sup>lt;sup>10</sup>http://saros-build.imp.fu-berlin.de/phpESP/public/survey.php?name=SarosFastUserFeedback\_1

Für die Auswertung wurde der Datensatz von 80 ausgefüllten Fragebögen verwendet, die von Anwendern zwischen 03.07.2011 und 27.03.2012 beantwortet wurden.

Eine der in diesem Fragebogen gestellten Fragen adressiert eben diese Ziele der Anwender und lautet:

"Which goals did you want to accomplish in your latest Saros session?"

Da noch keine typischen Verhaltensweisen der Anwender als Daten vorliegen, kommen die in den Antworten artikulierten Ziele dem am nächsten, da es sich bei Zielen um die Ursache für das Verhalten handelt [50, Vgl. S. 88f.]. Demnach stellen die Freitext-Antworten auf diese Frage einen ersten guten Ansatz zur Bestimmung von Interview-Kandidaten dar, der zudem auf realen Daten beruht.

Bei der Auswertung der Daten wurden die verschiedenen Antworten, nach einem ähnlichen Schema wie zuvor von [1] beschrieben, inhaltlich zusammengefasst.

Leider ist es schwierig auf diese Frage präzise, ehrlich oder gut artikuliert zu antworten [50, S. 88]. Das heißt, es ist nicht klar, ob es sich bei der Antwort wirklich um ein artikuliertes Ziel, nur ein Verhalten (task) oder etwas anderes handelt. Aus diesem Grund wurden die Gruppen zusätzlich auf drei Zieltypen hin untersucht und kategorisiert. Diese umfassen experience goals, end goals und life goals [50, S. 92ff.] (siehe Tabelle 3).

**Experience goals** sind einfach und persönlich. Sie drücken aus, wie sich jemand fühlen möchte während er ein Produkt benutzt.

Beispiele: Sich intelligent fühlen; Spaß haben; fokusiert und aufmerksam bleiben oder sich entspannt, cool oder hip fühlen.

End goals repräsentieren die Motivation von Anwendern eine bestimmte Aufgabe durchzuführen. Wenn man ein Mobiltelefon in die Hand nimmt oder ein Word-Dokument öffnet, hat man zumeist ein solches Ziel vor Augen.

Beispiele: Sich Problemen bewusst sein, bevor sie eintreten; in Verbindung mit seinen Freunden und der Familie bleiben; Abarbeiten der eigenen ToDo-Liste bis 17 Uhr oder Finden von guter Musik.

Life goals sind persönliche Bestrebungen die typischerweise über den Kontext des Produkts hinausgehen. Diese Ziele sind der Antrieb und die Motivation, die helfen zu verstehen, warum an Anwender versucht bestimmte end goals zu erreichen.

Beispiele: Ein schönes Leben führen; Erfolgreich eigene Bestrebungen erfüllen; Attraktiv und beliebt sein, bzw. von anderen respektiert werden.

Traf keines der aufgeführten Ziele auf die Antworten der Kategorie zu, so wurde dies entsprechend vermerkt.

Die Auswertung des Fragebogens befindet sich unter Anhang A.1.1 auf Seite 99. Sie enthält sowohl die Auflistung aller 80 Antworten als auch die vollständig kategorisierte Liste nach Tabelle 3.

### Weitere Einflussfaktoren

Neben den Zielen, die eine Ursache für Verhalten darstellen, gibt es auch noch eine Reihe weiterer Einflussfaktoren, die zur Bestimmung der Zielgruppe von Saros betrachtet werden

Category	Goal Type	Answer
teaching / learning	End Goal	"I started learning how to make games in python with a friend!" "Help newb with programming" "I helped someone with his problems in Java by tutoring through his project."
work on school / uni assignments	End Goal	"Solving an exercise task with my exercise partner. We didn't find a meeting-date, so we used saros - with success." "to complete our homeworks for university" "Scolar Coding"
free time: games & apps	End Goal	"We edit multiple classes for the [android] app" "We wanted to code a Minecraft mod." "Start coding our new game, The First Towner"
company	Goal or Task	"Pair programmer with remote developer in same company."
first time usage	Goal or Task	"Try out saros and my new jabber server that was created for this purpose" "Have been curious about Saros for a long time and finally sat down to try it out."
debugging	End Goal	"Fix some a script." "We want to fix different classes, which was very easy to do with this plugin." "Have a friend help me debug some code."
review	End Goal	",collaborative code review" ",chat and revision"
have fun	Experience Goal	"FUN"

Tabelle 3: Antworten auf die zweite Frage des Saros-Fragebogens - Kategorisiert nach Zielen und Zieltyp (Auszug)

#### müssen.

An dieser Stelle kam die von [1] zur Sprache gebrachte Brainstorming-Technik zum Einsatz. Neben den Zielen führt [50] mit den folgenden zwei - für Saros relevanten - Kategorien von Einflussfaktoren auf. Dazu gehören Verhaltensvariablen und bestehende Fähigkeiten im Bereich der Domäne und verwendeten Technologie.

### Verhalten und Verhaltensvariablen

Verhaltensvariablen sind nicht mit dem eigentlichen Verhalten gleichzusetzen oder zu verwechseln. Bei den Variablen handelt es sich - wie bei den Zielen - ebenfalls um Ursachen für bestimmtes Verhalten.

- Verhaltensvariablen umfassen bei Saros die Nutzungshäufigkeit, die Nutzungsart (Anzahl der Teilnehmer, Art der Projekte, Größe der Projekte) und den Nutzungskontext (Umgebung und verfügbare Hardware).
- Zu den Saros entscheidenden Fähigkeiten bezüglich der Domäne und Technologie zählt das Können in den Bereichen Eclipse und dessen Plugins, von verteilter Programmierung, der vorherrschenden Sprache Englisch und allgemeine Computerkenntnisse.

## Festlegung der Zielgruppe

Unter Hinzunahme der zwei zusätzlichen Einflussfaktoren, wurden die Kategorien aus dem Fragebogen in Benutzergruppen überführt. Diese sind gemeinsam mit den Zielgruppen in Tabelle 4 zusammengefasst.

	${\bf Benutzer gruppe}$	Zielgruppe
(1)	professioneller Einsatz	X
(1)	im Arbeitsbereich	Λ
(2)	Bearbeitung von Aufgaben	
(2)	im schulischen oder universitären Rahmen	X
(3)	Freizeitprogrammierung	A
(4)	Lernen und Lehren	

Tabelle 4: Festlegung der Saros-Zielgruppen anhand erhobener Benutzergruppen

### Benutzergruppen und Zielgruppen

Benutzergruppen bezeichnen alle Anwender, die Saros verwenden. Im Gegensatz dazu sind Zielgruppen, die Anwender, auf deren Bedürfnisse bei der Entwicklung von Saros insbesondere Rücksicht genommen wird.

Nicht alle Kategorien aus Tabelle 3 sind in obiger Tabelle 4 als Benutzergruppe aufgeführt. Dies liegt zum Einen daran, dass es sich bei Tabelle 3 nur um einen Auszug handelt und sie deshalb unvollständig ist und zum Anderen daran, dass sich die einzelnen Kategorien unter

Berücksichtigung der zusätzlichen Einflussfaktoren nicht immer eindeutig einer Benutzergruppe zuordnen lassen.

Mitglieder jeder Benutzergruppe können Saros zum ersten Mal benutzen ("first time usage"), Durchsichten durchführen ("review") oder auch nach Fehlern suchen ("debugging").

Die Entscheidung, welche Benutzergruppen im weiteren Verlauf als Zielgruppen betrachten werden sollen, wurde unter Abstimmung mit der Arbeitsgruppe "Software Engineering" getroffen. Sie nehmen durch ihren Standpunkt gegenüber dem Produkt Saros die Stakeholder-Rolle ein (siehe Kapitel 1.1), die eine solche Entscheidung für den zukünftigen Weg des Eclipse-Plugins treffen sollten.

Gleiches gilt für die Entscheidung die Benutzergruppen (2) und (3) zu einer Zielgruppe zusammenzufassen und als eine zu untersuchen.

Für den weiteren Verlauf der Arbeit werden nur noch die Zielgruppen (1) und (2+3) betrachtet und unter den Namen Arbeit und Freizeit referenziert. Diese sind in Tabelle 4 mit einem X gekennzeichnet.

# 4.1.2 Bestimmung des Interviewumfangs

Zur Bestimmung des Inhalts der Interviews wurden sowohl Fragen der abstrakten Ebene als auch für die konkrete Ebene entworfen. Dabei kommen die in Kapitel 3.5 ausgewählten Methoden zum Einsatz.

Für die Fragen nach Konzepten von Saros ist dies der szenariobasierte Ansatz mit handlungsorientierten Fragen nach [48] (konkrete Ebene). Die abstrakte Ebene wird - ähnlich der Herangehensweise von [1] - mit allgemeinen, möglichst offenen Fragen zur Zusammenarbeit vorbereitet.

Im Folgenden wird die Vorgehensweise zur Bestimmung der Interview-Fragen vorgestellt.

## Abstrakte Ebene

Ausgangspunkt für die Fragen zum allgemeinen Verständnis der Zusammenarbeit stellen die Einflussfaktoren dar, die auch zur Bestimmung der Benutzergruppen herangezogen wurden (Verhaltensvariablen und Fähigkeiten).

Zu den Einflussfaktoren der Kategorie "Verhaltensvariablen" gehören die Nutzungshäufigkeit, die Nutzungsart und der Nutzungskontext. Diese werden im weiteren Verlauf unter dem Begriff Aktivitäten zusammengefasst. Wie im vorherigen Abschnitt 4.1.1 bereits angeführt, setzen sich diese Aktivitäten in Bezug auf die Zusammenarbeit aus folgenden Teilaspekten zusammen:

- Nutzungshäufigkeit: Häufigkeit und Dauer
- Nutzungsart: Anzahl der Teilnehmer, Art der Projekte, Größe der Projekte
- Nutzungskontext: Arbeitsumfeld und verfügbare technische Hilfsmittel

Äquivalent dazu schlüsselt sich die Kategorie der Fähigkeiten in die nachstehenden Teilaspekte auf:

• Domäne: verteilte Programmierung, Englisch

• Technologie: Eclipse und Eclipse-Plugins, allgemeine Computerkenntnisse

Auch wenn [50] diese Kategorien vorgeschlägt und die einzelnen Aspekte sinnvoll erscheinen, so sind sie doch rein hypothetischer Natur und müssen im Rahmen der Interviews verifiziert oder falsifiziert werden. Das heißt, mittels der Antworten zu diesen Aspekten lässt sich später feststellen, ob sich die Zielgruppen wirklich durch diese Aspekte unterscheiden oder nicht. Ferner stellen denkbare Fragen zu diesen Kategorien eine solide Alternative zu den von [1] vorgeschlagenen themenbezogenen Stichworten dar (siehe Kapitel 3.4). So wären Fragen zum Kontext wie dem Arbeitsumfeld relevant für das allgemeine Verständnis der Zusammenarbeit und gleichzeitig unabhängig von jeglicher Technologie die dafür eingesetzt wird - wie zum Beispiel Saros.

Auf dieser Grundlage wurden die Fragen für die Kategorien "Aktivitäten" und "Fähigkeiten" konstruiert und für jeden Teilaspekt ausformuliert (siehe Tabelle 5 und 6).

	Aktivitäten
Nutzungshäufigkeit	<ul><li>Wie häufig arbeiten Sie zusammen? (pro Woche)</li><li>Wieviele Stunden arbeiten Sie zusammen? (pro Einheit)</li></ul>
Nutzungsart	<ul> <li>Wieviele Kollegen?</li> <li>Wieviele Projekte?</li> <li>Welche Projekte? (Details wie Art, Sprache, Größe)</li> <li>Woher wissen Sie, wann Ihr Kollege mit Ihnen zusammenarbeiten möchte?</li> </ul>
Nutzungskontext	<ul><li>Würden Sie kurz Ihr Arbeitsumfeld beschreiben?</li><li>Welche technischen Mittel stehen Ihnen für die Zusammenarbeiten zur Verfügung?</li></ul>

Tabelle 5: Interviewfragen zu Aktivitäten

Zusätzlich schlägt [50] eine Reihe von Themen vor, die ebenfalls mit in ein Interview einfließen sollten. So zum Beispiel sogenannte ziel- und systemorientierte Fragen [50, S. 66]. Zunächst soll hier nur auf die zielorientierten Fragen eingegangen werden. Die systemorienterte Fragen beziehen sich naturgemäß auf Saros und gehören deshalb in den Bereich der technologieabhängigen Themen. Diese Art von Fragen werden an späterer Stelle nochmals aufgegriffen (siehe Seite 39).

Fragen nach den Zielen der Anwender sind darauf ausgelegt die Motivation für bestimmtes Verhalten zu ergründen und eignen sich deswegen für diesen Bereich des Interviews ausgezeichnet. Tabelle 7 listet die Fragen nach dem Grund für die Zusammenarbeit auf.

Des Weiteren wurde noch eine Frage nach der subjektive Einstellung hinzugefügt, die Auskunft über die Haltung zur Zusammenarbeit geben soll und somit den Versuchspersonen eine Möglichkeit gibt, frei ihre Meinung zu äußern (siehe Tabelle 8). Außerdem könnte diese Frage mehr Hintergrundinformationen oder sogar eine Erklärung für andere Antworten liefern. Wurde beispielsweise angegeben, dass die Zusammenarbeit mit Kollegen nur selten stattfindet (Aktivitäten: Nutzungshäufigkeit), dann liefert die Antwort auf die Frage der Einstellung zur

	Fähigkeiten
Computerkenntnisse	<ul><li>- Wieviele Stunden pro Woche?</li><li>- Wofür nutzen Sie den PC bei der Arbeit / in der Freizeit?</li></ul>
Eclipse	<ul><li>- Wieviele Stunden pro Woche?</li><li>- Wofür?</li></ul>
verteilte Programmierung	<ul> <li>Wieviel Erfahrung haben Sie mit dieser Form der Zusammenarbeit?</li> <li>Wie läuft die Zusammenarbeit mit Ihren Kollegen üblicherweise ab?</li> <li>Wer macht von Ihnen was?</li> </ul>
Englisch	- Abgesehen von deutsch, welche anderen Sprachen verstehen Sie?

Tabelle 6: Interviewfragen zu Fähigkeiten

# Ziele (Motivation)

- Warum arbeiten Sie mit Ihren Kollegen zusammen?
- Was macht Ihnen bei Ihrer (Zusammen-)Arbeit am meisten Spaß?
- Was würden Sie vorziehen nicht zu tun?
- Wenn Sie mit Ihren Kollegen zusammenarbeiten: Woran arbeiten Sie dann?

Tabelle 7: Interviewfragen zu Zielen

Zusammenarbeit unter Umständen einen Grund dafür.

## Einstellung

Was halten Sie von dieser Form der Zusammenarbeit?

Tabelle 8: Interviewfrage zur Einstellung

Tabelle 9 fasst abschließend noch einmal alle vorgestellten Frage-Kategorien in einer Übersicht zusammen.

Alle Fragen dieser Kategorien haben zwei Ziele: Sie sollen das Anwender-Verständnis von Zusammenarbeit erheben und klarstellen, ob die zuvor getroffene Unterscheidung der Zielgruppen sinvoll war.



Tabelle 9: Überblick über alle Frage-Kategorien zur Zielgruppe

#### Vom Abstrakten zum Konkreten

Bevor im nächsten Schritt die Fragen für die konkreten Saros-Konzepte ausgearbeitet werden, wird in diesem Abschnitt auf die bereits erwähnten systemorientierten Fragen eingegangen. Fragen diesen Typs sollen nach [50] Aufschluss über signifikante Verhaltensmuster geben und sind deshalb ebenfalls von Interesse.

Die Fragestellungen nach [50, S. 66] lesen sich wie folgt:

**Function** What are the most common things you do with the product?

**Frequency** What parts of the product do you use most?

**Preference** What are your favourite aspects of the product? What drives you crazy?

Failure How do you work around problems?

**Expertise** What shortcuts do you employ?

Da es sich bereits um produktbezogene Präferenzen, Nutzungshäufigkeiten und Verhaltensweisen handelt, zählen Fragen dieser Art eigentlich zur konkreten Ebene. Andererseits besitzen sie auch das Potential Rückschlüsse für die Zusammenarbeit zu ziehen. Der Grund dafür liegt in den Fragen selbst, die unter anderem darauf abzielen Probleme aufzudecken. Dabei liefern eben diese Probleme die Antworten darauf, wie typischerweise zusammengearbeitet wird. Probleme treten erst auf, wenn das eigene Vorgehen bei der Zusammenarbeit und die damit

einhergehenden Erwartungen und Ansprüche nicht deckungsgleich mit der Software sind. So kann erwartet werden, dass auf Fragen nach *Preference* und *Failure* Probleme geschildert und Wünsche geäußert werden. Diese haben ihren Ursprung in der Art, wie sonst üblicherweise zusammengearbeitet wird.

Damit stellen diese Fragen für das Interview einen idealen Übergang von den allgemeinen Fragen zur Zusammenarbeit hin zu den konkreten nach den Saros-Konzepten dar. Dieser Abschnitt wird deshalb weder der abstrakten noch der konkreten Ebene zugeordnet.

Tabelle 10 zeigt einen Überblick über die Frage-Kategorien, die innerhalb dieses Abschnitts abgedeckt werden. Für die ausformulierten Fragen sei auf den vollständigen Interviewleitfaden im Anhang A.1.2 Seite 101 verwiesen.



Tabelle 10: Überblick über alle Frage-Kategorien zu Saros

## Konkrete Ebene

Die Interview-Fragen nach den Saros-Konzepten gründen auf einer Analyse der Abschlussarbeiten von [11] und [12]. Für die Auswahl der zu untersuchenden Konzepte und Funktionen wurden also keine neuen Daten erhoben. Vielmehr wurde auf bestehende Erkenntnisse und verfügbare Aufzeichnungen zurückgegeriffen (mehr Details zur Datengrundlage in der nachfolgenden Infobox).

### **Datenbasis**

- [11] hat sich in seiner Arbeit mit den Awareness-Informationen von Saros auseinandergesetzt. Er untersuchte wie prominent diese sind und ob sie überhaupt wahrgenommen wurden. Ein Ergebnis seiner Arbeit ist, dass dem "Führenden" beim Follow-Modus zu wenig Informationen bereitstehen, da dieser Person nicht immer klar ist, wer ihm oder ihr folgt. Zudem sei "das Verhalten des Follow-Modes dem Entwickler unklar gewesen" [11, S. 21].
- [12] hat neben den Erkenntnissen von [11] auch andere Methoden genutzt, um Usability-Probleme zu identifizierten und zu beheben. Eine dieser Methoden waren Usability-Tests. Bei dieser empirischen Methode werden Anwender bei der Bewältigung realer Aufgaben unter Verwendung der zu untersuchenden Software beobachtet [9, S. 27]. [12] hat die Nutzer nicht nur dabei beobachtet, sondern sowohl die Anwender selbst als auch deren Bildschirminhalte digital aufgezeichnet.

Zur Bestimmung der zu untersuchenden Saros-Konzept wurde zunächst eine tabellarische Übersicht erstellt, die alle Konzepte nach dem Ort ihrer Darstellung auflistet (siehe Abbildung 14).

So wurden zum Beispiel die Informationen um den Follow-Modus zum Saros-View und dort in den Bereich Session eingeordnet. Der linke Viewport dementsprechend im Bereich Vertical Ruler unter Editor.

Anschließend wurde die Arbeit von [11] auf Erkenntnisse zu den von ihm ermittelten Awareness-Informationen untersucht. Waren relevante Informationen angegeben, wurde dies in der Tabelle beim entsprechenden Saros-Konzept vermerkt.

Daraufhin wurden die Video-Aufzeichnungen der Usability-Tests (UT) von [12] analysiert. Wurde im Rahmen der Tests ein Problem mitgeschnitten, so wurden die Informationen in der Tabelle um Folgende erweitert:

- Ein Zitat der Anwender oder eine szenarienhafte Beschreibung des Problems,
- die Nummer des UTs inklusive einer sekundengenauen Zeitangabe und
- ob das Konzept oder die Funktionalität neu hinzugekommen ist.

Abbildung 14 auf Seite 43 zeigt eben diese tabellarische Übersicht. Sie enthält neben den genannten Informationen noch eine Reihe weiterer Anmerkungen, die im Folgenden beschrieben werden.

Im nächsten Schritt wurden alle Tabelleneinträge nach ihrer Fatalität bewertet, um so die Probleme nach ihrem Ausmaß zu sortieren.

Zur Bewertung der Fatalität wurde das von [9] und [12] vorgestellte Bewertungssystem verwendet<sup>11</sup>. Bei diesem System werden die Probleme auf der Grundlage folgender Faktoren bewertet: [9, S. 38]

Frequenz Tritt das Problem selten oder häufig auf?

Beispiel: Beim Versuch als Sitzungsteilnehmer ohne Schreibrechte Änderungen am Quelltext vorzunehmen, erscheint eine darauf hinweisende Meldung bei jeder Eingabe. Die Frequenz ist also sehr hoch.

Einfluss Wie stark wird der Anwender durch das Problem gestört?

Beispiel: Gehen Daten bei einer Saros-Sitzung verloren, stellt dies eine erhebliche Störung für den Anwender dar.

**Persistenz** Ist das Problem einmalig oder stört es den Anwender bei jedem Auftreten aufs Neue?

Beispiel: Für den Gebrauch von Saros ist ein XMPP-Konto notwendig. Dieses Usability-Problem ist immer vorhanden, verursacht aber nur einmalig Schwierigkeiten; nämlich dann, wenn man unerwarteter Weise davon erfährt und sich ein Konto einrichten muss.

Anhand dieser Bewertung lassen sich die Probleme auf eine Skala von 0 bis 4 abbilden und wie in Tabelle 11 beschreiben [9].

Lagen keine Informationen für einen Eintrag vor, so wurde dies unter Berücksichtigung der

<sup>&</sup>lt;sup>11</sup>Nielsen und Mack (1994) zitiert bei [9] und als Nielsen und Mack (1993) zitiert bei [12]

Skala mit einer -1 gekennzeichnet. Das heißt, es kann zu dem von Saros dargestellten Konzept keine Aussage getroffen werden, da es in der Datenbasis nicht erwähnt wird.

	Fatalität	Beschreibung
0	kein Usability-Problem	"I don't agree that this is a usability problem at all"
1	Kosmetisch	"Cosmetic problem only: need not to be fixed unless extra time is available"
2	Gering	"Minor usability problem: fixing this should be given low priority"
3	Schwer	"Major usability problem: important to fix, should be given high priority"
4	Katastrophal	"Usability catastrophy: imperative to fix this before the product can be released"

Tabelle 11: Einordnung und Beschreibung der Fatalität auf einer Skala von 0 bis 4 (nach [9, S. 37])

Zur besseren Orientierung wurde zudem das Farbschema von [9] für die Kennzeichung der Usability-Probleme verwendet. Tabelle 12 zeigt die Farbkodierung der relevanten Fatalitäten in einer vereinfachten Darstellung.

Auch die zusätzliche Spalte Schlüssel der Tabelle in Abbildung 14 dient der leichteren Orientierung. Sie enthält für jedes Saros-Konzept einen eindeutigen Bezeichner. Diese Bezeichner dienen der Nachverfolgung der Probleme über verschiedene Dokumente, Abbildungen und Tabellen hinweg.



Tabelle 12: Vereinfachte Darstellung der relevanter Fatalitäten (nach [9, S. 41])

Um die Länge des Interviews und damit die Zeit der Versuchspersonen nicht überzustrapazieren, wurden nur Probleme mit den Fatalitäten "Katastrophal" (rot) und "Schwer" (gelb) betrachtet. Bei Problemen diesen Schweregrades ist der eindeutige Bezeichner in fetter Schrift hervorgehoben.

Für diese priorisierten Probleme wurden möglichst realistische Szenarien und Fragen skizziert. Die folgenden zwei Fragen wurden bei der Herangehensweise als Hilfestellung benutzt:

- Was bedeutet die darstellte Information?
- In welchem Zusammenhang ist die Information relevant?

Zudem wurde bei der Formulierung der Szenarien und Fragen besonders darauf geachtet keine Saros spezifischen Begriffe oder Formulierungen zu verwenden. Nach dem User-Centered Design gilt "Speak the User's Language", weswegen die Terminologie der Benutzeroberfläche die Sprache der Anwender und nicht des Systems verwenden sollte [34, S. 123f.]

Ort der Information		Art der Info	Sinn der Info	Fatalität	Schlüssel	Quelle
Navigator/ Package/ Project Explorer	xplorer					
		grüner Punkt	fokusierte Datei; Welche Datei wird angeguckt?	4	OpenFile-1	nicht wahrgenommen (Solovjev, UT von Waldmann)
		gelber Punkt	offene Datei(en); Welche Datei(en) sind geöffnet?		OpenFile-2	nicht wahrgenommen (Solovjev, UT von Waldmann)
		Doppelpfeil-Symbol und "(shared)" bzw. "(partial share)"	synchronisierte Datei(en); Welche Projekte/ Dateien werden synchronisiert?	4	Session-5	NEU: textuelle Information nicht wahrgenommen (Solovjev, UT von Waldmann)
Vertical Ruler	ıl Ruler	Viewport Annotation links	sichtbarer Bereich bzgl. meines sichtbaren Bereiches; Welcher Ausschnitt ist sichtbar?	4	OpenFile-3	"der links läuft unter dem anderen" "kann man auch weglassen links, der rechts reicht doch" (UT 4 - Daniel 6:30)
Overview Ruler	w Ruler	Viewport Annotation rechts	sichtbarer Bereich bzgl. Datei; Welcher Ausschnitt ist sichtbar?	4	OpenFile-4	
editierbarer Bereich	rbarer eich	Cursor Indikation	aktuelle Cursor Position; Was wird fokusiert innerhalb des sichtbaren Bereiches?		OpenFile-5	"Siehst Du eigentlich meinen Cursor?" "Ich sehe den Cursor, aber wenn Du so markierst - eine Zeile - dann ist das besser. Manchmal sehe ich den nicht so eindeutig." (UT 10 - 9:10)
		Code Editieren	zuletzt geschriebener Code; Was wurde zuletzt geschrieben?	0	OpenFile-6	wird durch die farbige Hintergrund-Markierung schnell erkannt (UT 10 - Jerome 29:00)
		Code Markieren	markieren / zeigen	+	OpenFile-7	"Siehst Du eigentlich meinen Cursor?" "Ich sehe den Cursor, aber wenn Du so markierst - eine Zeile - dann ist das besser. Manchmal sehe ich den nicht so eindeutig." (UT 10 - 9:10)

Abbildung 14: Tabellarische Übersicht über die in Saros grafisch dargestellten Konzepte und deren Fatalitäten (Ausschnitt)

44 44 45 4 Erhebung

So wurde darauf verzichtet die englischen Begriffe wie Session, Saros-View, Host, Follow-Mode, Viewport oder deren deutsche Entsprechung wie Sitzung zu verwenden. Bei diesen Wörten handelt es sich um Begriffe, die Saros selbst und die Entwickler innerhalb des Saros Teams verwenden. Das heißt aber nicht, dass dies auch auf die Anwender zutreffen muss. Um diesen vermeintlich bestehenden Unterschied aufdecken zu können, müssen Szenarien und Fragen frei von solchen Vokabeln sein.

Sollte eine solche Differenz bestehen, so würde diese "Vorsichtsmaßnahme" einen weiteren Vorteil liefern. Neue und (hoffentlich) treffendere Begriffe müssen nicht erst gesucht oder erfunden werden (z. B. *Inviter* oder *Creator* anstelle von Host) [9, S. 93]. Durch das eigens verwendete Vokabular der Anwender, liefern diese schon neue Begriffe oder zumindest Ideen für diese. (Deutsche Wörter lassen sich nicht immer eins zu eins ins Englische übersetzen. *Creator* ist ein unglückliches Beispiel dafür: Das Nomen ist, in der englisch untypischen Form der Großschreibung, bereits mit einer religiösen Bedeutung konnotiert und bedeutet Schöpfer.)

Ein weiterer Grund für die Vorsicht bei den Begrifflichkeiten ist die Tatsache, dass unsere Sprache auch eng mit unseren mentalen Modellen verknüpft ist. Schon Nuancen können enorme Unterschiede ausmachen, wie wir etwas wahrnehmen, darüber denken und entsprechend artikulieren. Dabei denke man nur an typische Gender-Formulierungen. [1, Vgl. S. 110]

Unter Berücksichtigung der zwei angeführten Fragen und des Sprach-Aspektes wurden die handlungsorientierten Fragen im Szenario-Format formuliert.

Im Anschluss daran wurde überprüft, ob die Szenarien mehrere Lösungsmöglichkeiten als Antworten zulassen. Traf dies zu, so wurde versucht ein eindeutigeres Szenario zu finden. War dies nicht möglich, wurden alle Möglichkeiten als Musterlösungen vermerkt.

Abschließend galt es die formulierten Szenarien in eine semantisch sinvolle Reihenfolge zu bringen. Ziel war es der Versuchsperson möglichst selten eine neue Situation schildern zu müssen, damit möglichst wenig Gedankensprünge vollzogen werden müssen und dem Fluss der Situationen leicht zu folgen ist.

Ein Beispiel dafür ist das Host-Konzept, welches über den ganzen Lebenszyklus einer Sitzung hinweg zum Tragen kommt: Bei der Initiierung einer Sitzung, beim Vergeben von Rechten während einer laufenden Sitzung und beim Beenden dieser. Alle drei Situationen setzen sich mit dem gleichen Konzept auseinander. Doch scheint es wenig sinvoll alle Fragen in einem einzigen Szenario zu stellen. Aus diesem Grund wurden daraus zwei separate Szenarien geformt, wobei ein Abfrage zu Beginn und die Andere als letzte der handlungsorientierten Fragen gestellt wurde.

Anfang: Sie wollen mit einer weiteren Person zusammenarbeiten.

- Was machen Sie?
- Wie entscheiden Sie, wer von Ihnen wen einlädt?
- Was heißt das? (optional: nur wenn der Begriff "Host" fällt)
- Wie beeinflusst das Ihre Zusammenarbeit?

. . .

Ende: Wie haben Sie entschieden, wer von Ihnen als erster die Zusammenarbeit beendet?

Durch diesen Aufbau wurden die Versuchspersonen quasi durch eine virtuelle Sitzung geführt, die durch das erste Szenario (Anfang) gestartet und am Ende durch das zweite Host-Szenario auch wieder geschlossen wurde.

In Anlehnung an [47] und [48] (siehe Abschnitt 3.4.2) wurden für die Szenarien Screenshots von einer Eclipseumgebung mit Saros vorbereitet. Auf den Screenshots waren neben der abgefragten Information - wenn möglich - auch die anderen Lösungsmöglichkeiten abgebildet (siehe Anhang A.1.3 auf Seite 105).

Dabei sollten diese nur zum Einsatz kommen, wenn sich die Versuchsperson unsicher war oder nicht genau erinnern konnte. Die Screenshots wurden dann als Hinweis präsentiert. Hier wird auf die psychologische Tatsache zurückgegriffen, dass uns *Recognition* (das Wiedererkennen) leichter fällt als das bloße Erinnern (*Recall*) [52, S. 391].

Nach den handlungsorientierten Fragen soll das Interview mit vier demographischen Fragen abgeschlossen werden: Alter, Geschlecht und zwei Fragen zur Ausbildung (siehe Tabelle 13). Nach diesen Information wird in der Regel erst am Ende gefragt, da der Ansatz gilt: Die wichtigsten Themen sollten in der ersten Hälfte des Interviews abgedeckt werden [53, Kapitel 4].

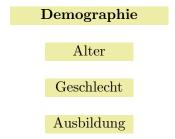
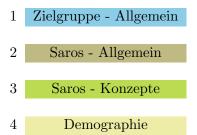


Tabelle 13: Überblick über alle Frage-Kategorien zur Demographie

### 4.1.3 Zusammenfassung

Dieses Unterkapitel 4.1 hat ausführlich dargestellt, wie die Interviews vorbereitet wurden. Im ersten Schritt wurden dafür zunächst die beiden Zielgruppen Arbeit und Freizeit bestimmt. Der zweite Teil beschäftigte sich mit dem inhaltlichen Umfang des Interviews. Hier wurden sowohl Fragen nach der Demographie, als auch für die abstrakte und konkrete Ebene konstruiert. Die einzelnen Interviewteile sind nachfolgend noch einmal zusammengefasst und von allgemein nach konkret geordnet.

Die Darstellung entspricht der in der Einleitung dieses Unterkapitels vorgestellten Tabelle 2.



Zur Erinnerung: 1 - abstrakte Ebene; 2 - beide Ebenen; 3 - konkrete Ebene

Nicht vorgestellt wurden in diesem Unterkapitel die Einstiegsfragen des Interviews und zusätzliche Meta-Informationen. Diese wurden ausgelassen, da sie nicht ausschlaggebend für das Interview und dessen Auswertung, sondern eher als Gedankenstützen zu verstehen sind. Sie umfassen Anweisungen (für den Interviewer) die Versuchsperson über den Ablauf zu instruieren, klarzustellen, dass es sich nicht um einen Test handelt und die Vetraulichkeit der Daten zu versichern.

Der vollständige Interviewleitfaden kann unter Anhang A.1.2 auf Seite 101 eingesehen werden.

# 4.2 Durchführung

Dieser Abschnitt beschäftigt sich mit der Durchführung des Interviews, also mit der eigentlichen Erhebung der mentalen Modelle. Dafür werden die verwendeten Materialien, der Versuchspersonen und die eigentliche Versuchsdurchführung vorgestellt.

## 4.2.1 Material

Das wichtigste Material für die Interviews besteht aus dem Leitfaden mit den ausgearbeiteten Fragen und den vorbereiteten Screenshots. Beides wurde bereits in Kapitel 4.1.2 vorgestellt.

Zudem wurden alle Interviews aufgezeichnet.

Haben diese persönlich stattgefunden, wurde das Gespräch unter Zuhilfenahme des "Olympus VN-8700PC Digital Voice Recorders" aufgenommen. Bei Interviews per Skype kam eine Kombination der freien und kostenlosen Programme Audacitiy, Soundflower und LineIn zum Einsatz. 12

Neben den Anwendern der beiden Zielgruppen wurden auch potentielle Anwender von Saros befragt (mehr dazu in Kapitel 4.2.3). Bei diesen Versuchspersonen kam die, unter 3.3 vorgestellte Methode, SLT zum Einsatz.

Das eingesetzte Material für die Struktur-Lege-Technik umfasst Karteikarten, Post-its und diverse Stifte.

Das vollständige Material mit allen Tonaufnahmen und enstanden SLT-Modellen kann auf der beiliegenden CD gefunden werden.

### 4.2.2 Versuchspersonen

Vier ist die minimale Anzahl von Interviews, die für jede Zielgruppe durchgeführt werden sollte, um Verhaltensmuster erkennen zu können [1, S. 35f.].

Es wurden insgesamt acht vollständige Interviews durchgeführt, davon jeweils drei für jede Zielgruppe und zwei weitere mit potentiellen Anwendern. Die minimale Anzahl konnte demnach nicht eingehalten werden. Eine Erklärung dafür folgt.

<sup>&</sup>lt;sup>12</sup>Für das Betriebssystem Mac OS X 10.5.8 wurden die folgenden Programm-Versionen verwendet: Audacity 2.0.0, Soundflower 1.5.2 und LineIn Version 2.0.3.

Alle drei Versuchspersonen der Zielgruppe Arbeit arbeiten bei einer Firma in Berlin, die Softwarelösungen zur digitalen Archivierung und Verwaltung von Dokumenten und Geschäftsprozessen entwickelt und vertreibt. Allesamt sind im Bereich der Entwicklung tätig. Dabei handelt es sich um die gleichen Versuchspersonen, mit denen [12] im Rahmen seiner Arbeit zwölf Usability-Tests durchgeführt hat. Durch diese Reihe von Tests ist gesichert, dass die Anwender bereits ein sehr gutes Verständnis von Saros haben.

Leider standen zu dem Zeitpunkt der Erhebungsphase nur drei der ursprünglich vier Personen zur Verfügung. Die vierte Person hatte zuvor noch bei den UTs teilgenommen, aber dann aus persönlichen Gründen die Verwendung der erhobenen Daten untersagt. Aus diesem Grund kam sie leider auch nicht für ein entsprechendes Interview in Frage. Andere Mitarbeiter der Firma waren zwar verfügbar, hatten aber keine Vorkenntnisse von Saros.

Die Versuchspersonen der Zielgruppe Arbeit sind alle männlich und im Durchschnitt 24 Jahre alt. Zwei streben einen Bachelorabschluss in Medien-Informatik an und die dritte Person befand sich zum Zeitpunkt der Befragung in der Ausbildung bei der entsprechenden Firma (siehe Abbildung 15).

ID	Zielgruppe	Alter	Geschlecht	Abschluss	angestrebter Abschluss	AVG Alter
01		24	m	Bachelor Informatik	Master Informatik	
02	Freizeit	29	m	Bachelor	Master	25
06		23	m	Bachelor Informatik	Master Informatik	
03		25	m	Abitur	Bachelor Medien-Informatik	
04	Arbeit	25	m	Ausbildung Mediendesign	Bachelor Medien-Informatik	24
05		22	m	Abitur	Ausbildung	
07	potential	22	W	Abitur	Bachelor Wirtschaftsinformatik	27
08	potentiell	32	W	Diplom Raum-/ Umweltplanung	Bachelor Wirtschaftsinformatik	21
	AVG Alter	25				

Abbildung 15: Überblick über die demographischen Daten der Versuchspersonen

Bei den Versuchspersonen der Zielgruppe *Freizeit* handelt es sich ausnahmslos um Studenten, die einen technischen Masterabschluss anstreben. Einer davon an der TU Darmstadt und die beiden Anderen an der FU Berlin. Die Versuchsteilnehmer sind im Schnitt 25 Jahre alt und ebenfalls ausschließlich männlich.

Auch hier wäre es nur durch unverhältnismäßigen zusätzlichen Zeitaufwand möglich gewesen weitere Teilnehmer für diese Zielgruppe zu finden. Bei einem Versuch über die FU-interne Mailingliste Studenten für ein Interview zu gewinnen, kam nicht eine einzige Rückmeldung. Ein Grund dafür könnte sein, dass der Zeitpunkt der Anfrage (6. März 2012) innerhalb der vorlesungsfreien Zeit lag.

Neben den Versuchspersonen der beiden Zielgruppen wurden auch zwei potentielle Anwender befragt. Die Ansichten potentielle Nutzter sind von Interesse, da ihre Anforderungen an die Software verglichen mit den Aussagen über die Software einen Einblick geben, wie der Gebrauch der Software Ansprüchen genügt oder sie verändert[50, Vgl. S. 56].

Die beiden Versuchspersonen ohne Saros-Kenntnis wurden als mögliche Anwender eingestuft, da sie die Arbeitstechnik Paarprogrammierung innerhalb eines halbjährlichen Praxissemester bei einer Firma kennengelernt und angewendet haben. Zum Teil wurde mithilfe von Skype,

Dropbox und SVN auch versucht die Technik der verteilten Paarprogrammierung zu simulieren.

Beide potentiellen Saros-Anwender sind Studentinnen an der HTW Berlin und streben einen Bachelor in Wirtschaftsinformatik ab. Dabei hält eine der beiden bereits einen Diplomabschluss in Raum- und Umweltplanung inne.

Wie anfangs erwähnt, konnte die Mindestanzahl an Interviews je Zielgruppe nicht erreicht werden. Dies hat eine Schmälerung der Reliabilität zur Folge und muss bei der Auswertung der Ergebnisse berücksichtigt werden. So sind die identifizierten Verhaltensweisen mit Vorsicht zu genießen und sollten eher als Verhaltenstendenzen begriffen werden.

## 4.2.3 Versuchsdurchführung

Es wurden insgesamt acht vollständige Einzel-Interviews geführt. Nicht alle liefen exakt gleich ab. Manche unterschieden sich durch den Ort der Durchführung, andere durch das verwendete Medium (Skype) und bei wiederum anderen wurde zusätzlich die SLT eingesetzt (siehe Kapitel 3.3).

Dieses Kapitel stellt für jede Zielgruppe den Ablauf der Versuchsdurchführung vor und schließt mit einigen Interview-Regeln, die bei der Durchführung befolgt wurden, ab.

# Zielgruppe Arbeit

Eine der Grundregeln besagt, dass Interviews an dem Ort stattfinden sollten, an dem die Anwender auch das Produkt nutzen [50, S. 65].

Dementsprechend wurden die drei Entwickler der in Berlin ansässigen Firma an ihrem Arbeitsplatz befragt. Da sich in einem Büro mehrere Arbeitsplätze befanden, wurde das Interview in einem separaten Raum durchgeführt, um so andere Mitarbeiter nicht bei der Arbeit zu stören.

Diese ersten Interviews wurden in einer recht frühen Phase dieser Arbeit durchgeführt. Die Interview-Teile (2) + (3) waren zu diesem Zeitpunkt noch nicht fertig. Aus diesem Grund beschränkte sich der Inhalt zunächst auf die allgemeinen Zielgruppen-Fragen (1) und den demographischen Teil (4). Die Teilnehmer bestätigten im Anschluss aber mit einem Folge-Interview einverstanden zu sein, welches etwa zwei Monate später durchgeführt wurde (siehe Abbildung 16).

Warum diese Interviews nicht erst zu einem späteren Zeitpunkt getätigt wurden, liegt an dem Umstand, dass in diesem Zeitraum UTs mit den Anwender durchgeführt wurden (siehe [12]). So konnte bei der Befragung nach den Saros-Konzepten davon profitiert werden, dass die Einnerungen an die Software noch sehr "frisch" waren und die gegebenen Antworten präziser [1, Vgl. S. 112].

### Zielgruppe Freizeit

Den Anwender dieser Zielgruppe wurde es freigestellt, wo das Interview stattfinden sollte. Weswegen zwei der Interviews per Skype durchgeführt wurden und eines am Institut für Informatik der FU Berlin. Auch hier wurde darauf geachtet, dass diese ungestört durchgeführt

4.2 Durchführung 49

Zielgruppe - Allgemein
 Saros - Allgemein
 Saros - Konzepte
 Demographie

Legende: 1 - abstrakte Ebene; 2 - beide Ebenen; 3 - konkrete Ebene

werden konnten. Dies war vor allem wichtig, da nach allen vier Interview-Teilen innerhalb einer Sitzung gefragt wurden. Eine Befragung dauerte etwa 45 Minuten.

## Potentielle Anwender

Das Interview mit den zwei potentiellen Anwenderinnen fand an deren Arbeitsplatz statt. Beide Studentinnen waren am Datum der Befragung beim gleichen Arbeitgeber tätig. Da auch hier mehrere Personen in einem Büro arbeiteten, fand das Interview ebenfalls in einem gesonderten Raum statt. Der Inhalt begrenzte sich auf den demographischen (4) und den allgemeinen Teil zur Zusammenarbeit (1) und dauerte insgesamt ca. 35 Minuten.

Direkt im Anschluss an das Gespräch wurde die etwa einstündige SLT mit den potentiellen Anwendern durchgeführt. Dafür wurden sie nach Anleitung, der in Abschnitt 3.3 vorgestellten Schritte der SLT, kurz in die Methode eingewiesen. Der inhaltliche Fokus lag auf dem Verständnis und dem Vorgehen bei der verteilten Zusammenarbeit. Beide Versuchspersonen hatten bisher verteilt gearbeitet ohne Saros dafür zu verwenden. Dieser Prozess und eventuelle Wunschfunktionen für eine Software sollten erkundet werden.

					Intervie	w-Teile	<b>=</b>	
#	Datum	ID	Zielgruppe	Z	S-A	s-K	D	Anmerkung
1		04	Arbeit					
2	12.03.2012	05	Arbeit					
3		03	Arbeit					
4	14.05.2012	01	Freizeit					Skype
5		02	Freizeit					Skype
6	24.05.2012	03	Arbeit					
7	24.00.2012	04	Arbeit					
8		05	Arbeit					
9	29.05.2012	06	Freizeit					
10	07.06.2012	07	potentiell					+ SLT
11	07.00.2012	80	potentiell					+ SLT

Abbildung 16: Zeitplan der Interview-Durchführung

## Einige Grundregeln zur Durchführung von Interviews

Abschließend werden an dieser Stelle noch einige Grundregeln für Interviews vorgestellt, die bei der Durchführung zum Einsatz kamen.

No Words Of Your Own Wie bei der Konstruktion der Fragen und Szenarien, so gilt auch bei dem Interview selbst, möglichst kein Saros spezifisches Vokabular zu verwenden. (Eine ausführliche Stellungnahme dazu wurde bereits im Teil "Konkrete Ebene" des Abschnitts 4.1.2 abgegeben.) Dieser Punkt ist bei der Durchführung umso mehr zu beachten, da gerade im allgemeineren Teil der Raum geboten ist, andere Themen, als nur die formulierten Fragen, anzuschneiden.

- Active Listening Beim aktiven Zuhören geht hauptsächlich darum das Interview neutral und ohne Wertung zu führen und den Befragten nicht zu unterbrechen. Zudem werden geäußerte Sachverhalte selbst umformuliert und durch direkte Rückfragen noch einmal bestätigt. Zielt ist es ein Thema aus mehreren Perspeketiven zu beleuchten. [53, Kapitel 4]
- Follow The Conversation Spricht der (potentielle) Anwender ein Thema an oder beantwortet eine Frage, noch bevor diese im Interviewleitfaden erwähnt wird, dann sollte der Antwort nachgegangen werden [1, S. 111]. Die Reihenfolge der Fragen war also nicht fest vorgeschrieben, sondern konnte durch den Befragten jederzeit durchbrochen werden. Aus diesem Grund wurde der Leitfaden als Tabelle formatiert, in der es eine Spalte zum Abhaken des jeweiligen Themas gab. So war leicht ersichtlich, welche Themen schon beantwortet wurden, und welche nicht.
- Encourage Storytelling Um besser zu verstehen, wie (potentielle) Anwender zusammenarbeiten und mit Saros interagieren, sind "Geschichten" und persönliche Erfahrung von großem Wert [50, S. 67]. Aus diesem Grund wurde besonders im Teil "Saros Allgmein" (2) versucht die Versuchspersonen zu ermuntern, mehr von ihren üblichen Abläufen zu erzählen, wie sie darüber denken und was sie davon halten.
- Being A Toddler Gerade wenn es darum geht, die Gründe für bestimmte Verhaltensweisen zu ergründen, kommt man manchmal nicht umhin, wie ein kleines Kind nach dem "Warum" zu fragen. Damit die Fragerei die Anwender im Gegensatz zu Eltern nicht in den Wahnsinn treibt, wurden sie stets anders formuliert und netter verpackt;). [1, Vgl. S. 115]

# 5 Auswertung

Im vorherigen Kapitel 4 wurde vorgestellt, wie die Vorbereitung der Interviews und ihrer Durchführung vorgenommen wurde. Folglich beschäftigt sich dieses Kapitel mit der Auswertung der erhobenen Daten.

Die Auswertung erfolgt entsprechend der beiden untersuchten Ebenen in einem separaten Unterkapitel. Ein zusätzliches Kapitel wird der Frage nach der Notwendigkeit der Zielgruppen-Unterscheidung gewidmet. Die Ergebnisse der einzelnen Abschnitte werden abschließend in Kapitel 5.4 noch einmal zusammengefasst.

### 5.1 Abstrakte Ebene

Dieser Abschnitt des Auswertungskapitels stellt das Vorgehen nach [1] vor und wie dies verwendet wurde, um ein mentales Modell zum Verhalten und Verständnis von Zusammenarbeit zu erstellen. Abschließend wird das Modell ausgewertet, indem die einzelnen Lücken bei Saros identifiziert, erklärt und mögliche Lösungen aufgezeigt werden.

# 5.1.1 Erstellung des mentalen Modells

Das Erstellen des mentalen Modells erfolgt in drei Schritten:

- 1. **Durchsuchen** der Interviews nach verbal geäußertem Verhalten.
- 2. Gruppieren des Verhaltens nach Verhaltensmustern.
- 3. Vergleichen der Verhaltensmuster mit Saros.

Wie diese einzelnen Schritte durchgeführt wurden, soll im Folgenden erläutert werden.

## Durchsuchen nach Verhalten

Als ersten Schritt werden nach [1] Interviews transkribiert. Aufgrund des enormen Zeitaufwandes wurde für diese Arbeit jedoch ein anderer Ansatz gewählt. Anstatt der Transkribierung der acht Interviews, wurden die Tonaufnahmen der Interviews gezielt auf die für diese Arbeit relevanten Beschreibungen von Verhaltensweisen durchgehört, die anschließend ihrem Wortlaut entsprechend erfasst wurden.

Was soll unter "Verhalten" verstanden werden?

Mit Verhalten ist nach [1] mehr als nur ein Mausklick gemeint. Aus diesem Grund soll dafür ein neuer Begriff *Task* eingeführt werden, der dieses umfassendere Konzept bündelt. [1] definiert das Wort Task wie folgt:

"I use 'task' to mean actions, thoughts, feelings, philosophies, and motivations - everything that comes up when a person accomplishes something, sets something in motion, or achieves a certain state." [1, S. 133]

Dementsprechend lassen sich fünf Arten von Tasks unterscheiden. Diese sollen für ein besseres Verständnis im Folgenden jeweils mit einem Beispiel verdeutlicht werden (Vgl. Abbildung 17):

52 5 Auswertung

Task Ein Satz, der eine Handlung oder einen Ausführungschritt beschreibt. Beispiel: Ich führe den Hund jeden Nachmittag auf die Straße.

Implied Task Ein Satz, der eine Handlung impliziert.

Beispiel: Der Hund muss jeden Nachmittag auf die Straße geführt werden.

Third-Party Task Ein Satz, der eine Handlung einer weiteren Person beschreibt.

Beispiel: Meine Mutter führt den Hund jeden Nachmittag auf die Straße.

**Philosophy** Ein Satz, der eine Begründung oder einen Glauben ausdrückt oder warum etwas auf eine bestimmte Art getan wird.

Beispiel: Hunde sollten zwei Mal am Tag auf die Straße geführt werden.

**Feeling** Ein Satz, der die Gefühle eine Person bei einer bestimmten Handlung beschreibt. Beispiel: Den Hund auf die Straße zu führen ist meine Art mich zu entspannen.

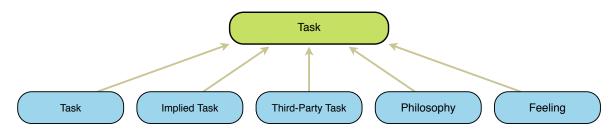


Abbildung 17: Unterscheidug der fünf unterschiedliche Task-Typen

Dabei sind diese fünf Task-Typen von anderen Aussagen zu unterscheiden. Bei diesen Task handelt es sich entweder um Verhaltensweisen, um Gründe für diese Verhaltensweisen oder um Gefühle bei einer solchen Handlung.

Bei Aussagen, die sich zu einer der folgenden Kategorien zuordnen lassen, werden keine Verhaltensweisen ausgedrückt und sind deshalb für die Analyse der Interviews nicht von Belang. Das angestrebte mentale Modell soll eine lange Lebensdauer besitzen und nach [1] ist dies für Verhalten und damit den oben genannte Kategorien der Fall.

Aussagen zu Präferenzen ("Ich mag"), Wünschen ("Ich hoffe"), Erwartungen ("Ich denke"), Medien ("Ich benutze..., um"), Umständen und Beschwerden bringen kein Verhalten zum Ausdruck und werden deshalb bei der Analyse der Interviews vernachlässigt [1, S. 132ff.].

Die Interview-Aufzeichungen wurden also auf die fünf Arten von Tasks hin untersucht. Wie bereits erwähnt, wurde bei einer entsprechenden Task (zuvor Verhalten genannt) dieses als Zitat (5) notiert. Damit dies auch später noch zurückverfolgt werden kann, wurde zudem eine sekundengenau Zeitangabe (6), die Nummer der Versuchsperson (3), die entsprechende Zielgruppe (4) und der Task-Typ (2) selbst in einer Tabelle festgehalten (siehe Tabelle 14).

Damit ähnliche Aussagen leichter zusammengefasst werden können, wurden die Zitate in eine sogenannte Atomic Task überführt. Diese atomare Task bricht das Zitat auf die "Essenz" der Aussage herunter. Dieser Schritt hat zum Ziel die Kernaussagen von verschiedenen Zitaten besser vergleichen und gruppieren zu können. Tabelle 14 veranschaulicht die Bestimmung einer Kernaussage.

Dieser Schritt stellte eine erhebliche Herausforderung dar. Zum Einen war es wichtig nur Tasks festzuhalten und zum Anderen diese auch richtig auf ihre Essenz herunterzubrechen.

5.1 Abstrakte Ebene 53

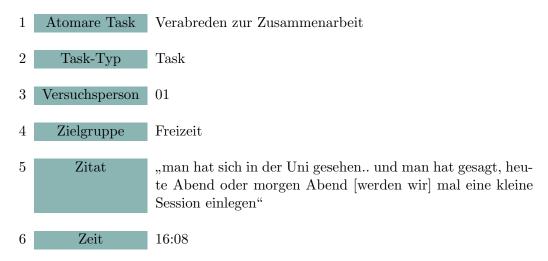


Tabelle 14: Beispiel für die verwendete Notation

Für beides gibt [1] eine ausführliche Beschreibung, einen Test und zeigt viele Positiv- und Negativ- Beispiel auf [1, S. 136ff.].

Manche Aussagen ließen sich trotz der guten Hilfestellung nicht eindeutig identifizieren oder einorden. Aus diesem Grund wurde nach dem Durchsuchen aller Interviews, sich jedes Interview noch Einmal einzeln vorgenommen und alle Aussagen auf die atomaren Tasks und ihren Task-Typ hin überprüft. So sollten zuvor getroffene Entscheidungen mit einem "frischen" Blick begutachtet werden.

### Gruppieren nach Verhaltensmustern

In diesem Schritt gilt es die Atomic Task in einer Bottom-Up-Manier zu immer größeren Gruppen zusammenzufassen. Dieser Vorgang wird in zwei Teilschritten absolviert. Das Ziel ist es die Aussagen der Versuchspersonen zu immer allgemeineren Verhaltensweisen zusammenzufassen. Dadurch ist es am Ende möglich die Funktionalitäten von Saros den Verhaltensmuster der typischen Zusammenarbeit gegenüberzustellen.

Im ersten Teilschritt wurden zunächst alle acht Einzeltabellen zu einer zusammengeführt. Diese Tabelle umfasst insgesamt 200 Einträge. In einem ersten Durchlauf sich ähnelnde Atomic Tasks gruppiert und in Tasks zusammengefasst. Zur Veranschaulichung wurde in Abbildung 18 dieser Vorgang schematisch dargestellt. Abbildung 19 zeigt einen beispielhaften Auschnitt aus der Tabelle.

In der ersten Spalte der Tabelle steht die übergeordnete Gruppe der Atomic Task (Task), dabei erscheint der neue Name für die Task nur bei der Erstnennung, alle weiteren Vorkommnisse sind mit dem Zeichen "^" markiert. Des Weiteren sieht man an diesem Beispiel, dass das Verhalten unabhängig vom Task-Typ (Type of Task) und der Zielgruppe (Aud Seg) zusammengefasst wird. Die ist der Fall, da nur ein mentales Modell für alle Zielgruppen entwickelt wird.

In dem fertigen mentalen Modell können zur Unterscheidung und zur Überprüfung der Zielgruppen diese Unterschiede farbig markiert werden. So lässt sich feststellen, ob sich die Zielgruppen wie eingangs vermutet, in ihren Verhaltensweisen unterscheiden. 54 5 Auswertung

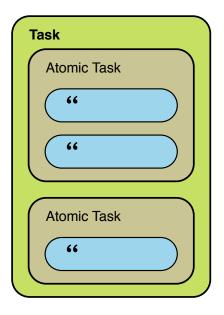


Abbildung 18: Zusammenfassen von Zitaten zu Atomic Tasks und Tasks

Task	Atomic Task	Type of Task	Aud Seg	ID	Quote
Fragen nach Hilfe	Um Hilfe fragen	Implied Task	Freizeit	02	"dass man sich zuarbeitet. Wenn ich an einer bestimmten Datei arbeite, aber noch ein Ergebnis auf eine andere Art ausgewertet haben will, dass ich dann per Chat schreibe, an der Stelle bräuchte ich ein bisschen Hilfe"
٨	Um Hilfe fragen	Task	Arbeit	03	"wenn ich Hilfe brauche, dann sag ich eben bescheid, komm doch mal, guck mal das kommt bestimmt 2-3 die Woche vor durchschnittlich eine Stunde"
^	Fragen nach Hilfe	Task	Arbeit	04	"dann sag ich komm mal rüber, ich hab ein Problem, guck mal auf den Bildschirm"
^	Fragen nach Hilfe	Task	Arbeit	05	"dann frag ich beispielsweise Jerome oder andere Kollegen"
^	Um Hilfe Fragen	Task	Potentiell	08	"und wenn man nicht weiter kam gefragt, wie es weiter gehen sollte"
^	Findet es wichtig immer jemanden fragen zu können	Feeling	Potentiell	08	"was ich wichtig finde, dass man immer jemand fragen kannst, wenn man nicht weiter weißt"

Abbildung 19: Beispiel für das Zusammenfassen von Zitaten zu Atomic Tasks und Tasks

5.1 Abstrakte Ebene 55

Der nächste Teilschritt dieses Verfahrens erfolgt fast analog zum Ersten, nur werden diesmal die Tasks in die nächst größere Gruppe *Tower* zusammengefasst. Dieser Name kommt von der Darstellung des fertigen mentalen Modells, in dem zusammengefasste Tasks übereinander gestapelt als "Türme" dargestellt werden. Die letzt höhere Gruppe über den Towern nennt sich *Mental Space* (siehe Abbildung 20 für eine schematsiche Darstellung).

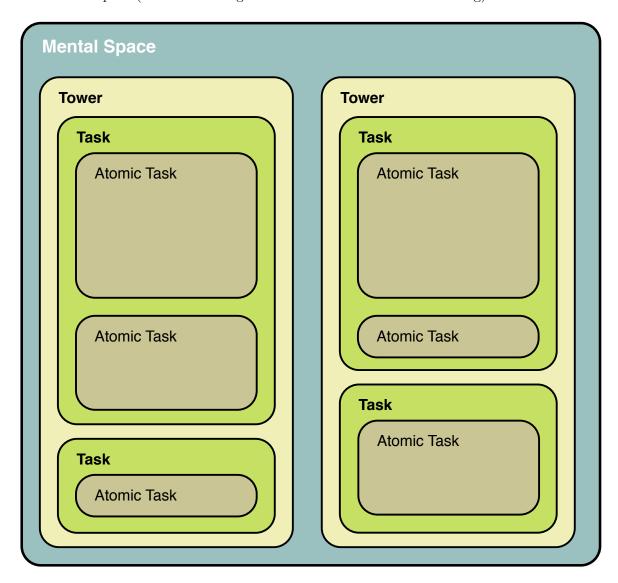


Abbildung 20: Zusammenfassen von Tasks zu Towern und Mental Spaces

Da es bei 200 Tabelleneinträgen schwer ist eine Übersicht zu behalten, wurden im weiteren Verlauf zwei vorgestelle Vorgehensweisen von [1] kombiniert. Beide Vorgehensweisen wurden von der Autorin selbst nicht angewendet, aber in ihrem Buch als Alternativen vorgestellt. Die eine Alternative sieht eine manuelle händische Gruppierung vor. Dies hat den Vorteil, dass es leichter ist Atomic Task, Tasks, Tower und Mental Spaces zu verschieben als in einem elektronischen Dokument. Die zweite Alternative schlägt einige Standardkategorien vor, die aus der Erfahrung der Autoren häufig vorkamen und ebenfalls die Sortierung erleichtern. Beide Alternativen werden im Folgenden beschrieben.

56 5 Auswertung

Für Teilschritt zwei wurde zuerst die vollständige Tabelle mit den 200 Atomic Tasks ausgedruckt und in einzelne Papierstreifen zerschnitten [1, Vgl. S. 179]. War es möglich einzelne Einträge wie in Abbildung 19 schon zu Tasks zusammenzufassen, wurden diese als ein Streifen belassen - und nicht in sechs einzelne zerschnitten. Daraus ergab sich ein "etwas übersichtlicherer" Stapel von gut 130 Papierstreifen, den es noch zu sortieren galt. Um der schier unüberschaubaren Masse an Tasks Herr zu werden, wurden die Tasks und die einzelnen Atomic Tasks den Kategorien "Entscheiden", "Planen", "Vergleichen" oder "Recherchieren" zugeordnet [1, Vgl. S. 188]. Im Laufe dieser Zuordnung wurden daraus präzisere Kategorien der Form "Planen von X", "Planen von Y", "Planen von Z" abgeleitet und formuliert. Passte eine Task in keine dieser Kategorien, wurde eine neue angelegt.

Diese Kategorien bilden die konzeptuellen Gruppen, die in dem mentalen Modell als Tower dargestellt werden. Für Tasks, die bis zum Ende nicht zugeordnet werden konnten, wurde ein Mental Space namens "Unsortiert" eingeführt.

Abbildung 21 zeigt das konkrete Beispiel des Mental Space "Finden von Terminen" mit den entsprechenden fünf Towern aus dem fertigen mentalen Modell.

Der Rahmen um die Tasks gibt an wie häufig diese von unterschiedlichen Versuchspersonen genannt wurden. Eine gestrichelte Linie steht für eine Einfachnennung, eine durchgezogene Linie für 2-3 Nennungen und fettgedruckte Linien stehen für Nennungen  $\geq 4$ .

# **Finden von Terminen**

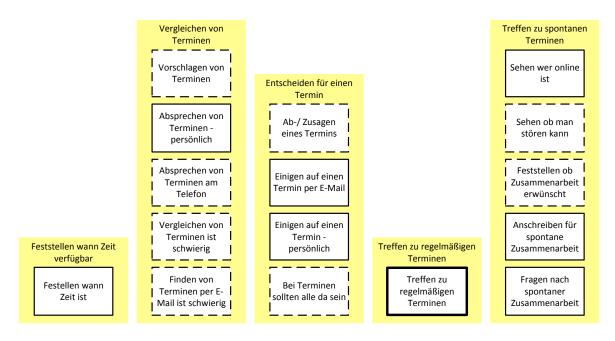


Abbildung 21: Beispiel für das Zusammenfassen von Tasks zu Towern und Mental Spaces - Mental Space "Finden von Terminen"

5.1 Abstrakte Ebene 57

### Automatisiertes Erstellen des mentalen Modells

Der Prozess Atomic Tasks in immer abstraktere Gruppen zusammenzufassen, lässt sich nicht automatisieren [1, Vgl. S. 188], da der Vorgang eines semantischen Verstädnisses der Aussagen der Versuchspersonen bedarf. Was sich jedoch erfolgreich automatisieren ließ, ist die grafische Erzeugung der ersten Hälfte des mentalen Modells. Hierzu wurden alle analog erzeugten Gruppen zurück in eine Excel-Tabelle überführt. Anschließend wurde diese in eine kommaseparierte CSV-Datei exportiert und mittels eines angepassten Python-Scripts in ein XML-Format für Visio überführt.

Abbildung 21 zeigt einen Ausschnit dieser Visio-Datei.

Das Original-Script von Young [1] steht auf der offiziellen Webseite des Verlags Rosenfeld Media frei zur Verfügung.

# Vergleichen mit Saros

Im letzten Teilschritt wird die zweite Hälfte des mentalen Modells erstellt, indem alle in Saros vorhandenen Funktionalitäten und Konzepte den entsprechenden Towern zugeordnet werden. Alle Einheiten, für die es keine Entsprechung gibt, werden in einem separaten leeren Mental Space gesammelt. In diesem findet sich zum Beispiel auch das Saros Host-Konzept wieder. Diese Einheiten existieren in Saros, haben aber in der realen Welt keine Entsprechung.

Hierzu wurde auf die Hilfe des Saros-Entwicklerteams zurückgegriffen. Diese haben die erste Hälfte des mentalen Modells zugeschickt bekommen und alle ihnen bekannten Funktionen zugeordnet. So konnte ein unvollständige Auflistung der Saros-Funktionen und -Konzepte meinerseits ausgeschlossen werden.

Bei der Zuordnung der Funktionen von Saros wurde zwischen *Primary Location* und *Secondary Location* unterschieden. Dabei hat jede Funktion nur eine Primary Location, kann aber mehrere Zweitbesetzungen haben [1, S. 237ff.] (Vgl. Abbildung 22).

Die erste Position (Primary Location) besagt, dass die Funktion genau für diese Task gedacht ist und sich dafür gut verwenden lässt. Diese Position beschreibt das primäre Ziel der Funktionalität. Wohingegen sich die zweite Position an mehreren Stellen befinden kann. Sie besagt nur, dass sich die Funktion dafür verwenden lässt. Zudem enthält Saros auch Funktionen, die sich noch in ihrem Entstehungsprozess oder einem unausgereiften Zustand befinden. Diese sind als "in Progress / experimental" gekennzeichnet.

All diese Einheiten sind in grüner Farbe markiert, um zu signalisieren, dass sie in irgendeiner Form in Saros vorhanden sind.

Nach diesem Schritt werden die ersten Lücken sichtbar: Tower denen keine Saros-Entsprechung gegenübersteht.

Diese Lücken wurden innerhalb des Saros-Entwicklerteams diskutiert und mit einigen Ideen gefüllt. Hierbei wurde sich darüber verständigt, ob es sich bei der Lücke um eine fehlende Funktionalität handelt, die in Saros wünschenswert wäre (rote Markierung) oder die von Anwendern durch andere externe Software abgedeckt muss (blaue Markierung).

Zusätzlich wurde bei folgenden Kollaborations-Werkzeugen nach möglichen Lösungen für die Lücken gesucht. Nach einigen nicht zufriedenstellenden Anläufen, wurde die Suche auch auf

58 5 Auswertung

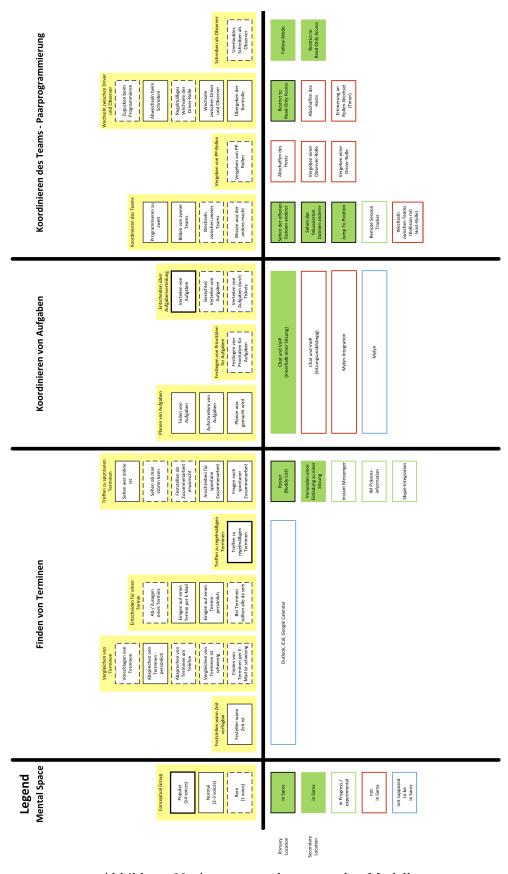


Abbildung 22: Auszug aus dem mentalen Modell

5.1 Abstrakte Ebene 59

andere Arten von Software ausgeweitet. Wurde eine passende Funktion für die Lücke gefunden, so wurde diese mit einem Vermerk auf die Software an entsprechender Position versehen.

- Kollaborationssoftware
  - ACE
  - CollabEdit
  - Gobby
  - GoogleDocs
  - SubEthaEdit
- Sonstige Software
  - Adobe Reader X Pro
  - Mylyn (Eclipse-Plugin)
  - Skype
  - Visio
  - Outlook, iCal, Google Calendar

### 5.1.2 Auswertung der abstrakten Ebene

Im vorherigen Teil wurde die Auswertung - durch die Erläuterung der Lücken im Modell - bereits angerissen. Dieser Teil beschreibt kurz wie die insgesamt 21 identifizierten Lücken schriftlich aufgearbeitet wurden, sodass sie auch auch ohne das Modell zu kennen, verständlich und nachvollziehbar sind. Tabelle 15 zeigt die dafür verwendete Notation.

Nr	Kategorie	Titel	Fatalität	
Sch	Schlüssel			
Prob	Problembeschreibung			
Que	Quelle			
Lösungsansätze				
Disk	Diskussion			

Tabelle 15: Notation der Problembeschreibung

Alle rot markierten Lücken des mentalen Modells wurden nach diesem Schema in eine separate Liste aufgenommen. Diese Liste umfasst insgesamt 21 Probleme, die wie angedeutet mit einer ausführlichen Beschreibung versehen sind. Zudem wurden die zuvor erwähnten möglichen Maßnahmen zur Lösung des Problems ausformuliert (*Lösungsansätze*). Die Zeile *Diskussion* beinhaltet eine Abwägung der unterschiedlichen Lösungsansätze und deren Machbarkeit im Rahmen dieser Arbeit. Für die *Fatalität* wurde die auf Seite 42 eingeführte Skala und deren entsprechende Farbkodierung verwendet.

Abbildung 23 veranschaulicht die verwendete Notation anhand eines Beispiels für die Lücken im Mental Space "Koordinieren von Aufgaben".

60 5 Auswertung

16	Coordination	sitzungsunabhängiges Koordinieren	3
	Tasks	von Aufgaben nicht möglich	

#### Coord-Tasks-1

Saros bietet Anwendern keine Möglichkeit sitzungsunabhängig ihre Aufgaben zu planen, mit Prioritäten zu versehen oder diese jemanden zuzuweisen.

Anmerkung: Alle Kommunikationswege von Saros sind von Sitzungen abhängig, weswegen ein sitzungsunabhängiger Austausch nicht möglich ist.

Quelle: Mentales Modell

#### Lösungsansätze:

- 1. Unterstützen eines sitzungsunabhängigen Chats
- 2. Unterstützen einer sitzungsunabhängigen VoIP-Verbindung
- 3. Integration von Mylyn

Mylyn ist ein Eclipse-Plugin zur Aufgabenverwaltung und setzt alle in der Problembeschreibung genannten Punkte um. Bezüglich der Kommmunikation zur Koordination von Aufgaben: Die meisten Anwender umgehen die Saros-Kommunikationswege mit Skype, um sowohl vor als auch nach einer Saros Sitzung sprechen und schreiben zu können.

Lösungsansatz 1 wurde bereits von S. Rossbach unabhängig vom Saros-Hauptbranch entwickelt (IM-Feature in Saros Light). Zum Zeitpunkt dieser Arbeit beschäftigt sich ein Team im Rahmen des Softwareprojekts "Agile Softwareentwicklung in einem Open-Source-Projekt" damit diese Funktionalität in den Saros-Master-Branch zu überführen. Die Umsetzung der Lösungsansätze 2-3 ist aus zeitlichen Gründen nicht machbar.

Abbildung 23: Beispiel für eine Problembeschreibung

5.1 Abstrakte Ebene 61

### Anmerkung:

Grundsätzlich handelt es sich nicht in jeder Hinsicht um gänzlich neue Informationen, die in dem mentalen Modell abgebildet sind. Jedoch hat es bisher in der Form noch keine Übersicht über diese Informationen gegeben. Die übersichtliche Darstellung hat den Vorteil, für jeden leicht lesbar zu sein und aufgrund ihrer Allgemeingültigkeit eine lange Lebensdauer zu haben. So können Entscheidungen für die zukünftige Entwicklung von Saros zu treffen.

### Ergebnisse der Struktur-Lege-Technik

Die SLT wurde mit den potentiellen Anwender direkt im Anschluss an die Interviews durchgeführt (siehe Abschnitt 4.2.3). Inhalt und Ziel war es, deren Vorstellung von verteilter Zusammenarbeit besser zu verstehen. Das heißt, inhaltlich gehören die entstandenen SLT-Modelle ebenfalls zur abstrakten Ebene und hätten mit in das mentale Modell einfließen können. An dieser Stelle wird kurz erläutert, warum dies nicht geschah.

Für die graphische Darstellung der Modelle sei auf den Anhang A.2.2 auf Seite 111 verwiesen.

Die SLT wurde - trotz mangelnder Vorerfahrung - aus zeitlichen Gründen ohne einen Pilot-Test durchgeführt. Da die Anwender zuvor bereits ein ca. 35 minütiges intensives Interview absolviert hatten, wurde die SLT auf eine Stunde beschränkt.

Aus diesen Gründen war es zum Teil nicht möglich sich genau an den Methodenablauf zu halten. So wurden zum Beispiel vergessen, während der Entstehung der Modelle den Ton aufzuzeichnen, weswegen nur die SLT-Modell als Endprodukt - aber nicht der Entstehungsprozess - vorhanden sind. Diese sind jedoch ohne nähere Erläuterungen wenig aussagekräftig. Zudem handelt es sich hierbei um lediglich zwei SLT-Modelle und nicht 12 wie bei der Studie von [27], weswegen man keine allgemeingültigen und generalisierbaren Aussagen treffen kann.

Nichtsdestotrotz lassen die Modelle einige wenige Aussagen zu.

So kann zum Beispiel der Bedarf einiger Saros-Funktionen untermauert werden. Folgende Funktionen finden in den SLT-Modellen Erwähnung: Die farbliche Annotationen, der zuletzt durchgeführten Änderungen (rechter unterer Quadrant beim SLT-Modell von Vpn 8), die "Malfunktion" des Whiteboards (linker oberer Quadrant) und die SVN- und Skype-Unterstützung (rechter oberer Quadrant).

In Bezug auf das mentale Modell ist vor allem das SLT-Modell von Versuchsperson 7 relevant. Die hier gewählte Anordnung der Karteikarten nach Affinität unterstützt einige der Mental Spaces Kategorien. Der Mental Space "Finden von Terminen" ist ähnlich den Karteikarten in linken oberen Quadranten und der linken untere Quadrant repräsentiert den Mental Space "Koordinieren von Aufgaben".

Zusätzlich kann einem Lösungsansatz für das von [12] entdeckte Usability-Problem 11 Nachdruck verliehen werden. Beide Versuchspersonen äußerten den Wunsch nach einer Kommentar- oder Notizfunktion und [12] hatte schon vorgeschlagen "eine Annotation-Ebene (wie eine Folie) auf jeden synchronisierten Editor zu legen, auf der wie in einem Whiteboard gemalt und geschrieben werden kann" [12, S. 133].

5 Auswertung

## 5.2 Zielgruppe

Dieser Abschnitt beschäftigt sich mit der Frage, ob sich die Differenzierung der Zielgruppen auch in deren Nutzungsverhalten, Motivation und Einstellung widerspiegelt. Zur Beantwortung der Frage wurde der einerseits der erste Interviewteil ausgewertet und andererseits die Tasks des mentalen Modells nach Zielgruppe differenziert.

### 5.2.1 Ergebnisse des Interviews

Bei der Auswertung des Interviewteils zur Zielgruppe wurden kaum unterschiede bei den Zielgruppen festgestellt. Dabei ist anzumerken, dass es sich bei der Erhebung quantitativer Natur handelt, die vermutlich aussagekräftigere Ergebnisse erziehlt hätten.

Nichtsdestotrotz lassen sich die folgenden Tendenzen ablesen:

(Für die vollständige Auswertung sei auf den Anhang A.2.3 auf Seite 112 verwiesen.)

#### • Aktivitäten:

- Es wird 2-3 Mal die Woche für jeweils 2 Stunden zusammengearbeitet.
- Wenn Eclipse verwendet wird, dann nur für Java-Projekte.
- Es wird nur zu zweit programmiert, auch wenn das Team vielleicht aus mehreren Personen besteht.
- Die Arbeit erfolgt an nur einem Projekt zur gleichen Zeit.

### • Motivation:

 Während bei der Zielgruppe Arbeit die Zusammenarbeit hauptsächlich durch das Helfen von Problemen geprägt ist, ist bei den Versuchspersonen der Zielgruppe Freizeit die universitäre Vorgabe die Ursache für das Zusammenarbeiten.

#### • Einstellung:

Grundsätzlich wird der persönliche Kontakt der entfernten Zusammenarbeit vorgezogen.

### 5.2.2 Ergebnisse des mentalen Modells

Zusätzlich das mentale Modell auf unterschiedliche Verhaltenweisen der Zielgruppen untersucht [1, Vgl. S. 218ff.].

Dafür wurden Tasks je nach Zielgruppe eingefärbt, wenn eine Task nur von einer einzigen Zielgruppe beheimatet ist. Sind einer Task mehrere Gruppen zugeordnet, dann wurde die ursprüngliche Farbe weiß beibehalten.

Da es sich bei den Versuchspersonen aus der Zielgruppe Arbeit auch um Studenten handelt, wurde zudem unterschieden, ob sich die getroffene Aussage auch die eigene Zielgruppe Arbeit bezog oder Erfahrung und Abläufe aus dem universitären Rahmen.

Das Ergebnis: Auch hier gibt es sehr viele Überschneidungen der beiden Zielgruppen und der potentiellen Anwender.

5.3 Konkrete Ebene 63

Interessanterweise ist dies weniger bei der Verwendung von Versionswaltungssystems und bei der eigentlichen Programmierarbeit ausgeprägt. In beiden Bereich ist unerklärllicherweise die vorherrschende Zielgruppe Freizeit. Dies gilt auch für den Mental Space "Finden von Terminen". Hier gibt es kaum eine Task, die ausschließlich der Zielgruppe Arbeit zugehörig ist. (Dies ist im eingefärbten mentalen Modell kaum auszumachen, aber auf Tabellen-Ebene deutlich sichtbar). Die Vermutung liegt nahe, dass von den sich stärker überschneidenden Arbeitszeiten bei der Arbeit herrührt. Dadurch fällt es womöglich leichter gemeinsame Termine zu finden.

Diese Erkenntnis ist für Saros aber irrelevant, da der Bereich der Terminfindung ausserhalb des angepeilten Umfangs des Plugins liegt.

### 5.2.3 Auswertung der Zielgruppe

Weder die Fragen innerhalb des ersten Interviewteils noch das mentale Modell lassen darauf schließen, dass die anfänglich vorgenommene Unterscheidung der Benutzergruppen in die Zielgruppe Arbeit und Freizeit sinnvoll war.

Der Vorschlag für zukünftige Untersuchungen wäre diese Differenzierung noch einmal genauer zu beleuchten und die Zielgruppen vielleicht eher von durchgeführten Aufgaben abzuleiten. So könnten Aufgaben wie das Programmieren und Testen vielleicht disjunktere Gebiete darstellen, da diese bei der untersuchten Firma so vorzufinden waren. Wenn man Saros für diese Zielgruppen entwickelt, so wären o.B.d.A. auch die hier untersuchten Zielgruppen Arbeit und Freizeit abgedeckt.

#### 5.3 Konkrete Ebene

Dieser Abschnitt widmet sich der Auswertung der konkreten Ebene.

Dafür wurden die Antworten auf die Fragen nach den Saros-Konzepten analysiert und die verwendete Terminologie der Anwender ausgewertet.

### 5.3.1 Ergebnisse der handlungsorientierten Fragen

Von den Saros verwendeten Konzepte wurden 9 mittels handlungsorientierten Fragen im Interview untersucht. Die Ergebnisse stammen von insgesamt 6 Versuchspersonen. Die anderen beiden Versuchspersonen bekamen diese Fragen nicht gestellt, da es sich um potentiellen Anwender handelt, die über keine Saros-Kenntnisse verfügen.

Die Antworten auf die handlungsorientierten Fragen wurden zunächst für jedes Interview möglichst wortgetreu und schriftlich festgehalten. Anschließend wurden die Aussagen nach hinsichtlich der folgenden drei Aspekte unterschieden: Wurde die Darstellung des Konzepts gesehen (bzw. sich während des Interviews daran erinnert), wurde das Konzept verwendet und wurde es richtig verstanden. Diese Unterscheidung wurde in einer Übersicht zusammengefasst (siehe Anhang A.2.4 auf Seite 115).

Aus diesen Informationen leitet sich ab, ob und inwiefern ein Konzept verbesserungswürdig ist oder nicht.

64 5 Auswertung

Von den 9 untersuchten Konzepten stellten sich 3 als zufriedenstellend heraus (Follow-1/-2 und OpenFile-5), an deren Stelle jedoch drei neue Probleme traten (Follow-3/-4 und OpenFile-6). Diese sollen im Folgenden kurz erläutert werden.

Obwohl der Follow-Modus verwendet und grundsätzlich gut verstanden wird, so herrscht bei einigen Sonderfällen doch eine gewisse Unsicherheit über dessen Status. So ist zum Beispiel unklar, in welchen Fällen er sich im "Pause-Modus" befindet und wann er deaktiviert wird. Das liegt unter anderem daran, dass der Follow-Modus bei Dateien, die nicht über eine Saros-Sitzung geteilt werden, aktiv bleibt, der Anwender aber keine Rückmeldung darüber bekommt (Usability-Problem Follow-3). Ähnliches gilt bei zusammengesetzten Editoren wie zum Beispiel XML-Dateien. Diese werden in speziellen Editoren dargestellt, die die XML-Datei aus mehreren grafisch aufbereiteten Sichten präsentiert und über Reiter auswählbar sind. Die reine XML-Quelltext-Ansicht ist ein Reiter davon. In dieser funktioniert der Follow-Modus wie gewohnt, jedoch nicht innerhalb der anderen Ansichten und beim Wechsel zwischen den einzelnen Reitern (Usability-Problem Follow-4).

Bei dem Usability-Problem *OpenFile-6* handelt es sich eher um ein Darstellungs- als ein Verständnisproblem. In Saros werden der Cursor als auch Markierungen im Quelltext mit der Farbe des Teilnehmers hinterlegt (*Selction Annotation*). Zudem werden auch die zuletzt durchgeführten Änderungen mit einer ähnlichen Teilnehmer-Farbe markiert (*Contribution Annotation*). Es stellte sich heraus, dass der Kontrast der beiden teilnehmerbezogenen Annotationen zu gering und dadurch kaum unterscheidbar ist.

Darüber hinaus vermochte es keiner der Anwender die Funktion der linken Viewport Annotation von der rechten zu unterscheiden, weswegen die zwei Probleme OpenFile-3 und OpenFile-4 fortan als ein Problem OpenFile-3/-4 aufgelistet werden.

Daraus ergeben sich insgesamt 8 Usability-Probleme, die durch die Analyse auf der konkreten Ebene entdeckt worden sind (siehe Tabelle 16).

Schlüssel	Kurzbeschreibung	
Follow-3 Status des Follow-Modus bei externen Dateien		
Follow-4 Status des Follow-Modus bei zusammengesetzten E		
Session-5 Unauffällige Markierung für Projekte der aktuelle		
OpenFile-1	Unauffällige Markierung für fokusierte Datei	
OpenFile-2	Unauffällige Markierung für offene Datei(en)	
OpenFile-3/-4	Bedeutungsunterschied der zwei Viewports unklar	
OpenFile-6	Zu geringer Farbkontrast zwischen Selection und	
Openi ne-o	Contribution Annotation	
Roles-1	Unklarheit über die Bedeutung des Hosts	

Tabelle 16: Zusammenfassung der Usability-Probleme der Konzepte

Abschließend sei noch angemerkt, dass es sich bei dem Host-Konzept ist, um das Problem mit dem höchsten Fatalitätsgrad handelt.

Diese Konzept wurde von den Anwendern nur unzureichend verstanden. Das liegt unter anderem daran, dass Saros dem Anwender zwar signalisiert, wer sich in dieser Sonderrolle befindet, aber nicht erklärt welche Konsequenzen damit einhergehen.

5.3 Konkrete Ebene 65

Dieses Problem wird in Abschnitt 5.3.3 auch noch aus einer anderen Perspektive beleuchtet und abschließend in Form einer kompletten Problembeschreibung vorgestellt (siehe Abbildung 24 auf Seite 67).

#### 5.3.2 Beschwerden und Wünsche

Neben den Antworten auf die handlungsorientierten Fragen wurden auch noch zwei weitere Listen gepflegt: Die der Wünsche und der Beschwerden. Auch wenn es sich dabei um kein Verhalten per se handelt, so sind sie dennoch sehr hilfreich bei der Problemfindung [1, Vgl. S. 136].

Jedes mal, wenn ein Anwender also einen Wunsch für eine neue Funktionalität äußerte oder sich über eine bestehende beschwerte, so wurde dies vermerkt. Für die endgültige Zusammenstellung wurden nur Wünsche und Beschwerden betrachtet, die nicht schon im Sourceforge-Bugtracker eingetragen waren, genau genug beschrieben wurden und noch aktuell waren. Daraus ergab sich ein Datensatz insgesamt 13 Einträgen, bestehend aus 7 Feature-Wünschen und 6 Beschwerden. Davon überschnitten sich 3 mit bereits gelisteten Problemen und je 3 weitere beschrieben den gleichen Sachverhalt und konnten deshalb zusammengefasst werden.

Daraus ergaben sich die in Tabelle 17 aufgelisteten 7 neuen Usability-Probleme (siehe Anhang A.2.5 für die ausführliche Liste).

Schlüssel	Kurzbeschreibung	
Session-6	Beibehalten der eigenen Farbe über mehrere Sitzungen	
OpenFile-7	Unklarheit über Synchronisation von Breakpoints	
Focus-1	Fehlende Synchronisation von Dialog-Fenstern	
Focus-2	Fehlende Synchronisation von Programmen bei Ausführung	
Focus-3	Fehlende Synchronisation der Package Explorer Ansicht	
GettingStarted-1	Anlegen eines XMPP-Kontos unbequem	
GettingStarted-2	langwieriger Einladungsprozess	

Tabelle 17: Zusammenfassung der Usability-Probleme der Wünsche und Beschwerden

### 5.3.3 Ergebnisse der verwendeten Terminologie

Wie im Teil "Konkrete Ebene" des Abschnitts 4.1.2 erstmals erwähnt, wurde bei der Durchführung der Interviews sehr auf das verwendete Vokabular geachtet. Dadurch sollte festgestellt werden, welche Begriffe wirklich von den Anwendern benutzt werden und wo gravierende Unterschiede zur Saros-Terminologie bestehen. Tabelle 18 zeigt eine Übersicht über diese verwendeten Begriffe und wie diese üblicherweise im Saros-Jargon genannt werden.

Die Analyse der Terminologie brachte weniger Erkenntnisse als erhofft.

Dies kann zwei Ursachen haben, entweder reicht der ausschließliche Vergleich von verwendetem Begriff mit dem "realen" Saros-Begriff nicht aus um irgendwelche Rückschlüsse zuzulassen oder es liegen tatsächlich wenig Dissonanzen bei den Anwendern vor.

66 5 Auswertung

Verwendeter Begriff der Versuchsperson		
Host, Leiter, Driver, Server, Erklärer		
Beobacher-Rolle, Client, Benutzer, Follower, Mitstreiter		
Driver		
Observer, Follower		
folgen, Follow-Modus, Follow Mode, Folge-Modus		
Session, Instanz, Sitzung		
Verbindung trennen		
veröffentlichen		
Scrollrand, farbiger Rand, Ränder, Scrollbar, Scrollbalken,		
Balken, Rahmen, farbliche Markierung		
Zeichenfläche, Zeichenbrett, Malprogramm		
VNC, Share Monitor		
XAMP, Jabber		
VoiceChat		

Tabelle 18: Terminologie

Dennoch sind zwei Ungereimtheiten festzustellen:

Zum Einen werden die aus der Paarprogrammierung kommenden Begriffe *Driver* und *Observer* verwendet. Dabei wurden diese bereits 2011 im Rahmen der Abschlussarbeit von [9] abgeschafft und anstelle die Lese- und Schreibrechte eingeführt. Warum diese Begriffe trotzdem sehr präsent im Sprachgebrauch der Anwender vorhanden sind, bleibt unklar.

Zum Anderen fiel auf, dass andere Sitzungsteilnehmer - unabhängig, ob sich diese im Follow-Modus befanden oder nicht - auch als *Follower* bezeichnet wurden. Der Host wurde häufig mit der entsprechenden anderen Position in Verbindung gebracht und auch als "Erklärer" oder "Leiter" benannt. Dies deutet darauf hin, dass die Anwender Host-Konzept nicht von der Follow-Mode-Funktionalität unterscheiden konnten und als zwei voneinander unabhängige Konzepte wahrnehmen. Diese Vermutung wird vor allem durch die folgende Antwort einer Versuchsperson der Zielgruppe Arbeit bestärkt:

"Einerseits war es auch wiederum so, dass der Host jemand Anderem followen kann, also verfolgen. Das.. so ein bisschen im Überlauf ist, dass man nicht so einen konkreten festen Host nur hat, sondern das der Andere auch einen Host spielen kann, so wie ein abgespeckter Host. Da waren die Rollen nicht zu 100% gleich verteilt. Ich glaub, das hatten wir auch gar nicht eingestellt." [Vpn 4]

Dies untermauert das bereits durch die Analyse der handlungsorientierten Fragen identifizierte Verständnisproblem des Host-Konzepts. Das heißt, aus den Analyse der Terminologie ergaben sich keine neuen Usability-Probleme.

### 5.3.4 Auswertung der konkreten Ebene

Insgesamt wurden durch die Analyse auf der konkreten Ebene 15 Usability-Probleme aufgedeckt. Davon wurden 8 durch die handlungsorientierten Fragen nach den Saros-Konzepten

5.3 Konkrete Ebene 67

und 7 durch geäußerte Wünsche und Beschwerden identifiziert.

Diese 15 Probleme wurden wie die der abstrakten Ebene behandelt (Vgl. Abschnitt 5.1.2). Auch sie wurden in eine separate Problemliste aufgenommen und mit einer ausführlichen Beschreibung, einer Quellen-Angabe, möglichen Lösungsansätzen und deren Diskussion versehen. Dafür wurde die Notation aus Tabelle 15 wiederverwendet (siehe Seite 59). Gleiches gilt für die Angabe der Fatalität.

Abbildung 18 zeigt das bereits erwähnte und schwerwiegendste Problem der konkreten Ebene: Das Host-Konzept von Saros.

Die in der Abbildung dargestellte Anmerkung "Saros  $\rightarrow$  MM" bedeutet, dass es sich bei dem Lösungsansatz um eine Anpassung seitens Saros, die sich dem mentalen Modell der Anwender annähert. Zeigt der Pfeil in die andere Richtung, so wird eine Änderung vorgenommen, um das mentale Modell an Saros anzupassen.

13	Comprehension / MM Host	Unklarheit über Bedeutung des Hosts	4
	1.551		

#### Host-1

Anwender wissen zwar, wer sich in der Rolle des Hosts befindet, aber nicht welche Besonderheiten damit einhergehen und welche Auswirkungen dies auf die laufende Sitzung hat. Zudem konnte nicht eindeutig zwischen der Host-Rolle und Funktionen des Follow-Modus differenziert werden.

Anmerkung: Wer die Sitzung initiiert hat und damit die Host-Rolle innehält, ist in der Saros Ansicht zu erkennen: Vor dem Namen des entsprechenden Teilnehmers steht das Wort "Host".

Quelle: Interviews + UT[Kahlert2011]+ UT[Waldmann2012]

### Lösungsansätze:

- Saros → MM
   Aufheben der Host-Sonderrolle, sodass niemand alleinig f
  ür eine Sitzung zuständig ist (Ellis1991).
- Saros → MM
   Ermöglichen der Übertragung der Host-Rolle an andere Sitzungsteilnehmer.
- 3. MM → Saros
  - 1. Erläuterung der Notwendigkeit und damit einhergehender Sonderrechte des Hosts (z.B. visuell in Form eines Comics; Webb2012)
  - 2. Umbenennen des Hosts in "Manager" oder "Coordinator" (S. 46, Ellis1991)

Zeitlich und derzeit grundsätzlich sind 1. und 2. nicht umsetzbar, da die Rolle eines zentralen Sitzungsleiters zu tief in Saros verankert ist. Deswegen sollte mindestens Lösung 3 umgesetzt und die Host-Rolle erläutert werden. Diese Information ist zunächst nur für die Person in der Host-Rolle von Bedeutung, sollte aber grundsätzlich jedem zur Verfügung stehen (z.B. auf der Saros-Webseite).

Arbeitspaket "Getting Started" mit OpenFile-2, OpenFile-3/4 und GettingStarted-1.

68 5 Auswertung

## 5.4 Zusammenfassung und Priorisierung der Usability-Probleme

Durch die Analyse der Interviews wurden 21 Usability-Probleme auf der abstrakten Ebene und 15 über die handlungsorientierten Fragen ermittelt. Zur Veranschaulichung wurde je ein Beispiel im entsprechenden Kapitel angeführt.

Im Folgenden wird ein Überblick über die Ergebnisse dieses Kapitels gegeben. Dafür werden die Usability-Probleme quantitativ aufgearbeitet und abschließend priorisiert, um im nächsten Schritt diese Probleme beheben zu können.

### 5.4.1 Zusammenfassung der Ergebnisse

Von den 36 identifizierten Usability-Probleme überschneiden sich 2 Probleme, die sich beiden Ebenen zuordnen lassen. Dadurch ergeben sich insgesamt 34 disjunkte Usability-Probleme (siehe Abbildung 25). Die vollständige Auflistung dieser kann dem Anhang A.2.6 ab Seite 118 entnommen werden.

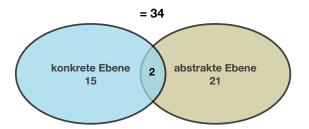


Abbildung 25: Verteilung der Probleme auf die verschiedenen Ebenen

Bei der Sortierung der Probleme nach dem Grad der Fatalität, fällt auf, dass ungefähr 30% der Probleme in die Kategorie *Katastrophal* und *Schwer* fallen (siehe Abbildung 26).

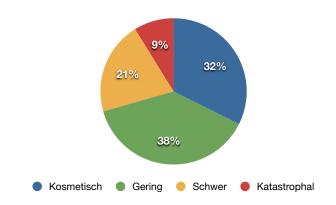


Abbildung 26: Verteilung der Probleme nach Fatalität

Eines der 3 schwerwiegendsten Usability-Probleme wurde bereits vorgestellt: Das Host-Konzept. Dieses Problem wurde durch die Analyse der konkreten Ebene identifiziert. Wo kommen also die anderen Beiden her?

Die beiden übrigen katastrophalen Probleme können der abstrakten Ebene zugeordnet werden

Zur genaueren Untersuchung, werden die 21 Probleme dieser Ebene nach dem 3K-Modell in die Kategorien Kooperation, Koordination und Kommunikation aufgeschlüsselt (siehe 3K-Modell nach Teufel (1995) im Grundlagenkapitel 2.2.1). Abbildung 27 zeigt, dass sich beide katastrophalen Probleme im Bereich Kommunikation ansiedeln.

Das Problem Comm-Verb-1 beschreibt die Einschränkung der verbalen Kommunikationswege auf Saros-Sitzungen. So ist es nicht möglich sich ohne eine bestehende Sitzung zu unterhalten. Zudem ist die VoIP-Funktionalität als experimentell einzustufen.

Comm-2 schildert ein ähnliches Problem für die schriftliche und verbale Kommunikation. Grundsätzlich geht es hier um die mangelnde Unterstützung seitens Saros, um sich gegenseitig zu helfen. Ursache dafür sind die komplizierten Kommunikationswege als auch das Fehlen einer Kommentar- oder Notiz-Funktion.

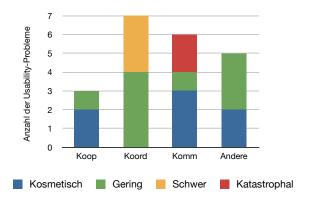


Abbildung 27: Verteilung der 21 Probleme der abstrakten Ebene nach dem 3K-Modell

Darüber hinaus fällt auf, dass die Probleme im Bereich Kooperation einen niedrigen Schweregrad besitzen und zahlenmäßig gering ausfallen. Ein Grund dafür könnten die zwei vorherigen Arbeiten von [9] und [12] sein, die sich ebenfalls mit der Verbesserung der Usability von Saros auseinandergesetzt haben. Beide haben fast den gleichen methodischen Ansatz gewählt und mittels Nutzertests das konkrete Benutzungsverhalten von Saros untersucht. Da es sich Saros um ein Kollaborationswerkzeug handelt, dessen Hauptfunktion die Kooperation ist (siehe Abschnitt 2.2.1), könnte das der Grund für diese Ausprägung sein.

#### 5.4.2 Priorisierung der Usability-Probleme

Das Ziel von Prioritäten ist es möglichst viel in kurzer Zeit zu schaffen. Dafür soll eine Variante des Eisenhower-Prinzips verwendet werden. Bei diesem Prinzip werden Aufgaben nach Wichtigkeit und Dringlichkeit sortiert, wodurch sich vier Quadranten aufspannen lassen. Für die drei wichtigsten ergeben sich die folgenden Handlungsweisungstypen: Aufgaben im Quadrant A sind sofort erledigen, die in Quadrant B zu terminieren und die Quadrant C-Aufgaben zu delegieren (siehe Abbildung 28). [54, Vgl. Kapitel 31]

Nach diesem Verfahren wäre es nun möglich jedes der 34 Probleme einzuordnen. Jedoch unterscheiden sich die Lösungsansätze für jedes Problem so erheblich, dass ein enormer Zeit-

5 Auswertung

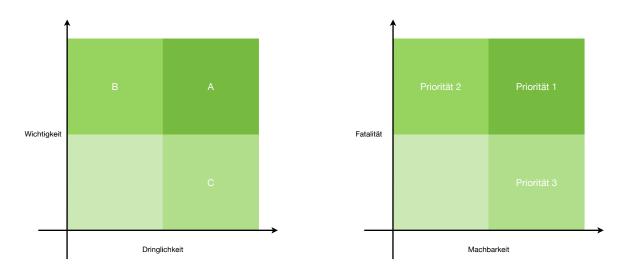


Abbildung 28: Vergleich des eigenen Priorisierungsprinzips mit dem von Eisenhower

aufwand bei der Einarbeitung in das jeweilige Problem hängen bleiben würde. Diese Zeit geht im Umkehrschluss bei der eigentlich Bearbeitung verloren. Dennoch sollten nach [54] gleichartige Aktivitäten gebündelt werden, sodass diese "Umschaltzeit" möglichst gering ausfällt. [54, S. 202]

Aus diesem Grund werden zur effizienten Bearbeitung nicht die Probleme selbst, sondern deren Lösungsansätze betrachtet. Dafür wurden zunächst die insgesamt 63 möglichen Maßnahmen (für die 34 Probleme) nach ähnlichem Bearbeitungskontext zusammengefasst. Dabei schlossen sich die einzelnen Ansätze nicht immer aus, sondern können zum Teil auch ergänzend zueinander betrachtet werden. Die Gruppierung der Lösungsmöglichkeiten ergab insgesamt 26 Arbeitspakete.

Wie eingangs erwähnt wird bei der Priorisierung nicht exakt die gleiche Matrix des Eisenhower-Prinzips verwendet. Die Achse Wichtigkeit bleibt inhaltlich bestehen, wird aber ausschließlich von der Fatalität bestimmt und deshalb auch so benannt. Je schwerwiegender also ein Problem ist, desto wichtiger ist dies anzusehen. Der Punkt Dringlichkeit wird von der *Machbarkeit* abgelöst. Da dieser Arbeit ein begrenzter Zeitrahmen vorgegeben ist, wurde jedes Problem - in Anbetracht der eigenen Fähigkeiten - auf seine Umsetzung hin abgeschätzt (Vgl. Abbildung 29).

Skala der Priorität			
Priorität	Priorität Bedeutung		leutung
1 hoch	hohe Fatalität	+	zeitlich umsetzbar
2	hohe Fatalität	+	zeitlich "nicht" umsetzbar
3	niedrige Fatalität	+	zeitlich umsetzbar
4 niedrig	niedrige Fatalität	+	zeitlich "nicht" umsetzbar

Abbildung 29: Priorisierung nach Fatalität und Machbarkeit

Daraus ergeben sich die im nächsten Kapitel genauer vorgestellten Arbeitspakete.

## Anmerkung:

Da im nächsten Kapitel die relevanten Arbeitspakete vorgestellt und behandelt werden, sei an dieser Stelle für die vollständige Liste auf den Anhang A.2.7 auf Seite 137 verwiesen.

# 6 Behandlung und Behebung der Usability-Probleme

In diesem Kapitel werden die Arbeitspakete vorgestellt, die im vorherigen Kapitel mithilfe der abgewandelten Form des Eisenhower-Prinzips priorisiert wurden. Tabelle 19 zeigt einen Überblick über die Arbeitspakete, die in diesem Kapitel behandelt werden.

Kurztitel	Beschreibung
Getting Started	Gute Einführung in Saros
Follow Mode	Erläuterung des Status im Verfolger-Modus
Saros-View	Überarbeitung der Saros-View einschließlich der Farben
Send File	Senden von Dateien an mehrere Teilnehmer

Tabelle 19: Zusammenfassung der bearbeiteten Arbeitspakete

Im Folgenden werden die einzelnen Arbeitspakete vorgestellt und die entsprechend betroffenen Usability-Probleme erläutert werden. Dabei wird im Allgemeinen das Schema verfolgt zunächst das Problem zu schildern, dann die erarbeiteten Lösungsansätze vorzustellen, diese hinsichtlich der Machbarkeit diskutiert und anschließend die Umsetzung vorgestellt.

Um die Umsetzung der Arbeitspakete "Getting Started" und "Send File" besser nachvollziehen zu können, sind noch einige Informationen bezüglich Saros notwendig.

Saros ist ein Plugin und Plugins werden in Eclipse durch zusätzliche Informationen beschrieben. Ein Teil dieser Informationen befindet sich in der plugin.xml. So werden in der plugin.xml von Saros beschrieben, an welchen Stellen Saros andere Plugins erweitert Extensions und welche Erweiterungspunkte Saros für andere Plugins bereitstellt Extension Points. Dadurch kann Saros die Funktionen von Eclipse erweitern und sogar neue hinzufügen, so zum Beispiel Views, Menüs, Menüeinträge oder Annotationen. [9, Vgl. S. 67]

In den zwei genannten Arbeitspaketen findet eine Anpassung der plugin.xml statt.

## 6.1 Arbeitspaket "Getting Started"

In diesem Arbeitspaket geht es darum, dem Anwendern einen guten Einstieg in eine Saros-Sitzung zu ermöglichen. Die betroffenen die Usability-Proleme sind GettingStarted-1, OpenFile-1, OpenFile-2, OpenFile-3/-4, Session-5 und Roles-1.

Die Behebung der Usability-Probleme wird in drei Teilen vorgenommen.

- 1. Das Host-Konzept
- 2. Die Viewport-Annotation
- 3. Die Saros-Einstiegsseite

Die entsprechenden Probleme werden in dem jeweiligen Teil noch einmal genannt und erklärt.

## 6.1.1 Das Host-Konzept

In diesem Teil des Arbeitspakets geht es darum, das Verständnis der Anwender vom Host-Konzept zu verbessern und so das Usability-Problem *Roles-1* zu beheben.

Das Problem besteht darin, dass Anwender zwar grundsätzlich wissen, wer sich in der Rolle des Hosts befindet, aber nicht welche Besonderheiten damit einhergehen und welche Auswirkungen dies auf die laufende Sitzung hat (siehe Grundlagenkapitel 2.2.3). Zudem wurde innerhalb der durchgeführten Interviews festgestellt, dass die Nutzer nicht eindeutig zwischen der Host-Rolle und der Funktionen namens Follow Mode differenzieren können. So wurde angenommen, dass wenn die Person in der Host-Rolle jemand anderem folgt, damit automatisch die Sonderrolle an denjenigen abtritt. Dies ist jedoch nicht der Fall.

### Lösungsansätze

Zur Verbesserung dieses Umstands wurden drei verschieden Ansätze herausgearbeitet.

Lösungsansatz eins ist der naheliegendste und schlägt eine Abschaffung der Sonderrolle Host vor. Diese Idee ist nicht sonderrlich neu oder originell [14, Vgl. S. 46f.]. [14] beschreibt an einem Groupware-Beispiel ähnlich zu Saros, dass eine ehemals zentrale Architektur in eine verteilte geändert werden musste und welche Hürden damit einhergehen.

Ansatz sieht eine Übertragung der Host-Rolle an andere Sitzungsteilnehmer vor. So wäre es möglich, dass ein Host die Sitzung verlassen kann ohne diese für alle zu beenden.

Variante drei umfasst die minimale Anforderung zur Behebung des Problems und sieht vor dem Anwender in einer geeigneten Form die Notwendigkeit zu erläutern und die damit einhergehenden Sonderrechte zu beschreiben. Zusätzlich wäre eine Umbenennung in zum Beispiel "coordinator" denkbar [14, Vgl. S. 46].

#### Diskussion

Die Lösungen 1 und 2 sind leider im Rahmen dieser Arbeit zeitlich und grundsätzlich nur schwer umsetzbar. Die Rolle eines zentralen Sitzungsleiters ist derzeit zu tief in Saros verankert. Die voraussichtlichen Änderungen sind zum Teil so umfangreich, dass eine eigenständige Master- oder Diplomarbeit denkbar wäre.

Aus diesem Grund wird der dritte Ansatz umgesetzt und die Host-Rolle dem Anwender in einer geeigneten Form erläutert.

#### Umsetzung

Die Erläuterung des Konzepts sollte zur besseren Veranschaulichung nicht rein textuell vorgenommen werden. Deshalb wurde ein Comic als Präsentationsform der Information gewählt. Dabei fiel die Entscheidung auf diese ungewöhnliche Form, weil sich herausgestellt hat, dass sich Comics gut zum Austausch von Informationen eignen und diese wesentlich besser verstanden werden [55]. Grund dafür ist die gleichzeitige visuelle und textuelle Präsentation, die das Verständnis der dargebotenen Information fördert [36, Vgl.]. Dabei sind Comics nicht allzu ungewöhnlich und wurden schon mehrfach zur Erklärung von komplexen Informationen

eingesetzt, allen voran von Andy Field in seiner Buchreihe "Discovering Statistics". Aber auch Google setzte zur Präsentation der Funktionen des Chrome Browsers auf eine grafische Umsetzung<sup>13</sup> und Brad Colbow erläuterte mit einem Comic den Unterschied zwischen XHTML, XHTML 2 und HTML  $5^{14}$ .

Der Nachteil von Comics ist, dass sie wesentlich schwerer zu warten sind. Dieser Aspekt kann jedoch vernachlässigt werden, da sich das grundlegende Host-Konzept mit höchster Wahrscheinlichkeit in naher Zukunft nicht verändern wird.

Für die Umsetzung wurde zunächst mit dem Saros-Team abgestimmt, welche Informationen des Host-Konzepts im Comic widergespiegelt werden sollen. Hierbei einigte man sich innerhalb des Teams auf vier der sechs im Grundlagenkapitel Abschnitt 2.2.3 genannten Punkte. Punkt 2 wurde nicht mit in den Comic aufgenommen, die Information "Die Daten des Hosts stellen Ausgangsbasis für eine Sitzung dar." im Rahmen der Interviews und UTs nicht als Problem erkannt wurde. Punkt 3 ("Der Host verteilt auf technischer Ebene die Nachrichten eines Sitzungsteilnehmers an die übrigen Teilnehmer.") beschreibt eine technische für den Anwender vernachlässigbare Eigenschaft des Konzepts und wurde deshalb ebenfalls nicht mit eingebracht.

Aus den restlichen vier Punkten enstand eines erstes *Storyboard*, welches die darzustellenden Informationen in eine logische Reihenfolge brachte (siehe Abbildung 30).

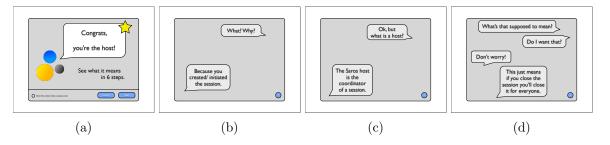


Abbildung 30: Ausschnitt aus dem Storyboard des Host-Comics

Im Anschluss daran wurde das Storyboard unter Verwendung von frei verfügbaren Grafiken<sup>15</sup> in einem Comic umgesetzt (siehe Abbildung 31). Dabei wurde eine männliche Person als Character gewählt, da dies auf Grund des schwindend geringen Frauenanteils in der Informatik statistisch eher der Zielgruppe von Saros entspricht [56, Vgl. S. 10].

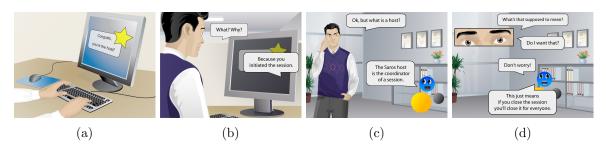


Abbildung 31: Ausschnitt aus dem 7-teiligen Host-Comic

<sup>13</sup>http://www.google.com/googlebooks/chrome/

 $<sup>^{14}</sup>$ http://coding.smashingmagazine.com/2009/07/29/misunderstanding-markup-xhtml-2-comic-strip/

<sup>15</sup>http://www.designcomics.org

Erklärung, die der Nutzer über das Verhalten von Saros erhält, werden als Conceptual Model bezeichnet. Dieses konzeptuelle Modell bildet die Grundlage für ein mentales Modell, welches der Anwender über das Host-Konzept bildet. Bekommt der Anwender kein Conceptual Model präsentiert, bildet er dennoch ein mentales Modell, nur ist dieses mit höherer Wahrscheinlichkeit falsch. [7, S. 48]

Um den Anwender dabei zu unterstützen, das mentale Modell effizient und erfolgreich bilden zu können, ist zu bedenken, zu welchem Zeitpunkt die Information präsentieren wird. [6, S. 272] fand heraus, dass das konzeptuelle Modell dann präsentiert werden sollte, wenn es notwendig ist, Vorhersagen über das System treffen zu müssen. Da das Host-Konzept über den ganzen Lebenszyklus einer Sitzung zum Tragen kommt, ist dies nur bedingt hilfreich. Eines lässt sich daraus aber dennoch ableiten: Die Information sollte zu jedem Zeitpunkt durch den Anwender abrufbar sein. Zudem kommen Anwender zum ersten Mal mit dem Begriff des Hosts in Berührung, wenn sie eine Sitzung selbst starten. Wodurch dieser Zeitpunkt ebenfalls angemessen erscheint.

Im Rahmen der Abschlussarbeit von [12] wurde der vorherige GettingStarted-Wizard durch eine Saros-Webseite abgelöst. Diese öffnet sich nach der Einrichtung und noch vor der ersten Saros-Sitzung automatisch im Eclipse-internen Browser. Darüber hinaus ist die Seite über den Punkt "Getting Started with Saros..." im Saros Menü jederzeit wieder aufrufbar. Dieser Ort erfüllt damit alle zuvor festgestellten Anforderungen und wird für die Präsentation des Comics verwendet werden.

Im letzten Teil des Arbeitspakets wird die Überarbeitung dieser Einführungsseite näher erläutert. Im Zuge dessen wird auch beschrieben, wie sich der Comic in diese Seite eingliedert.

#### Anmerkung:

Während dieser Arbeit wurde die Saros-Webseite auf ein neues System umgestellt. Zum Abgabetermin war diese Umstellung noch nicht vollständig abgeschlossen. Eine Vorabversion der neuen Webseite ist deshalb noch unter einer anderen Adresse erreichbar. Unter dieser ist auch der Comic - unter Verwendung des neuen Designs - einsehbar: www.plat-forms.org/saros/host-comic

#### 6.1.2 Die Viewport-Annotation

Dieser Teil des Arbeitspakets beschäftigt sich mit dem zusammengefassten Usability-Problem OpenFile-3/-4.

In Saros gibt es eine Viewport-Annotationen, die in Form von zwei farbigen Balken links und rechts vom Editor dargestellt wird. Diese spiegeln die Position der anderen Sitzungsteilnehmer im gleichen Dokument wider (Vgl. Grundlagenkapitel Abschnitt 2.2.2).

Das Problem der Annotationen ist die unscheinbare Darstellung der farbigen Balken, die deshalb nur bedingt wahrgenommen werden. Wenn die Viewports von den Anwendern wahrgenommen wurden, dann wurde sie auch verwendet. Nichtsdestotrotz ist deren unterschiedliche Bedeutung verwirrend und selbst die Entwickler des Saros-Teams müssen sich regelmäßig erinnern, welche der beiden Balken welche Information darstellt.

### Lösungsansätze

Der erste der beiden Lösungsansätze sieht vor, die Annotationen auf einen Viewport zu reduzieren. Denkbar wäre eine Lösung ähnlich der von SubEthaEdit, die den Viewport innerhalb des eigenen Scrollbereiches darstellen (siehe Abbildung 32).

Betrachten die Personen den gleichen Bildschirmausschnitt überlagert der Scrollbalken die Annotation. Guckt man hingegen verschiedene Bereiche an, wird der Viewport ober (oder unter) dem eigenen Scrollbalken angezeigt. Dadurch wird deutlich, ob sich die andere Person weiter oben (oder unten) im Dokument befindet.

Lösungsansatz zwei sieht eine Verbesserung der Informationen für die Anwender vor. So soll auf der "Getting Started" Seite ein Hinweis auf die beiden Viewport-Annotation gegeben sein und deren unterschied erläutert werden.

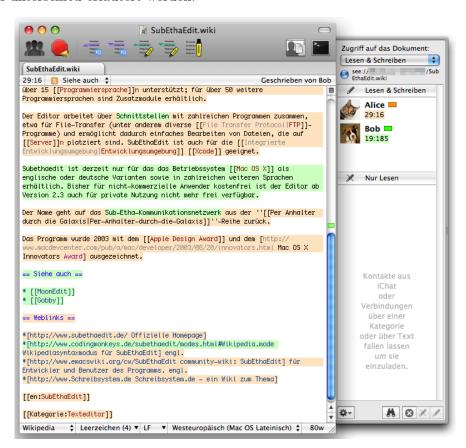


Abbildung 32: Die Viewport-Annotation bei SubEthaEdit rechts innerhalb des Scrollbereiches

### Diskussion

Der gewählte Ansatz ist eine gesunde Mischung aus beiden Lösungen.

Lösungsansatz eins ist von der Idee her gut, jedoch ist die Darstellung innerhalb des Scrollbereichs für Eclipse untypisch. Für eine Umsetzung wäre also zusätzliche Recherchearbeit notwendig, die im Rahmen dieser Arbeit nicht zur Verfügung stand.

Deswegen fiel die Entscheidung einerseits einen der beiden Viewports nicht mehr darzustellen

und andererseits auf der "Getting Started" Seite auf diesen und dessen Bedeutung hinzuweisen. Die Motivation ist also eine übersichtlichere Oberfläche, anstelle einer verwirrenden Informationsflut [28, S. 86].

Nun stellte sich die Frage, welchen der beiden "deaktiviert" werden soll. Die Entscheidungsgrundlage bildeten die Auswertungsergebnisse der Saros-Konzepte (siehe Anhang A.2.4): Die linke Viewport-Annotation wird unwesentlich häufiger wahrgenommen, aber auch seltener benutzt. Richtig verstanden wurden beide gleich häufig. Zusätzlich war zu bedenken, dass der linke Viewport nicht immer eingeblendet wird, der rechte hingegen schon. Letztendlich gab der folgende Dialog zweier Versuchspersonen den letzten Hinweis für die Entscheidung:

```
"der links läuft unter dem anderen"
"kann man auch weglassen links, der rechts reicht doch" (UT 4 - Minute 6:30)
```

Das heißt, der linke Viewport wird ab sofort nicht mehr standardmäßig angezeigt werden.

### Umsetzung

Auf technischer Ebene handelt es sich bei den farbigen Balken um eine ViewportAnnotation, die eine Erweiterung der Klasse SarosAnnotation ist. Zudem besitzt Saros noch zwei weitere Annotationstypen: Contribution und Selection.

Annotationen sind Anmerkungen eines bestimmten Typs, die einem Dokument zeichengenau zugeordnet werden können. Eclipse selbst benutzt einige zur Markierung von Lesezeichen, Suchergebnissen und Fehlern. Saros erweitert also die Eclipse-Annotationen um drei Eigene.

Wo, wie und ob eine Annotation standardmäßig dargestellt werden soll, ist in der plugin.xml festgelegt. Entsprechend ist in dieser Datei definiert, dass die ViewportAnnotation von Saros links (vertical ruler) und rechts (overview ruler) neben dem Editor abgebildet wird. Dabei legt das Attribut verticalRulerPreferenceValue fest, ob die Annotation links dargestellt wird.

Um also die Darstellung der linken Viewports zu unterdrücken, wurde dem Attribut der boolesche Wert false zugewiesen (siehe Listing 1).

```
<specification
annotationType="de.fu_berlin.inf.dpp.annotations.viewport.1"
...

verticalRulerPreferenceKey="de.fu_berlin.inf.dpp.annotations.
    viewport.ruler.1"
verticalRulerPreferenceValue="false"/>
```

Listing 1: Ausschnitt der Definition der Viewport-Annotation in der plugin.xml

Wie bereits erwähnt, handelt es sich bei dieser Änderung um eine Saros-Standardeinstellung, die durch den Anwender jederzeit über das Eclipse-Menü geändert werden kann (siehe *Preferences* > *General* > *Editors* > *Text Editors* > *Annotations*).

#### 6.1.3 Die Webseite

In diesem Teil wird die Überarbeitung der Saros-Einführungsseite beschrieben. Die dadurch adressierten übrigen Usability-Probleme sind GettingStarted-1, OpenFile-1, OpenFile-2 und Session-5.

Bei den letzteren dreien geht es um die eingeschränkte Wahrnehmung der Awareness-Mechanismen von Saros. Im Falle von *OpenFile-1* handelt es sich dabei, um die fokusierte Datei der anderen Teilnehmer, die im Package Explorer durch eine grünen Punkt markiert ist. Ähnlich ist es bei den entsprechend gelben Annotationen, die anzeigen, welche Dateien der andere Teilnehmer offen hat (*OpenFile-2*). *Session-6* umfasst die Awareness-Information der geteilten Projekte, die mit einem blauen Doppelpfeil im Project Explorer gekennzeichnet sind.

Das Problem mit dem Bezeichner GettingStarted-1 fällt etwas raus, da es hierbei um das anzulegende XMPP-Konto geht, welches für die Benutzung von Saros notwendig ist. Hierbei soll stärker darauf hingewiesen werden, dass es auch möglich ist Konten anderer E-Mail-Anbieter, wie Google oder GMX zu verwenden.

Wie [12] erwähnte bedarf die Einführungsseite einer Überarbeitung. Jedoch ist dies - wie prognostiziert - nicht durch die Bachelorarbeit von [57] abgedeckt. Die Arbeit von [57] sah die Umstellung der Seite auf ein neues System und eine rein strukturelle Verbesserung vor. Die inhaltliche Überarbeitung der Einführungsseite wird im Rahmen dieser Arbeit durchgeführt. Aus diesem Grund wird auch die zuvor verwendete Gliederung an dieser Stelle durchbrochen und keine weiteren Lösungsansätze vorgestellt. Was aber auch daran liegt, dass für Probleme wie OpenFile-2 keine Alternativen gefunden wurden. Andere Werkzeuge stellen Informationen dieser Art noch nich einmal dar.

#### Diskussion

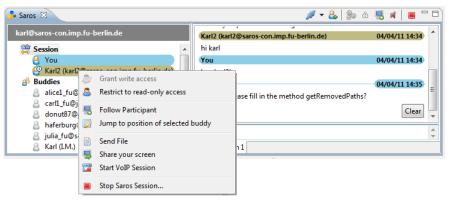
Da es keine Lösungsansätze abzuwägen gibt, sei an dieser Stelle erwähnt, warum die Einführungsseite dringend überarbeitet werden muss.

Probleme wie *OpenFile-1* und *Session-6* wurden durch [12] schon erheblich abgemildert und liegen mittlerweile bei einer weit niedrigeren Falatität als zuvor. Jedoch sind die Informationen auf der Seite nie aktualisiert worden. So werden veraltete Icons beschrieben (siehe Abbildung 33) und Menüs gezeigt, die in der Form nicht mehr existieren. Dafür werden Begriffe wie Driver und Observer verwendet, die ebenfalls so nicht mehr in Saros vorhanden sind.

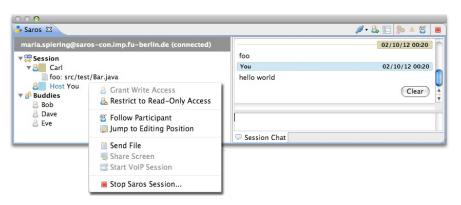
Kurzum die Einführungsseite muss auf den neuesten Stand gebracht werden.

### Umsetzung

Die Umsetzung wurde auf der Plattform der neuen Saros-Webseite erstellt. Die Einführungsseite wurde dabei dahingehend optimiert, dass Erst-Anwender möglichst schnell zum Ziel kommen, also ihre erste Saros-Sitzung starten können. Zuvor wurden hier viele - zum Teil sogar alle - Wege beschrieben. So wurde durch einen großen Screenshot alle drei Saros-Menü-Wege zum Starten einer Sitzung gezeigt. Grundsätzlich ist es wichtig, dass es diese Alternativen gibt, jedoch müssen sie für eine erste Anwendung nicht alle beschrieben werden. Deshalb wurde nach dem Simplicity-Ansatz von [28] auf alle Mehrfachlösungen verzichtet und direkteste



(a) Alter View und Kontextmenü



(b) Neuer View und Kontextmenü

Abbildung 33: Veränderungen des Saros-Views, der Icons in der Toolbar und des Kontext-Menüs seit der letzten Aktualisierung der Einführungsseite

für die Darstellung auf der Seite gewählt [28, Vgl. S. 86]. Des Weiteren wurden die wie in Abbildung 33a gezeigten, veralteten Bildschirmfotos und Beschreibungen aktualisiert. Zudem wurde der Host-Comic, wie in Abbildung 34 dargestellt, in Form einer kleinen Vorschau integriert.

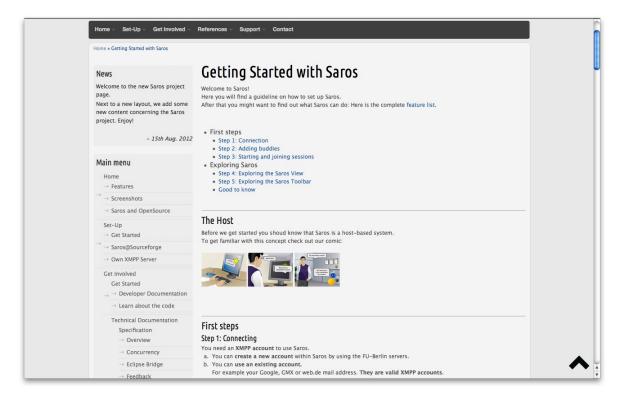


Abbildung 34: Die neue Einführungsseite

#### Anmerkung:

Die überarbeitete Einführungsseite ist erreichbar unter www.plat-forms.org/saros/eclipsegetstarted.

### 6.2 Arbeitspaket "Follow Mode"

Dieses Arbeitspaket umfasst die Usability-Probleme Follow-3 und Follow-4, bei denen es um den Status des Follow-Modus geht. (Beide Probleme wurden im Rahmen der Auswertung in Abschnitt 5.3.1 vorgestellt.)

In Saros ist es möglich einem anderen Sitzungsteilnehmer zu folgen. Startet man diesen Verfolger-Modus, dann wird der Sichtbereich des verfolgten automatisch an den *Follower* übertragen. Öffnet der verfolgte Teilnehmer eine Datei, dann wird diese auch beim Verfolger geöffnet (mehr dazu im Grundlagenkapitel 2.2.2).

Bei Dateien, die nicht im Rahmen einer Sitzung mit den anderen Teilnehmern geteilt sind, und zusammengesetzten Editoren kommt es jedoch zu einer ungenauen Darstellung über den Status dieses Verfolger-Modus.

Wird eine nicht geteilte Datei geöffnet, dann erhält der Verfolger keine Notiz davon, er sieht immer noch die zuletzt fokusierte Datei und ist in dem Glauben, dass es sich dabei auch um die sichtbare Datei des verfolgten handelt.

Ähnlich verhält es sich bei zusammengesetzten Editoren. Der Verfolger kann nicht sehen, wenn zwischen Reitern innerhalb dieses Editors gewechselt wird. Zudem wird die Datei selbst nur automatisch geöffnet, wenn der Verfolgte den Reiter mit der Ansicht des reinen Quelltexts anzeigt.

### Lösungsansätze

Für das geschilderte Problem der externen Dateien (Follow-3) wurde nur eine Lösung erarbeitet: Die Erweiterung der angezeigten Status-Information "(following)" innerhalb des Saros-Views.

Für die etwas komplexere Situation zusammengesetzten Editoren (Follow-4) gibt es mehrere Möglichkeiten.

Ein Lösungsansatz ist die Anzeige eines Dummy-Editors, der auf die fehlende Synchronisationsmöglichkeit hinweist, da sich die bisherige Benachrichtigung in Form einer Sprechblase als zu subtil herausgestellt hat [12, S. 158, Usability-Problem 86].

Im Rahmen des Lösungsansatz zwei, würde der Anwender über Pause-Status des Verfolger-Modus informiert und somit auf eine Einschränkung dessen hingewiesen.

Die optimalste Lösung wäre jedoch die vollständige Unterstützung der zusammengesetzten Editoren. Dafür wäre es notwendig die Aktivität "Wechseln des Reiters" mit Saros nachzustellen. Dazu müssten neue *Activities* geschrieben werden, die dann vom Verfolgten versendet und auf der Gegenstelle verarbeiten werden müssten.

### Diskussion

Die Lösungsansätze 1 und 3 des Follow-4-Problems sind für den zeitlichen Rahmen der Arbeit nicht umsetztbar. Aus diesem Grund wurde sich entschieden, die Information über den Status des Verfolger-Modus zu erweitern. Das heißt, wird eine externe Datei oder ein zusammengesetzter Editor geöffnet, soll darauf hingewiesen werden, dass der Verfolger-Modus noch aktiv, aber pausiert ist.

Zudem existiert in dem Saros-View für jeden Sitzungsteilnehmer bereits ein Element, welches den Pfad der aktuell fokusierten Datei einblendet. Handelt es sich dabei um eine externe Datei, dann werden keine Information angezeigt (siehe Abbildung 35a).

Das heißt, diese Darstellungsmöglichkeit blieb bisher ungenutzt und eignet sich bestens, um dem Anwender mitzuteilen, dass eine Datei geöffnet wurde, die nicht Teil der Sitzung ist.

#### Umsetzung

Für die Umsetzung wurden hauptsächlich drei Klassen umgeschrieben.

Der AwarenessInformationCollector kümmert sich dabei um die Anzeige-Logik und die beiden Klassen UserElement und FollowModeInformationTreeElement um die Darstellung



(a) Alte Darstellung

(b) Neue Darstellung

Abbildung 35: Vergleich der dargestellten Status-Informationen des Follow-Modus

selbst. Die eigentliche Arbeit übernimmt also die Collector-Klasse, die alle Awareness-Informationen bündelt. In ihr wird bestimmt, ob der aktive also fokusierte Editor Teil der Sitzung ist oder nicht. Die Art der Darstellung wird von den anderen beiden Klassen übernommen.

Öffnet der Verfolgte eine sitzungsexterne Datei, so wird der Verfolger mittels der Information "non-shared file open" darüber in Kenntnis gesetzt. Besteht die Sitzung aus weiteren Teilnehmern, wird ihnen diese Information ebenfalls angezeigt.

Des Weiteren werden Sitzungsteilnehmer bei externen Dateien und zusammengesetzten Editoren darüber informiert, dass der Verfolger-Modus nur pausiert, aber sehr wohl noch aktiv ist (siehe Abbildung 35b).

## 6.3 Arbeitspaket "Saros-View"

Dieses Arbeitspaket umfasst neben dem Usability-Problem OpenFile-6 auch die Umgestaltung des Saros-Views.

Bei dem Problem *OpenFile-6* handelt es sich um Darstellungsproblem von verschiedenen Farben im Editor. In Saros werden zuletzt geänderten Zeichen einer Datei mit der Farbe des entsprechenden Teilnehmers hinterlegt. In einer etwas helleren Farbabstufung der gleichen Farbe werden die Position des Cursors und Markierungen kenntlich gemacht. Im Rahmen der Interviews kam die Beschwerde auf, dass diese Markierungen innerhalb der zuletzt geschriebenen Zeichen nur schwer zu erkennen ist, da sich die beiden Farbtöne zu ähnlich sind (siehe Abbildung 37a).

Zudem wurden Teile der Umgestaltung des Saros-Views durch [12] noch einmal in Frage gestellt. So wurden unter anderem eingefärbte Kacheln, die die Farbe des jeweiligen Sitzungsteilnehmers anzeigen, neu eingeführt. Wird im Saros-View ein Teilnehmer markiert, verschwindet jedoch diese Farbindikation (siehe Abbildung 36). [12, S. 93f.]

Diese Umstellung wurde damit begründet, dass a) die vorherige Darstellung unsauber sei, weil sie nur unvollständige abgerundete Ecken und Kanten verwendet (siehe Abbildung 39b auf Seite 85) und b) eine subtilere Darstellung den Versuchspersonen in der Evaluation genügt hätte. Zudem verstößt laut [12] der unter a) beschriebene Umstand gegen die Heuristik 8 "Ästhetisches und minimalistisches Design".

Für eine Erläuterungen der Heuristiken und deren Einsatz bei Saros sei auf die Arbeiten von [9] und [12] verwiesen.

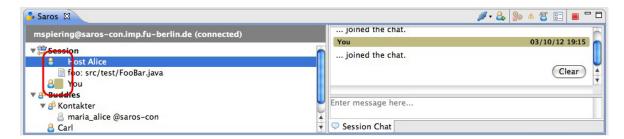


Abbildung 36: Fehlende Farbkachel bei Selektierung eines Teilnehmers

#### Lösungsansätze

Zur Verbesserung des Saros-Views sollen die eingeführten Farbkacheln durch eine farbliche Hinterlegung des Teilnehmernamens über die volle Breite ersetzt werden [9, Vgl. Arbeitspaket 4.3]. Dabei gilt es die zu beachten, dass die durch [12] kritisierte Darstellung überarbeitet wird und bei Markierung eines Teilnehmers, dessen Farbe weiterhin sichtbar bleibt.

Der erste Lösungsansatz für *OpenFile-6* sieht vor die Standardeinstellung der zwei verwendeten Farbtöne für den Cursor bzw. Markierung und der Hintergrundfarbe des zuletzt Geschriebenen deutlicher voneinander abzuheben.

Des Weiteren wäre vorstellbar, dass der Anwender die Ausprägung des Farbunterschieds selbst bestimmen kann. Eine Möglichkeit zur Skalierung der Farbe mittels eines Schiebereglers wurde bei SubEthaEdit entdeckt (Ansatz 2).

Ebenso wäre als drittes eine deutlichere Hervorhebung des Cursors der anderen Sitzungsteilnehmer denkbar. So wird bei GoogleDocs der Cursor eines Teilnehmers immer mit dessen Namen und Farbe angezeigt, sobald dieser etwas schreibt oder den Cursor umsetzt.

#### Diskussion

Für *OpenFile-6* wäre eine Kombination aller drei Lösungsansätze durchaus wünschenswert, ist aber aus zeitlichen Gründen nicht umsetzbar.

Grundsätzlich sollte aber auch eine Standardeinstellung der beiden Farbtöne so gut zu unterscheiden sein, dass eine Skalierung nicht notwendig ist. Dabei kann die Sehstärke/-schwäche eines jeden sehr wohl unterschiedlich ausgeprägt sein, weswegen eine Selbstbestimmung des Anwender über den Farbunterschied durchaus legitim wäre. [9] hatte aber bereits die fünf Grundfarben auf solche Sehschwächen hin optimiert, weswegen eine Skalierung der Farbe durch den Anwender eher als Spielerei angesehen werden kann.

Bei der Umgestaltung des Saros-Views spricht einiges für die farbliche Hinterlegung der Teilnehmernamen gegenüber der Farbkachel. Zumal sowohl in den Interviews als auch in den UTs von [12] zum Vorschein kam, dass sich Anwender sehr an den Farben orientieren ("der eine kriegt die eine Farbe der andere kriegt die andere Farbe und dann behalten diese beiden Leute auch immer diese Farbe. Dann ist es einfacher sich zu orientieren..." [Vpn03]). Dass die Farbe bei der Selektion des Sitzungsteilnehmers dann nicht mehr angezeigt wird, ist also eher suboptimal.

### Umsetzung

Bei der Bearbeitung des Usability-Problems ist zudem aufgefallen, dass die Berechnung der unterschiedlich dargestellten Farben fehlerhaft war. Die Berechnung erfolgte fälschlicherweise innerhalb des RGB-Farbraums, wodurch einige Farben nicht dunkler (oder heller) wurden, sondern in einer gänzlichen Anderen als der Ursprungsfarbe resultierten. Deswegen wird die Farbberechnung nun nach dem HSL-Modell vorgenommen. HSL steht hierbei für Hue, Saturation und Lightness und sieht bereits eine Komponente zum Aufhellen von Farben vor (Vgl. Abbildung 37).

```
Bar.java
                                                                                            J A.java
            J Start.java
                        J Bar.java
                                                           J A.java
                                                                     Start.java
      package test:
                                                               package test:
                                                         H
H
       public class FooBar {
                                                               public class FooBar {
                                                            10
               (a) Alte Darstellung
                                                                        (b) Neue Darstellung
```

Abbildung 37: Verbesserung des Farbkontrasts

Die Umstellung des Saros-Views konnte nicht vollständig abgeschlossen werden. So existiert der fertig umgstaltete Saros-View für Mac OS X, jedoch unterscheidet sich der View in seiner Darstellung zu Windows-Betriebssystemen so sehr, dass die Änderung zunächst nicht übernommen wurde. Eine entscheidene Rolle spielt hierbei, dass eine erhebliche größere Anzahl der Saros-Nutzer das Windows-Betriebssystem verwenden und der Anteil der Mac-Nutzer dagegen eher gering ausfällt (Vgl. Abbildung 38).

Der für Mac OS X erstellte Prototyp liegt in dem Review-Board Gerrit von Saros vor.

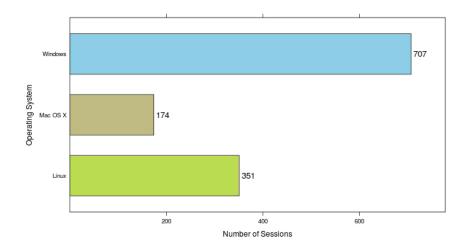


Abbildung 38: Anzahl von Saros-Sitzungen gruppiert nach den verwendeten Betriebssystemen (Statistik vom 26.03.2012)

Im Folgenden wird kurz erläutert, welche Änderungen übernommen wurden und bei welchen die Darstellung der Unterschied zwischen den Betriebssystemen so erheblich war, dass diese nicht in Saros aufgenommen werden konnten.

Der Saros-View besteht aus zwei Widgets, dem linken BuddySessionDisplayComposite und dem rechten ChatRoomComposite.

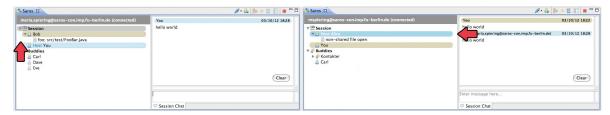
Änderungen wurden nur für den Chat übernommen. Aus ästhetischen Gründen werden mittels der neuen Farbberechnung nun die Farben dezent helleren Tönen dargestellt.

Änderungen im linken Teil liegen dem erwähnten BuddySessionDisplayComposite zugrunde. In diesem werden die einzelnen Elemente in einer Baumstruktur organisiert und dargestellt. Wie diese einzelnen Elemente gezeichnet werden, kann selbst bestimmt werden. Dazu muss der Baum auf bestimmte Ereignisse hören. So wird das Ereignis SWT.EraseItem dafür verwendet, um den Hintergrund und/ oder die Selektionsmarkierung zu verändern. Dies wurde in diesem Fall genutzt und entsprechend angepasst.

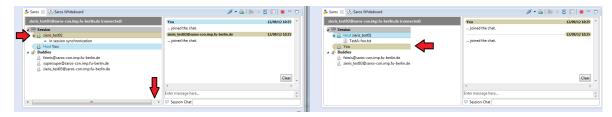
An dieser Stelle wird das verschiedenartige Verhalten der Betriebssysteme vermutet und so scheint es, dass die Erzeugung des SWT.EraseItem Ereignisses sich zwischen den Betriebssystemen unterscheidet. Dadurch kommt es zu den, in Abbildung 39 dargelegten, zwei unterschiedlichen Darstellungen.

#### Anmerkung:

Die Beschreibung fällt an dieser Stelle etwas detaillierter aus, um nachfolgenden Entwicklern, die sich dieser Problematik annehmen, die Fehlersuche zu erleichtern.



(a) Mac OS X



(b) Windows 7

Abbildung 39: Darstellung des Saros-Views bei den verschiedenen Betriebssystemen

## 6.4 Arbeitspaket "Send File"

In diesem Arbeitspaket wird das Problem Comm-3 behandelt.

Projekte in der Entwicklungsumgebung Eclipse können über Saros mit Anderen geteilt werden. Jedoch nicht alle projektbezogenen Dateien befinden sich in Eclipse. So können aufgemalte Ideen, Aufgaben- oder Zeitpläne unabhängig von den eigentlichen Programmdateien und Eclipse gespeichert sein. Damit diese Dateien zwischen Teilnehmern ausgetauscht werden können, ohne sie in Eclipse importieren zu müssen, gibt es die Funktion "Send File". Mit dieser Funktion ist es jedoch nur möglich Dateien an einzelne Sitzungsteilnehmern zu senden. Möchte man eine Dateie an alle Teilnehmer verschicken, so muss dies für jede Person einzeln getan werden.

#### Lösungsansätze

Eine Möglichkeit eine Datei an alle Sitzungsteilnehmer zu schicken, wäre ein zusätzlicher Eintrag im Kontextmenü des Saros-Views und / oder im Saros-Menü selbst.

Eine weitere Option, die die erste Lösung nicht ausschließt, entspräche einer Umsetzung wie in Skype. Hier ist es möglich Dateien per Drag & Drop in das Chat-Fenster an eine oder mehrere Personen gleichzeitig zu versenden. Der Saros-View stellt einen Chat bereit, der sich für diese Umsetzung eignen würde.

#### Diskussion

Da sich die beiden Lösungsmöglichkeiten nicht ausschließen, wäre eine Umsetzung beider Ansätze denkbar. An dieser Stelle sei vorweggenommen, dass beide Ansätze aus zeitlichen Gründen nicht mehr fertig gestellt werden konnten.

Ansatz eins konnte jedoch durch ein ausführliches *Refactoring* vorbereitet werden. Zudem liegt der prototypische Funktionalität im Review-Board von Saros. Im Folgenden wird ein Teil der Umsetzung des ersten Ansatzes vorgestellt.

## Umsetzung

In Saros ist die "Send File"-Funktion über einen Eintrag im Kontextmenü des Saros-Views aufrufbar. Die Kontextmenü-Einträge sind über sogenannte Actions realisiert. Eine *Action* setzt eine Funktion um, indem sie ein Stück Programmcode ausführt und über einen Menüeintrag, ein grafisches Symbol in einer Toolbar oder einer Tastenkombination angesprochen werden kann.

In Saros sind viele Actions als interne Klassen definiert. Dies hat den Nachteil, dass "die internen Klassen eine hohe Kopplung mit ihrer Elternklasse aufweisen und damit nicht wiederverwendbar sind" [9, S. 69]. Dies gilt auch für die SendFileAction-Klasse.

Aufgrund der hohen Kopplung wurden Actions mit der Eclipse-Version 3.3 von Commands abgelöst. Commands lösen das Problem der Kopplung, indem sie ihre Deklaration von der Anzeige und der Implementierung trennen. Ein Command verkörpert damit nur noch die abstrakte Idee einer Ausführung von Programmcode. Die eigentliche Ausführung wird durch sogenannte CommandHandler übernommen. Je nach Systemstatus können unterschiedliche Handler - aber maximal einer je Status - aktiviert werden. [9, Vgl. S. 70]

Dadurch wäre nur noch ein globaler SendFile-Command notwendig, welcher in verschiedenen Menüs eingebunden werden kann und sich unterschiedlich verhält, je nachdem welcher Handler angesprochen wird.

In einem ersten Schritt war dafür eine umfangreiche Änderung der SendFileAction notwendig. Die erwähnten internen Klassen wurden in eigentständige Dateien extrahiert, damit diese nicht mehr an die Action gekoppelt sind. Diese Änderung ist eine Form des Refactorings und wurde erfolgreich vorgenommen.

Die folgenden Schritte seien hier nur noch hypthetisch aufgeführt, da sie nicht komplett fertiggestellt werden konnten.

Die nächsten Schritte sehen die Bekanntmachung des neuen Commands über die Erweiterung des Extension Points org.eclipse.ui.commands in der plugin.xml vor. In dieser muss der neue Command wie folgt definiert werden [16, Vgl.].

```
<extension point="org.eclipse.ui.commands">
    <command
    categoryId="org.eclipse.ui.category.file"
    id="de.fu_berlin.inf.dpp.ui.commands.SharedProjectSendFile"
    name="Send File...">
    </command>
    </extension>
```

Listing 2: Deklaration des SendFile-Commands

Anschließend kann der Command einem CommandHandler zugeordnet werden. In diesem Fall soll dieser nur aktiv sein, wenn man angemeldet ist (isConnected) und sich in einer laufenden Saros-Sitzung befindet (isSarosSessionRunning).

```
<extension point="org.eclipse.ui.handlers">
  <handler
      class="de.fu_berlin.inf.dpp.ui.commandHandlers.
         SharedProjectSendFileHandler"
      commandId="de.fu_berlin.inf.dpp.ui.commands.
         SharedProjectSendFile">
    <activeWhen>
      <and>
        <reference
          definitionId="de.fu_berlin.inf.dpp.ui.definitions.
        </reference>
        <reference
          definitionId="de.fu_berlin.inf.dpp.ui.definitions.
             isSarosSessionRunning">
        </reference>
      </and>
    </activeWhen>
  </handler>
</extension>
```

Listing 3: Deklaration des SendFile-CommandHandlers

Durch diese Änderungen wäre es nun möglich eine Datei an alle Sitzungsteilnehmer gleichzeitig zu senden. In einem weiteren Schritt wäre es dann nur noch notwendig dem Chat einen DropListener hinzuzufügen, der auf Dateien reagiert und diese mithilfe der durch das Refactoring überarbeiteten Jobs versendet.

## 6.5 Andere Arbeitspakete

Aus zeitlichen Gründen konnten im Rahmen dieser Arbeit nur die zuvor genannten Arbeitspakete umgesetzt werden.

Jedoch wurde Anfang September an der FU-Berlin ein sechswöchiges Softwareprojekt (kurz SWP) durchgeführt, bei dem ein fünfköpfiges SWP-Team an Saros mitenwickelte. Diesem Team wurden sowohl bestehende Probleme aus dem Sourceforge-Bugtracker und als auch die vier schwerwiegendsten noch offenen Probleme dieser Arbeit vorgestellt.

Dem SWP-Team wurde freigestellt, an welchen Aufgaben sie arbeiten wollten. Zwei der bearbeiteten Probleme sind die Arbeitspakete "Chat" und "Color", die in dieser Arbeit definiert wurden und deshalb im Folgenden kurz vorgestellt werden.

#### 6.5.1 Chat

Die Chat-Funktionalität war zuvor an die Bedingung einer laufenden Sitzung geknüpft. Im Zuge dieses Arbeitspakets wurde der Chat von dieser Vorbedingung losgelöst, so dass es nun auch möglich ist Nicht-Sitzungsteilnehmern zu schreiben.

Dadurch wurden zwei schwere und ein katastrophales Usability-Problem behoben (*Coord-Task-1*, *Coord-Team-1* und *Comm-2*).

### 6.5.2 Color

Dieses Arbeitspaket adressiert das Usability-Problem Session-6. In der bisherigen Saros-Version war es Anwendern nicht möglich sich eine Farbe über mehrere Sitzungen zuzuordnen. Die Farbzuweisung erfolgte durch den Host, der die Farben seiner letzten Sitzung auf die nächste überträgt. Dabei spielt es keine Rolle, wer die Teilnehmer dieser Sitzung sind und welche Farben sie zuvor hatten.

Das SWP-Team hat das Farbmanagement wie folgt überarbeitet.

Anwender können ihre Farbe nicht mehr frei wählen, sondern haben die Wahl zwischen fünf Standardfarben. Diese fünf Farben wurden unter farbpsychologischen, ästhetischen und unter Aspekten der Farbfehlsichtikeit von [9] überarbeitet. Die fünf Standardfarben sind für Saros ausreichend, da derzeit maximal fünf Personen an einer Sitzung teilnehmen können.

Zudem können Anwender eine dieser fünf Farben als ihre Lieblingsfarbe deklarieren (siehe Abbildung 40). Diese Farbpräferenz wird auf jede Sitzung unabhängig vom Host angewendet. Mögliche Farbkonflikte werden über einen Algorithmus seitens des Host gelöst.

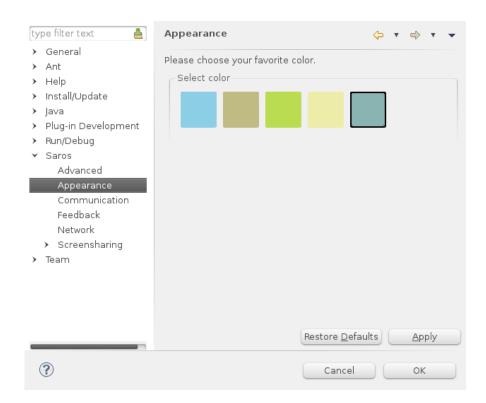


Abbildung 40: Auswahl der Saros-Lieblingsfarbe in den Einstellungen von Eclipe

90 7 Evaluation

## 7 Evaluation

Nachdem im vorherigen Kapitel die Arbeitspakete und deren Bearbeitung vorgestellt wurden, setzt sich dieses Kapitel mit der Evaluation dieser vermeintlichen Verbesserungen auseinander. Dafür werden die einzelnen Arbeitspakete erneut betrachtet und überprüft, ob die Änderungen zu einer Verbesserung der Gebrauchstauglichkeit beitragen.

Eine empirische Evaluation im Sinne des User-Centered Design konnte aus zeitlichen Gründen nur für den enstanden Comic des Arbeitspakets "Getting Started" durchgeführt werden. Die anderen Arbeitspakete konnten zum Teil nicht vollständig fertiggestellt werden, weswegen auch deren Evaluation aussteht. Deren vermeintlichen Verbesserungen sollen dennoch kurz diskutiert werden.

## Die Arbeitspakete "Saros-View" und "Send File"

Die Arbeitspakete "Saros-View" und "Send File" konnten aus genannten Gründen nicht vollständig abgeschlossen werden. Im Rahmen des Arbeitspakets "Saros-View" konnte jedoch das Usability-Problem *OpenFile-6* behandelt werden, weswegen eine Diskussion dieses Arbeitspakets folgt.

#### Saros-View

Zur Behebung des Uability-Problems *OpenFile-6* wurde der Kontrast zwischen den Farben für Markierungen und für die letzte Änderung verbessert. Der Kontrast selbt wurde mit einem Werkzeug namens *Contrast Analyser* gemessen, dass sowohl den Farb- als auch den Helligkeitsunterschied zwischen zwei Farben berechnet. Das Programm der Firma the Paciello Group basiert auf dem *Contrast Ratio Algorithm*. Der Algorithmus ist eine Empfehlung des *World Wide Web Consortium (W3C)* (December 2008), um zu bestimmen "whether or not the contrast between two colours can be read by people with colour blindness or other visual impairments."<sup>16</sup> Das Programm ist kostenlos und frei erhältlich.

Mit dem Programm konnte bestätigt werden, dass sowohl der Farb- als auch der Helligkeitsunterschied verbessert wurde. Zudem wird die Helligkeit Farbe nun nach dem korrekten HSL-Modell berechnet.

Die Änderung wurde von einem weiteren Entwickler des Saros-Teams begutachet und als geeignet angesehen. Dies ersetzt jedoch keine Evaluation durch den Anwender, so bleibt offen, ob diese Verbesserung als ausreichend angesehen werden kann. Dabei sollte angemerkt werden, dass eine Beurteilung der Änderung von dem verwendeten Monitor abhängt (z.B. matt oder glänzend) und so eine Evaluation erschwert.

Die Änderung der Farbindikation von den ursprünglichen Kacheln zu einer Hinterlegung des Teilnehmernames über die gesamte Breite in dem Saros-View konnte nicht fertiggestellt werden. Abschließend sei trotzdem kurz erläutert, warumg dieser Änderung wünschenswert ist und worauf sie sich begründet.

<sup>&</sup>lt;sup>16</sup>http://www.paciellogroup.com/resources/contrast-analyser.html

Die in dem Saros-View verwendete Baumstruktur zur Darstellung der Sitzungsteilnehmer und deren Awareness-Informationen nimmt viel Platz ein. Dies wurde mehrfach von Entwicklern des Saros-Teams bemängelt, die die Akzeptanztests vor einem neuen Release durchführen. Andererseits werden hier viele nützliche Informationen abgebildet.

Um Objekte sinngemäß zu gruppieren wird im Bereich des *User Interface Designs* häufig auf die *Gestaltgesetze* zurückgegriffen, die einige Regeln unserer Warhnehmung beschreiben. Das Gesetz der "Nähe" besagt, dass Objekte die zeitlich oder örtlich nah aneinander liegen, als zusammenhängend wahrgenommen werden. [58, Vgl. S. 114]

Aufgrund des schon vorherrschenden Platzmangels kommt dieses Gesetz jedoch nicht zum Einsatz. Die Einführung der neuen Darstellung könnte diesem Problem abhilfe schaffen, da diese die einzelnen Teilnehmer-Elemente und deren Awareness-Informationen optisch voneinander trennt.

Bei ein Umsetzung für alle Betriebssystem wäre zudem der Verstoß gegen die Heuristik "Ästhetisches und minimalistisches Design" aufgehoben und damit die ursprüngliche Änderung durch [12] hinfällig.

## Das Arbeitspaket "Follow Mode"

Für dieses Arbeitspaket und die damit adressierten Usability-Probleme konnte aus Zeitmangel keine Evaluation mit Anwendern durchgeführt werden. Aus diesem Grund folgt lediglich eine kurze Diskussion der umgesetzten Lösung und warum diese den Anwendern hilft ein korrektes mentales Modell aufzubauen.

Das Ausgangsproblem war die Unsicherheit über den Status des Verfolgermodus, die sich in Aussagen, wie der folgenden widerspiegeln.

"Ich glaub, beide konnten scrollen, aber sobald man scrollt, ist der Follow Mode wieder weggewesen. [...] Sobald man irgendwas ändert, verlässt man den Follow Mode." [Vpn 4]

Grundsätzlich kamen Anwender aber gut mit dem Verfolgermodus zurecht. Probleme bereiteten nur die Sonderfälle des Modus (Wechseln zu nicht geteilten oder nicht unterstützten Datei-Formaten). Das heißt, Anwender nehmen die Funktionalität und deren Anzeige über den Saros-View wahr. Zudem wird der Follow-Modus häufig verwendet und wurde - soweit durch die unzureichende Rückmeldung möglich - richtig verstanden.

Woher kommt also das Problem?

Auf konzeptueller Ebene existieren für den Verfolgermodus drei Zustände: An, Aus und Pause. Die Saros-Oberfläche (System Image) bildet jedoch nur zwei der Zustände ab (An und Aus). Diese unvollständige Darstellung wurde behoben, sodass Saros nun eine korrekte grafische Rückmeldung über den Status des Verfolgermodus indiziert.

Ob die Information tatsächlich genügend Hilfestellung bietet und beim Erlernen des mentalen Modells "Follow-Modus" dient, bleibt aufgrund der fehlenden Evaluation ungeklärt.

92 7 Evaluation

## Das Arbeitspaket "Getting Started"

Das Arbeitspaket "Getting Started" besteht aus drei Teilen: Dem Host-Comic, der Viewport-Annotation und der Einstiegsseite. Wobei es für die Überarbeitung der Viewport-Annotation und der Webseite keiner Evaluation bedarf.

Im Zuge der Änderung der Viewport-Annotation wurde auf die Darstellung des linken Viewports verzichtet. Dadurch werden weniger Informationen über die anderen Sitzungsteilnehmer dargestellt. Im Gegensatz zum Verfolger-Modus entsteht dadurch aber kein falsches Abbild des konzeptuellen Modells. Auch wenn weniger Awareness-Informationen übertragen werden, zieht dies zwangsweise keine Verschlechterung der Usability nach sich. Zuvor wirkten die Anwender aufgrund der zwei Viewports verunsichert und verwirrt. Mit nur einem Viewport entsteht dieses Problem nicht. Das heißt, trotz der geringeren Anzahl an Informationen, wird die Gebrauchstauglichkeit verbessert.

Ähnliches gilt für die Überarbeitung der Saros-Einstiegsseite. Inhaltlich wurde - bis auf die Integration des Host-Comics - nichts verändert. Die vorhandenen Informationen wurden aktualisiert und dahingehend optimiert, dass eine schnelle Erst-Benutzung möglich ist. So werden Erst-anwender nicht überfordert und kommen schneller zu einer ersten Sitzung, da sie sich mit weniger Informationen auseinandersetzen müssen.

Somit steht Evaluation des Host-Comics noch aus.

#### **Evaluation des Host-Comics**

Für die Evaluation des Host-Comic wurde ein Online-Fragebogen mit sieben Fragen erstellt. Dabei decken fünf Multiple-Choice-Fragen die vier im Comic präsentierten Aspekte des Host-Konzepts ab und zwei weitere Fragen dienen dazu, um zu ermitteln wie intensiv Saros bisher verwendet wurde.

Im Rahmen des Fragebogens wurden zwei Situationen skizziert und dann entsprechende Fragen zu dieser Situation gestellt.

Anhand eines Beispiel soll hier das Schema der Fragen veranschaulicht werden.

### Situation (1)

Vorbedingung:

- Alice, Bob und Carl wollen zusammenarbeiten.
- Alle haben das Java-Projekt, an dem sie gemeinsam arbeiten wollen, lokal auf ihrem Rechner.
  - 1. Frage: Bob wird von Alice zu einer Sitzung eingeladen. Bob lädt Carl in die Sitzung mit ein. Wer ist der Host?
    - (a) Alice
    - (b) Bob
    - (c) Carl
    - (d) Alice und Bob

Diese Frage wurde gestellt um den Aspekt (1) "Als Host wird der Benutzer bezeichnet, der die Sitzung eröffnet hat." des Host-Konzepts zu erfragen.

Die übrigen Fragen werden ohne die Antwortmöglichkeiten vorgestellt, folgen aber dem gleichen Schema. Der vollständige Fragebogen befindet sich im Anhang B.1.3 auf Seite 148.

- 2. Frage: Alice und Bob wollen beide, dass Carl über die möglichen Änderungen am Java-Projekt bestimmt. Wer sollte am Besten die Sitzung starten?
  - (4) Nur der Host kann Lese- und Schreibrechte vergeben.
- 3. Frage: Was passiert, wenn Alice die Sitzung verlässt?

  (6) Nur der Host kann als ganzes die Sitzung beenden.
- 4. Frage: Was passiert, wenn Carl die Sitzung verlässt?

  (6) Nur der Host kann als ganzes die Sitzung beenden.
- 5. Frage: Alice benutzt den Follow-Modus um Carl zu folgen. Wer ist der Host? (5) Die Rolle des Hosts ist nicht übertragbar. Des Weiteren adressiert die Frage ein, in den Interviews identifiziertes, Verständnisproblem beim Follow-Modus. Anwender waren der Ansicht, dass die Rolle über den Verfolgermodus übertragbar sei.

An der Umfrage nahmen insgesamt 106 Personen teil. Durch die zwei Fragen nach den bisherigen Erfahrungen mit Saros konnte ermittelt werden, dass 13 der Teilnehmer Saros-Entwickler und 21 Anwender sind. Die übrigen 72 Personen fallen in die Kategorie "potentielle Anwender", da sie keine Saros-Kentnisse besitzen.

Grundsätzlich überwiegen die Richtignennung jeder Frage bei weitem die falschen Antworten. Die Anzahl der richtigen Antworten liegt immer über 69% und stellt somit ein gutes Ergebnis dar (Vgl. Abbildung 41). Die Unsicherheit über Rolle des Hosts in Kombination mit dem Verfolgermodus wurde in den Interviews festgestellt und ist auch hier deutlich wiederzuerkennen. Frage fünf zielte auf diesen Sachverhalt ab und bestätigt das Problem mit 31% falschen Antworten. Das heißt für den Comic, dass der nicht vorhandene Zusammenhang noch deutlicher herausgearbeitet und erklärt werden muss. Die bestätigt auch der folgende Kommentar eines Teilnehmers:

"Die Frage nach dem Host, wenn Carl Follower von Alice ist, sieht für mich nicht eindeutig aus."

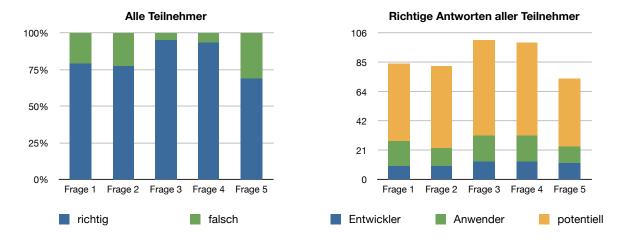


Abbildung 41: Die Antworten aller Teilnehmer zusammengefasst

94 7 Evaluation

Für eine differenziertere Auswertung veranschaulicht Abbildung 42 die Antworten kategorisiert nach den Teilnehmern. Hier lässt sich ablesen, dass die Saros-Entwickler wie erwartet am besten abschneiden.

Vor allem fällt hier positiv auf, dass die Ergebnisse der "potentiellen Anwender" besser als die der Saros-Anwender ausfallen. Dies ist insofern positiv zu vermerken, als dass sich die Antworten der potentiellen Anwender allein auf den Comic stützen. Dies legt die Vermutung nahe, dass dieser genügend Informationen beinhaltet, um die Fragen besser zu beantworten als mit praktischen Erfahrungen zu Saros. Dies ist besonders bei Frage 2 aufällig. Hier liegt der Unterschied der Richtignennungen bei ungefähr 20% und bei Frage 5 bei fast 10%.

Auch wenn der Comic diese beiden Aspekte des Host-Konzepts noch deutlicher zum Ausdruck bringen könnte, so ist trotzdem ein positiver Trend zu erkennen.

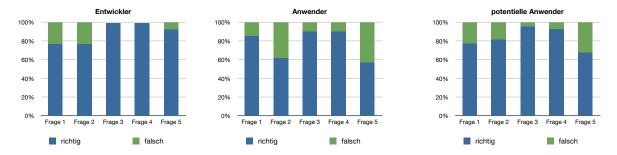


Abbildung 42: Die Antworten der Teilnehmer kategorisiert nach deren Saros-Kentnissen

Die Präsentation der Information in Form eines Comics stieß bei den Teilnehmern auf gemischte Gefühle. So gab es einerseits skeptische Rückmeldungen der Art:

"ich versteh den Sinn des Comics nicht.. mit 2 Sätzen hätte man dieselben Infos rüberbringen können"

Ein anderer Teilnehmer äußerte sich hingegen positiv:

"Wirklich cooles Comic! Davon sollte es noch viel mehr geben!"

Auch wenn die Differenzierung zwischen dem Verfolgermodus und dem Host noch deutlicher herausgearbeitet werden könnte, so konnte der Comic dazu beitragen, dass trotz fehlender Saros-Kenntnisse die Fragen gut bis sehr gut beantwortet wurden. Aus diesem Grund kann der Host-Comic als Usability-Verbesserung angesehen werden.

# 8 Fazit

Ziel dieser Arbeit war, unter Verwendung eines User-Centered Design (UCD) Ansatzes die Usability von Saros zu verbessern. Dafür wurde das Verständnis der Benutzer von Zusammenarbeit im Allgemeinen und den Saros-Konzepten im Speziellen untersucht. Im Zuge der Untersuchung wurde der Begriff des mentalen Modells eingeführt und dessen Zusammenhang mit Usability und dem UCD erläutert.

Für die Erhebung der mentalen Modelle der Anwender wurden verschiedene Methoden untersucht. Das Interview stellte sich für den Untersuchungszweck als die geeigneteste Methode heraus. Die Grundlage für die Interviews bildeten szenariobasierte, handlungsorientierte Fragen nach Bravo-Lillo et al. (2011) [48] und Rode et al. (2004) [47] um das Verständnis der Saros-Konzepte auf der konkreten Ebene zur erfassen. Für die Untersuchung des Verständnisses der Zusammenarbeit auf allgemeiner Ebene wurde an die Idee von Young (2008) [1] angeknüpft. Es Fragen konzipiert, die den Versuchspersonen den Freiraum gaben, eigene Themen anzusprechen.

Im Hinblick auf die Durchführung der Interviews wurden die Zielgruppen Arbeit und Freizeit bestimmt. Es wurden insgesamt 8 Befragungen durchgeführt, wovon jeweils 3 mit Personen einer Zielgruppe stattfanden und 2 weitere Versuchspersonen zu den potentiellen Anwendern zählen. Durch die Analyse der Interviews enstand ein mentales Modell, anhand dessen und der Antworten auf die handlungsorientierten Fragen konnten 34 Usability-Probleme identifiziert werden.

Anschließend wurde für jedes dieser Probleme mehrere Lösungsansätze erarbeitet. Zur Bearbeitung der Ansätze wurden diese in Arbeitspakete zusammengeführt und priorisiert. Die Arbeitspakete mit dem besten Verhältnis zwischen Machbarkeit und Fatalität wurden anschließend behandelt. Durch die 4 behandelten Arbeitspakete wurden 10 Probleme behandelt und 9 der insgesamt 34 Usability-Probleme gelöst.

Am Ende der einfachen UCD-Iteration wurde für den Comic ein Evaluation in Form eines Fragebogens durchgeführt. Hier konnte gezeigt werden, dass Teilnehmer ohne Saros-Kenntnisse und nur auf Grundlage des Comics in der Lage waren, die Fragen zu richtig beantworten. Für die anderen behandelte Probleme steht eine Evaluation noch aus. Jedoch gibt es guten Grund zur Annahme, dass diese ebenfalls eine positive Auswirkung auf die Usability von Saros haben.

Zusammenfassend lässt sich der Schluss ziehen, dass durch die Erhebung des mentalen Modells im Rahmen des UCD, viele Probleme festgestellt werden konnten und durch die Behebung einiger davon, die Usability von Saros verbessert wurde.

# Ausblick

Neben der Behebung der Usability-Probleme bestand der Hauptteil dieser Arbeit in der Analyse. Ein Ergebnis dieser ist die Liste mit den ursprünglichen 34 Usability-Problemen und deren Lösungsansätzen. Diese Liste besteht, trotz des Softwareprojekts "Agile Softwareentwicklung in einem Open-Source-Projekt" und deren Behebung von 4 zusätzlichen Problemen, aus 21 Usability-Problemen und ist damit noch sehr lang. Diese Probleme bleiben deshalb

96 8 Fazit

zunächst noch offen. Auf drei der - meiner Ansicht nach - wichtigsten Probleme, wird im Folgenden eingegangen.

Die Zusammenfassung der Probleme und den dazugehörigen Lösungsansätzen befindet sich im Anhang A.2.6 Seite 118.

## Das mentale Modell

Das mentale Modell ist ein weiteres Ergebnis und Artefakt der Analyse. Durch die Gegenüberstellung des Verhaltens der Anwender mit den Funktionen von Saros, werden viele gute Möglichkeiten zur Weiterentwicklung von Saros aufgezeigt. Gut sind diese vor allem deshalb, weil das Modell eine lange Lebensdauer besitzt. Die verwendeten Werkzeuge mögen sich vielleicht ändern, aber das, in dem Modell widergespiegelte, Verhalten bleibt bestehen und ändert sich wenn, dann nur langsam [1, Vgl. S. 24].

Bei der Auswertung des mentalen Modells in Kapitel 5 wurde bereits erwähnt, dass die Bereiche Kommunikation und Koordination noch verbessert werden können. Eines Kommunikationsprobleme wurde durch das Team während des Softwareprojekts behoben. Zuvor war es nur möglich mit anderen Teilnehmern zu chatten, wenn man sich in einer gemeinsamen Saros-Sitzung befand. Durch ihre Verbesserung ist es nun möglich unabhängig von einer Sitzung mit Anderen zu schreiben. Das analoge Problem wäre zum Beispiel noch für die VoIP-Sprachkommunikation zu beheben.

#### Die Awareness-Informationen

Ein Teil der Verbesserung des Saros-Views wurde in dieser Arbeit bereits vorgenommen. Da bei der Darstellung des Views betriebssystemabhängige Probleme auftauchten und die Entwickler sich regelmäßig über die Platzverschwendung durch die dargestellten Awareness-Informationen beschwerten, sehe ich hier die nächst größere Baustelle.

Bisher nimmt die Darstellung des Verfolgermodus seitens des Verfolgten durch ein zusätzliches Kindslement in der Baumstruktur viel Platz ein. Abhilfe könnte der folgende Lösungsansatz schaffen: So wäre es möglich anstelle des zusätzlichen Elements, den Verfolger selbst als Kindselement des Verfolgten darzustellen. Dies würde den Verfolgermodus sinngemäß veranschaulichen, da der Verfolger als Kindselement an dem Verfolgten "dran hängt" und eingerückt dargestellt wird. Zudem würde man mit dieser Lösung den gewünschten Platz sparen.

## Das Host-Konzept

Abschließend sei hier das Host-Konzept und der umgesetzte Comic noch einmal aufgegriffen.

Auch wenn die Abschaffung der Sonderrolle des Hosts wünschenswert wäre, da sie auch technischen Gegebenheiten erwachsen ist und nicht aus den Anforderungen der Anwender, so ist dies aus genannten Gründen nicht abzusehen (siehe Kapitel 6.1). Aus diesem Grund sollte versucht werden, den Anwender so gut wie möglich dabei zu unterstützen ein korrektes mentales Modell aufzubauen. Der Comic ist eine Hilfestellung und Entwicklung in diese Richtung.

Im nächsten Schritt wäre es beispielsweise denkbar, den Comic in den Wizard des Einladungsprozesses mit einzubinden. Die Anwender beschwerten sich, dass dieser zu lange dauert (siehe Usability-Problem *GettingStarted-2*). Diese Wartezeit könnte auf Seiten des Hosts zum Beispiel mit diesem Comic überbrückt werden.

Nichtsdestotrotz sollte das primäre Ziel immer die Anpassung von Saros an das mentale Modell der Benutzer sein und nicht umgekehrt.

# A Interviews

# A.1 Vorbereitung der Interviews

# Auswertung des Saros Fragebogens

response_id	#	done		Answer
2	1	<b>V</b>		i managed to teach php in a short time to another developer
3	2	<b>V</b>		Fix some a script.
4	3	✓ ✓		a semi-completed cave generation framework for a new-style spelunking game  Testing Saros only.
5 6	5	<b>V</b>	[5]	
9	6	<b>V</b>		too many, i didn't have much time so i've accoplish a few
10	7	<b>V</b>		We wanted to pair program on a python (PyDev project) file.
11	8	<b>V</b>		Testing collaborative programming
12	9	V		sdfsfds
15	10	<b>V</b>		] 2.5d raycast engine
17	11	<b>V</b>		] pair programming
18	12	<b>V</b>		] No goals at allI just logged to update my project.
19	13	V	[13	] Development work on 2-person-project.
20	14	<b>✓</b>	[14	l] Testing
21	15	V	[15	ij Teaching friend Java
22	16	<b>V</b>		i) 2 people editing same project, teaching one another
23	17	V		] We use it for remote pair programming (Java/JS).
24	18	<b>V</b>		] Try out saros and my new jabber server that was created for this purpose
25	19	<b>V</b>		l) Loads.
26	20	<b>V</b>		)] pair programming
27	21	<b>V</b>		Implementing a cache while remote pairing.
28	22	✓		] I started learning how to make games in python with a friend!
29 30	23 24	✓		I) Just a test. I) Solving an exercise task with my exercise partner. We didn't find a meeting-date, so we used saros - with success.
31	25	✓		i) A few goals, but mostly modeling the solution of a problem that will be implemented next session
32	26	<b>V</b>		of A rew goals, but mostly modelling the solution of a problem that will be implemented flext session  if just collabration purposes
33	27	<b>V</b>		J Some cool stuff.
34	28	<b>V</b>		started on my plugin with my friends, for bukkit (bukkit.org) using there api. we've come to a start, basicly Twitter in minecraft. :P
35	29	V		of nothing i watched porn instead
37	30	V		) to complete our homeworks for university
38	31	V		collaborate for a minecraft mod using MCP
42	32	<b>V</b>		] Homework.
49	33	V	[33	] Pair programmer with remote developer in same company.
50	34	V	[34	l] to be able to collaborate real time on a piece of code we were working on, and also to be able to look at classes together as we spoke about ther
54	35	V	[35	i) scolar coding
58	36	V		is) make programming with a friend easy
67	37	<b>✓</b>		] Get more help on the spawning command. (QA3: "[] mistakes we fixed")
69	38	V		] Working with team mates in projects
70	39	V		] collaborative code review.
71	40	<b>V</b>		] Develop a python prototype with some friends, collboratively
72	41	V		] Finished some code i had todo.
73	42	<b>V</b>		! testing it out
74	43	<b>V</b>		l testing for remote pairing
75 76	44	✓		I) chat and revision I) Alot of coding done. Now we can do same amount of coding as we did before but in half the time
77	45 46	✓		i) Not of coding done. Now we can do same amount of coding as we did before but in hair the line.  i) We edit multiple classes for the app (QA1: "[] editing code for our android app.", QA3: "[] flushed to the others.")
78	47	<b>V</b>		The eart manage classes for the app (art : [] earling code for our analold app. ; arc. [] mustled to the outers. )
79	48	<b>V</b>		s) Work on one project with multiple users at the same time and in the same document. Prepare Tasks for University in a group.
80	49	<b>V</b>		Work together on files of a c project
81	50	<b>V</b>		) Simple co-editing of a Java source file.
82	51	<b>V</b>		We want to fix different classes, which was very easy to do with this plugin. (QA1: "Me and my friend program a game")
83	52	<b>V</b>		] just working together :D (QA3: "[] the others have to")
84	53	V		Working remotely on the same project, both being able to participate
85	54	<b>V</b>		l] Help newb with programming
86	55	V	[55	] Trying out software
87	56	<b>✓</b>		s) remote pair programming
88	57	<b>V</b>		] group programming and debugging.
89	58	V		] Have a frient help me debug some code.
90	59	<b>V</b>		l) Synced development for a college lab assignment.
91	60	<b>V</b>		] I helped someone with his problems in Java by tutoring through his project.
92	61	<b>V</b>		] We are learning extreme programming and we used Saros to practice Pair Programming. This worked very well and was extremely easy.
94	62	<b>V</b>		] I wanted it to work so we could sync our files.
95	63	<b>V</b>		)] Finishing our java coding and exporting.
96	64	<b>V</b>		o] colaborative work
97 98	65 66	<b>V</b>		S) Completing a mod S) World peace
98	67	✓		ly world peace ] getting an impression if and how it works (> it does!;-)
100	68	<b>V</b>		I getting at impression if and now it works (> it does : ,-)  Reeping a stable connection between users and see real-time updates.
100	69	<b>V</b>		ly Reeping a stable connection between users and see real-time updates.
101	70	<u>~</u>	-	l) Start coding our new game, The First Towner
102	71	<b>V</b>		] The goals is to check and correct some mistake on a java source file
103	72	<b>V</b>		I test the usability
105	73	V		just socks5 connection
106	74	V		) Paring.
107	75	<b>V</b>		] Have been curious about Saros for a long time and finally sat down to try it out.
108	76	V		) FUN
109	77	V		I Improve and ease Pair Programming
111	78	V		0 Update code together simultaneously.
112	79	<b>V</b>		l) test
113	80	<b>V</b>	[80	

_				Categorized Answers
Category	Goal Type	id 2	#	Answer
		21	15	i managed to teach php in a short time to another develope Teaching friend Java
		22	16	2 people editing same project, teaching one another
		28	22	I started learning how to make games in python with a friend!
teaching / learning	End Goal	67	37	Get more help on the spawning command. (QA3: "[] mistakes we fixed")
icuring		85	54	Help newb with programming
		91 92	60 61	I helped someone with his problems in Java by tutoring through his project.  We are learning extreme programming and we used Saros to practice Pair Programming. This worked very well and was extremely easy.
		109	77	Improve and ease Pair Programming
		30	24	Solving an exercise task with my exercise partner. We didn't find a meeting-date, so we used saros - with success.
work on		37	30	to complete our homeworks for university
school / uni	End Goal	42 54	32 35	Homework. Scolar Coding
assignments		79	48	Work on one project with multiple users at the same time and in the same document. Prepare Tasks for University in a group.
		90	59	Synced development for a college lab assignment.
		4 15	3 10	a semi-completed cave generation framework for a new-style spelunking game
		28	22	2.5d raycast engine I started learning how to make games in python with a friend!
		34	28	i started on my plugin with my friends, for bukkit (bukkit.org) using there api. we've come to a start, basicly Twitter in minecraft. :P
free time	End Goal	38	31	collaborate for a minecraft mod using MCP
games and apps)	Liid doai	77	46	We edit multiple classes for the app (QA1: "[] editing code for our android app.", QA3: "[] flushed to the others.")
		82 97	51 65	We want to fix different classes, which was very easy to do with this plugin. (QA1: "Me and my friend program a game")
		101	69	Completing a mod  We wanted to code a Minecraft mod.
		102	70	Start coding our new game, The First Towner
company	Goal or Task	49	33	Pair programmer with remote developer in same company.
		-		Testing Cause sub-
		5 10	8	Testing Saros only. Testing collaborative programming
		20	14	Testing collaborative programming
		24	18	Try out saros and my new jabber server that was created for this purpose
		29	23	Just a test.
		73	42	testing it out
first time usage	Goal or Task	74 86	43 55	testing for remote pairing Trying out software
		99	67	getting an impression if and how it works (> it does!;-)
		100	68	Keeping a stable connection between users and see real-time updates.
		104	72	test the usability
		105	73	just socks5 connection
		107 112	75 79	Have been curious about Saros for a long time and finally sat down to try it out.
		112	/9	test
		11	7	We wanted to pair program on a python (PyDev project) file.
		17	11	pair programming
		19	13	Development work on 2-person-project
		23	17	We use it for remote pair programming (Java/JS).
		26 27	20 21	pair programming
		49	33	Implementing a cache while remote pairing.  Pair programmer with remote developer in same company.
DF-	-	50	34	to be able to collaborate real time on a piece of code we were working on, and also to be able to look at classes together as we spoke about ther
DPP	Task	58	36	make programming with a friend easy
		74	43	testing for remote pairing
		78 84	47 53	Work on a project together with my buddy  Working remotely on the same project, both being able to participate
		84	56	Working remotely on the same project, both being able to participate remote pair programming
		92	61	We are learning extreme programming and we used Saros to practice Pair Programming. This worked very well and was extremely easy.
		106	74	Pairing.
		109	77	Improve and ease Pair Programming
		69	38	Working with team mates in projects
		71	40	Develop a python prototype with some friends, collboratively
DPP > 2	Task	77	46	We edit multiple classes for the app (QA1: "[] editing code for our android app.", QA3: "[] flushed to the others.")
		82	52	just working together :D (QA3: "[] the others have to")
		88	57	group programming and debugging.
		32	26	just collabration purposes
		72	41	Just consultation purposes Finished some code i had todo.
collaboration		76	45	Alot of coding done. Now we can do same amount of coding as we did before but in half the time
(unknown, if #	Goal or Task	80	49	Work together on files of a c project
users= 2 or >2)		81	50	Simple co-editing of a Java source file.
		95 96	63 64	Finishing our java coding and exporting. colaborative work
		30	04	OURSE TOTA
modeling (not	Goal or Task	31	25	A few goals, but mostly modeling the solution of a problem that will be implemented next session
implementing)	Goal of Task	31	25	n rew goals, but mostly fillodelling the solution of a problem that will be implemented next session
		18	12	No goals at allI just logged to update my project.
synchronize	End Goal	18 94	12 62	No goals at allI just logged to update my project.  I wanted it to work so we could sync our files.
project		111	78	Update code together simultaneously.
		3	2	Fix some a script.
		67	37	Get more help on the spawning command. (QA3: "[] mistakes we fixed")
debugging	End Goal	82 88	51 57	We want to fix different classes, which was very easy to do with this plugin.
		88	58	group programming and debugging.  Have a frient help me debug some code.
		103	71	The goals is to check and correct some mistake on a java source file
review	End Goal	70	39	collaborative code review
		75	44	chat and revision
	Experience Goal	108	76	FUN
have fun		.50	70	
have fun	Experience doar			
have fun	Experience doar	6	5	k
have fun	Experience don	9	6	too many. i didn't have much time so i've accoplish a few
have fun	Experience doar	9 12	6 9	too many. i didn't have much time so i've accoplish a few sdfsfds
have fun	-	9 12 25	6 9 19	too many. i didn't have much time so i've accoplish a few sdfslds Loads
	-	9 12 25 33	6 9 19 27	too many, i didn't have much time so i've accoplish a few sdfsfds Loads Some cool stuff.
	-	9 12 25	6 9 19	too many, i didn't have much time so i've accoplish a few saffsids Loads Some cool stuff. nothing i watched pom instead
	-	9 12 25 33 35	6 9 19 27 29	too many, i didn't have much time so i've accoplish a few sdfsfds Loads Some cool stuff.

# Interviewleitfaden

05.05.2012

# Anfang = Begrüßung

- Ich bin ...
- Meta
  - Interview
    - · Mein Ziel ist es ...
    - · Dauer: ca. 45 min
    - Art der Fragen KEIN richtig oder falsch
  - · sonst
    - Gegenüber wird nicht getestet
    - sondern hilft mir Saros zu verbessern
- Alle Angaben werden selbstverständlich anonym und ausschließlich für wissenschaftliche Zwecke im Rahmen meiner Abschlussarbeit verwendet.
- Fragen?
- Einverständnis Aufzeichnung des Gesprächs
- · Gerät einschalten
- erneutes Einverständnis für Aufzeichnung des Gesprächs

05.05.2012

# **Interview**

# einfache (Einstiegs-) Fragen

- Würden Sie mir erzählen wie lange Sie schon bei \_\_\_\_ arbeiten?
  Wie sind Sie zu \_\_\_\_ gekommen?

# Zielgruppe - Allgmein (15min)

	· · · · · · · · · · · · · · · · · · ·	
Fähigkeiten	Computerkenntnisse - Wieviele Stunden pro Woche? - Wofür nutzen Sie den Computer auf Arbeit/ i.d. Freizeit? Vielfalt	
	Eclipse - Wieviele Stunden pro Woche? - Wofür?	
	DPP - Im Folgenden interessiert mich die Zusammenarbeit, bei der Sie mit Ihren Kollegen nicht im selben Raum sitzen: Wieviel Erfahrung haben Sie mit dieser Form der Zusammenarbeit? • Wie läuft die Zusammenarbeit mit Ihren Kollegen üblicherweise ab? • Wer macht von Ihnen was?	
Aktivitäten: Nutzungshäufigkeit	- Wie häufig Arbeiten Sie zusammen? pro Woche - Wieviele Stunden Arbeiten Sie zusammen? pro Einheit  Dauer	
Einstellung	- Was halten Sie von dieser Form der Zusammenarbeit? Spaß, anstrengend, produktiver allein, Mittel zum Zweck?	
Aktivitäten: Nutzungsart	- Wieviele Kollegen? - Wieviele Projekte? - Was für Projekte = Projektdetails (Art, Sprache, Größe) - Woher wissen Sie, wann Ihr Kollege mit Ihnen zusammenarbeiten möchte? Verabreden zur Zusammenarbeit	
Aktivitäten: Nutzungskontext	- Würden Sie kurz Ihr Arbeitsumfeld beschreiben? - Welche technischen Mittel stehen Ihnen für die Zusammenarbeit zur Verfügung? - Welche nutzen Sie? Bsp.: Maus Tastatur (Headset, Monitor(e),)	
Motivation + Ziele	- Warum arbeiten Sie mit Ihren Kollegen zusammen? - Was macht Ihnen bei Ihrer (Zusammen-)Arbeit am meisten Spaß? - Was würden Sie vorziehen nicht zu tun? - Wenn Sie mit Ihren Kollegen zusammenarbeiten: Woran arbeiten Sie dann?	
Fähigkeiten	Englisch - Abgesehen von deutsch, welche anderen Sprachen verstehen Sie? • ja: Wie haben Sie sich Ihre Englisch-Kenntnisse angeeignet? • nein: Welche Sprachen in der Schule?	

05.05.2012

# Saros - Allgmein (10min)

- Haben Sie Saros schon einmal verwendet?
- Wie haben Sie Saros kennengelernt?

Aktivitäten: Nutzungshäufigkeit	Saros - Nutzung - Wie oft haben Sie Saros bisher verwendet? - aktuell: Wie häufig pro Woche? - Wieviele Stunden nutzen Sie Saros? pro Einheit	
Motivation + Ziele	<ul> <li>Wie wichtig ist Saros für Sie?</li> <li>Wenn Sie mit Saros arbeiten: Woran arbeiten Sie dann?</li> <li>Neugier/ Angst vor Fehlern</li> <li>Was verstellt im Saros Menü?</li> <li>Wenn ja, was?</li> <li>Wenn nein, warum nicht?</li> </ul>	
Aufgaben	- Welche Funktionen sind für Sie in Saros am wichtigsten? bzw. Welche verwenden Sie am häufigsten? - Gibt es etwas, wodurch Saros noch nützlicher für Sie wäre? - Gibt es etwas in Saros, was Sie für überflüssig halten?	

Welche Sachen liefen gut (Freude)? Was richtig schlecht (Frust)?

# Saros-Konzepte (20min)

Roles-1	- Sie wollen mit Ihrem Kollegen zusammenarbeiten. Wie gehen Sie vor?										
"host"	- Wie entscheiden Sie, wer wen einlädt? Information "Host"										
Follow-1	- Sie arbeiten an einem Ihnen unbekannten Package. Aber Kollege würde Sie einweisen und erklären. Wie würden Sie weiter vorgehen?										
	ganzes Package / Ordner     Woher klar, dass Aktion erfolgreich?										
Follow-2	- Situation andersherum! Sie = Experte, Kollege = Anfänger;)										
OpenFile-5 Cursor	- Sie = Experte. Kollege folgt Ihnen. Sie erklären, worum es ein einer Datei. Sehen großen Ausschnitt der Datei. Sie möchten, dass Kollege weiß welchen Teil Sie gerade ansehen, den Sie erklären. Wie stellen Sie das sicher?										
OpenFile-1 grün fokus	- Zusammenarbeit - Sie sprechen über die Datei, die Sie vor sich sehen. Sie haben das Gefühl Sie reden aneinander vorbei - Idee: sehen nicht die gleiche Datei. Wie überprüfen Sie Ihre Annahme?										
	- (A: Kollege fragen)> Und dann?										

05.05.2012

OpenFile-3 Viewport L	<ul> <li>Angucken gleicher Datei - lang - 600 Zeilen. Sie gehen gemeinsam durch die Datei und versuchen diese zu verstehen. Um das zu erleichtern, wollen Sie beide möglichst den gleichen Ausschnitt sehen. Wie würden Sie vorgehen?</li> <li>Follow Modus = sobald der andere scrollt, Sprung zu der Position</li> </ul>
OpenFile-4 Viewport R	- immer noch beide in derselben Datei. Kollege fängt an über einen Teil zu reden, den Sie anscheinend nicht sehen. Sie wollen aber wissen, was Ihr Kollege sieht. Was machen Sie?
Session-5 sync Datei	- voller Workspace, viele Projekte offen. Sie sind nicht mehr sicher, welches davon Ihr über Saros zusammenarbeitet. Wie gehen Sie vor?
Roles-1 "host"	- Wie entscheiden Sie, wer als erster die Saros Zusammenarbeit beendet?

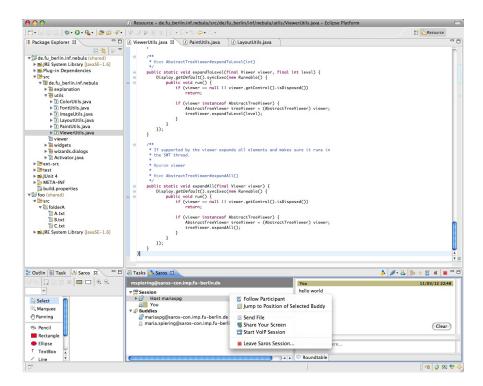
# **Demographie**

- Alter
- Geschlecht
- Ausbildung
  - Abschluss
  - angestrebter Abschluss

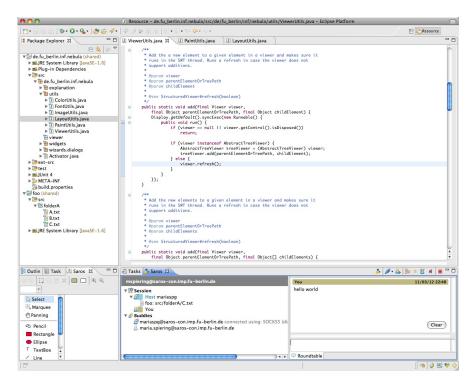
# **Ende = Verabschiedung**

- alle Fragen geklärt
- sonstige Fragen vom Gegenüber?
- Gerät ausschalten
- Meta
  - Vielen Dank!
  - Vertraulichkeit der Daten Alle Angaben werden selbstverständlich anonym und ausschließlich für wissenschaftliche Zwecke im Rahmen meiner Abschlussarbeit verwendet.
- Kontakdaten
  - Name
  - E-Mail

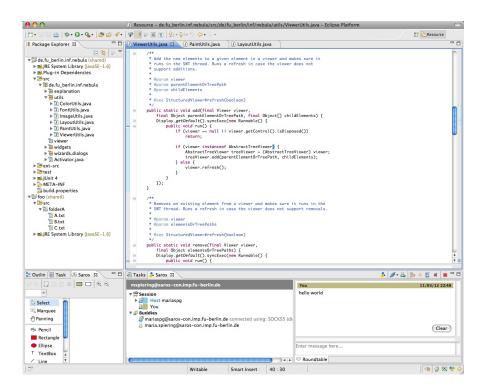
# A.1.3 Screenshots



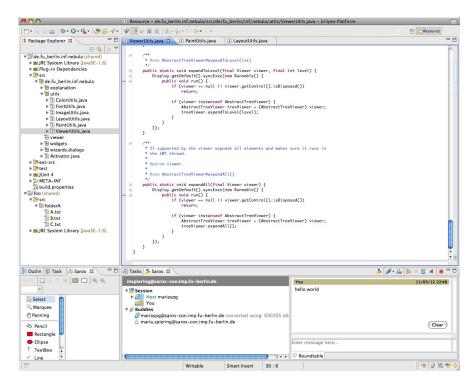
Screenshot zum Zeigen des Kontexmenüs



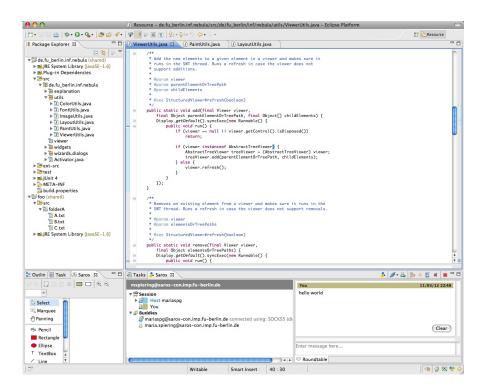
Screenshot zum Zeigen des Dateipfads im Saros-View



Screenshot zum Zeigen der linken Viewport-Annotation



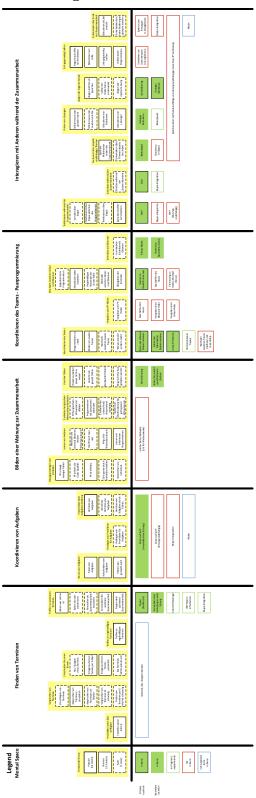
Screenshot zum Zeigen der rechten Viewport-Annotation

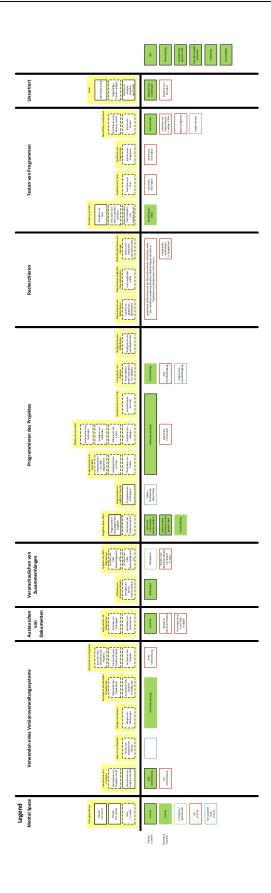


Screenshot zum Zeigen der rechten Viewport-Annotation

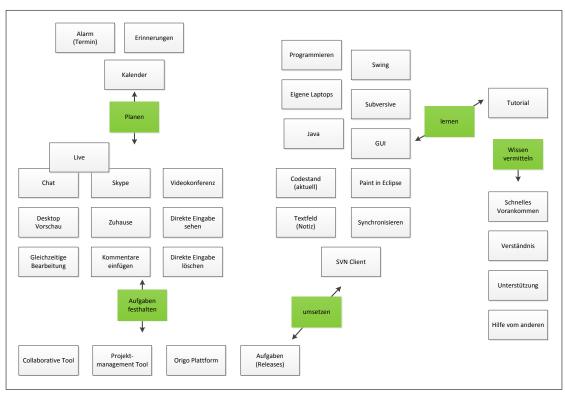
# A.2 Ergebnisse der Interviews

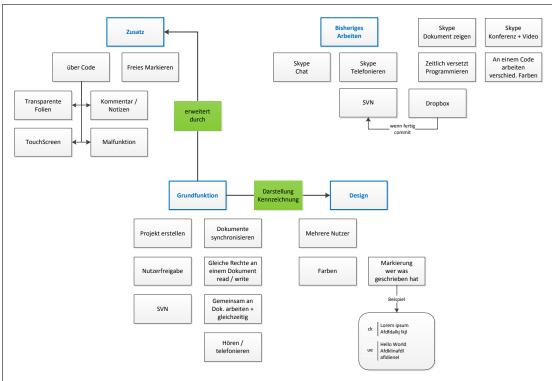
# Das mentale Modell zum allgemeinen Verständnis von Zusammenarbeit





# SLT-Modelle der Versuchspersonen 7 und 8





# Auswertung der Zielgruppe

	1	2598	3,0			2	-	Java	zu Hause	Laptop	
	ntiell	80	3,0	I	1-5 Mal	aktiv zu 2. (Team: 4-6)	τ-	Java	zu Hause	- Laptop (13 / 14 Zoll)	
	potentiell	20	3,0	3,0		aktiv zu 2. (Team: 4-6)	τ-	Java	Uni oder zu Hause	- Laptop	
	7	24.90	3,0	1,2		2	-	Java	Büro mit mehreren Personen	PC mit 2 Monitoren und einem Telefon	
		05	1,5	1,5		zu 2.			Büroraum für 3 Personen	- PC - Maus - Tastatur - Telefon - 2 Monitore	
	Arbeit	8	5,0	1,0		zu 2. manchmal 3) (manchmal 3)	7 ingesamt, 1 zur gleichen Zeit	Java	Büroraum für Büroraum für 2 Personen 3 Personen	- PC - Maus - Tastatur - 2 Monitore (16:9, 4:3)	
Aktivitäten		03	2,5	1,0		zu 2. (manchmal 3)	<del>-</del>	1	Büroraum für 2 Personen	- PC - Maus - Tastatur - 2 Monitore (je 22 Zoll)	
	Jak	2139	2,2	2,7		2	-	Java	zu Hause	PC mit 1-2 Monitoren und einem Headset (+evtl. Notebook)	
		90	1,5	4,0		aktiv zu 2. (Team: 4-5)	-	- Java (50-70 Klassen) - Javascript (10 Module)	zu Hause (bei sich selbst oder jmd anderem)	- PC - Headset - 2 Monitore (je 24 Zoll) oder - Notebook (14,2 Zoll)	
	Freizeit	02	2,0	1,5		aktiv zu 2. (Team: 4)	-	Java, Webseiten	zu Hause	- PC - Maus - Tastatur - Kopfhörer (inkl. Mikro) - 2 Monitore (je 24 Zoll) - Telefon	
		10	3,0	2,5		zu 2.	-	Java - bis 100MB	zu Hause	- PC - Maus - Tastatur - Headset - 1 Monitor oder - Laptop	
	Zielgruppe	Vpn	Anzahl	Dauer je EInheit		# Kollegen GES	# Projekte GES	Projektart	Umfeld	Hilfsmittel	
	Ziel		:: :::::::::::::::::::::::::::::::::::	nadiiigkeit			Art			Kontext	

	0	2612	34,5		AVG	7,25	Java						
	tiell	80	34,0	Java, C# UML	6h pWt + We	7	Java	0-8h	PP durch halbjähriges universitäres Praktikum	Skype, OneNote	Aufgaben in Teilprojekte aufteilen	lesen, 1 Jahr Ausland, 2 Unikurse	
	potentiell	20	35,0	Java, Protégé, IBM Rational DOORS, UML	6-8h pT	7,5	Java	5-10h	PP durch halbjähriges universitäres Praktikum	SVN, Skype	Aufgaben	Schule (ab 6. lesen, 1. Klasse, Leist- Ausland, ungskurs), 2 Unikurs	
	C L F	5617	38,2		AVG	11,2	Java	16h = 03+04					
		05	27,5	Qualitätsmg mt, Testen, Javascript, JQuery		1,5	Java		arbeitet/ testet hauptsächl. alleine	bei Problemen, an einem PC		Schule, Filme	
	Arbeit	04	55,0	Java, Java, Python, script, InDesign Basic, XSLT	10-12h pT	20	Java	5T * 4h	1	bei Problemen, persönlich	1	lesen, hören, schreiben, Schule, CNN, BBC	
Fähigkeiten		03	32,0	Java, Javascript, Visual Basic, XSLT		12	Java	4T * 3h	1	bei Problemen, persönlich	"Einer tippt, der Andere schaut mit rauf", wechseln ab und zu	lesen, Schule (ab 5. Klasse)	
Œ.	C L F	2539	30,8		AVG	20,2	Java						
		90	47,5	Java, Ruby, Freemarker, Javascript, MongoDB, C, C++	TO-11h pT	38	C, <b>Java</b> (Closure)	%08	ī	Git + Github, Skype, E- Mail, Jabber, ICQ, Skype, Facebook	Aufgaben verteilen, Übungszettel ohne	Schule, BA auf Englisch	
	Freizeit	02	20,0	Web, XML, Javascript, PHP, Python, Java, Datenbank		15	Java, (PHP, Python, XML)		1	Jabber / Pidgin	Aufgaben verteilen	Schule + Anglistik- Studium	
		Б	25,0	Web, CMS, Java, Html, Notepad++		7,5	Java		PaarProgram mierung (Driver und Observer)	Skype	TestDriven + Ping Pong: Test <> Impl.	1 Jahr Amerika	
	Zielgruppe	Vpn	Std pro Woche	Vielfalt		Std pro Woche	Vielfalt		Erfahrung	Ablauf	Arbeitsverting.	Englisch	
	Ziel			DG .			Eclipse			D/PP		En	

			Ein	Einstellung				
Zielgruppe		Freizeit			Arbeit		potentiell	ntiell
Vpn	04	02	90	03	40	05	20	80
Einstellung PP / DPP	PP besser "perschilch", effizient, Zeit sparend, da keine Zeit verloren geht irgendwo hinzufahren	"schwierig" "mehr Koordinationsaufw and" "es geht mehr schieg" "PP geht nicht [] schade"	"Ich finde es Ok, aber ich find es immer besser, wenn ma zusammen vor Ort arbeitet. Da sind die Kommunikationsw ege viel kürzer" "laber] manchmal ist man effektiver, wenn man sich nicht trifft"	persönl. Zusammenarbeit: "gut von der Idee her" "angenehm"	and and Video	"sehr gut" "wenn jetzt einer "persönlicher mitarbeiten würde, onfakt ist, so dem wüsste ich eines als über nicht, wen hat er Skype oder vorbereitet und flexibler" brauch ich das für spätere Tests, da kann ich es auch gleich selber machen"	"ganz gut, super" "ich halt davon "man kann [] nicht soviel von Wissen austauschen" "ich priprogramming." hab viel dazu mir das nur was, wenn jemand stärker ist als ich [] da ich das [Programmieren] selber nicht so gut kann"	"ich halt davon nicht soviel von dem Pairprogramming. im Prinzip bringt mir das nur was, wenn jemand stärker ist als ich [] da ich das [Programmieren] selber nicht so gut kann"

# Übersicht über die Ergebnisse der Saros-Konzepte

Kategorie Coordination Follow Session	Schiüssel Follow-1 Follow-2 Session-5	Erläuterung Wem folge ich? Wer folgt mir? Doppelpfeil		P3 · P3	P1 P3 P4 P5	hen S × E		N A			richtig v	s P4	Ď	Anmerkungen  - walrgenommen und grundsätzlich verstanden - aber unklar, wann Follow-Modus beender (2x) - und unsicher, ob noch aktiv -> Rückfragen (2x) - gucken in der Saros View (1x) - durch	Fazit  Follow-1/-2 = OK  - Erklärungen von Sonderfällen: - externe Datelen (reu: Follow-3) - und zusammengesetzte - durch "shared" alle Fälle abgedeckt - könnte noch prägnanter sein, gennge
OpenFile	OpenFile-1	"shared" grüner Punkt Saros View		× × ×	×××	× >	· × ·	× × ×	> × ×	> × >	× × ×	×××	×	nicht wahrgenommen - Alternativen verwendst: Vlewport (1x), Follow (1x), frager (1x) - für >2 Teilnehmer nicht gelegnet, de aline eindeutige - Inlehmerzuordung nicht inöglich - neu / teilweise wahrgenommen + verstanden	Farailität - Verbesserung notwendig
	OpenFile-2	gelber Punkt		×	×	×		×	×	×	×	×	×	- nicht wahrgenommen - und für >2 Teilnehmer nicht geeignet, da eine eindeutige Teilnehmerzuordnung nicht möglich	- Verbesserung notwendig
	OpenFile-3	Viewport L Viewport R	1 1	×	2 2	D D	1 1	×	×	×	×	×	××	<ul> <li>- Jump to this Position, anstelle (1x)</li> <li>- Unterschied unklar (3x)</li> <li>- verwendet + falsch verstanden / für den andem gehalten (2x)</li> </ul>	- Verbesserung notwendig - Zusammenlegung des Problems, da Unterschied unklar (OpenFile-3/-4)
	OpenFile-5	Cursor	>	>	N N	<b>&gt;</b>	>	>	>	×	<b>N</b>	<b>N</b>		wahrgenommen und verstanden     - aber häufig wurde marklett, anstelle des Cursors (3x) -     - Wermutung: Markleren kann mehr zeigen Cursor-Position     zuletzt geschriebenes und Cursor-Parböne zu ähnlich	- OpenFile-5 = OK - aber, Farbtöne des zuletzt geschriebenen und Cursor/Markierung zu ähnlich (neu: OpenFile-6)
Miscellaneous										-					
Roles	Roles-1	Host	1 1	××	> >	> >	1 1	1 1	××	××	× ×			<ul> <li>- besofulfiellen Zusammenarbeit zu Ende, speichern, beenden (2x)</li> <li>- vermischen Host mit Follow-Modus / Driver (2x)</li> <li>- Schreibrecher, klart (2x)</li> </ul>	esserung notwendig
			Legende: - ke	ande: keir	ne Aus	gende: keine Aussace (da nicht erwähnt/oefract)	da nic	ht erv	ähnt/a	efraat					

# Auswertung der Beschwerden und Wünsche

# Beschwerden

Kategorie	Erläuterung	ID	Zitat	Part	Time	Schlüssel
Awareness Focus	Synchronisation von Dialog- Fenstern	01	"Refactoring, hat man ja sehr viel wenn er da ein Rename macht, das bekommt man als Observer dann nur teilweise mit oder gar nicht mit, je nachdem welche Dialoge das sind. Dann sind diese Sprünge, wenn er eine Methode ausgelagert, die wirken dann so plötzlich, weil man den Dialog vorher nicht gesehen hat. Das wäre eine Verbesserung, wenn man das sehen könnte Das wichtigste wäre, dass man die Dialoge sehen könnte. Wenn ich rechtsklick drücke und irgenwas mache."	1	27:20	Focus-1
OpenFile	Debug	03	"ich hab einen Brechpunkt gesetzt, ich weiß aber nicht ob er den auch gesehen hat"	2	09:20	OpenFile-7
OpenFile	Wechsel zwischen Session-Datei und nicht- Session-Datei	03	"Das hatte ich mit XXX auch. Wie war denn das. Die Aufgabenstellung hatten wir, die war glaub ich nicht synchronisiert, weil es kein Sourcecode war oder sowas. Auf jeden Fall hab ich mir die Aufgabenstellung angeguckt, weil ich nochmal nachschlagen wollte, wie war denn das jetzt.  A ich wusste nicht, dass er nicht weiß, dass ich dort bin. Er war immer noch im Sourcecode und wusste auch nicht, dass ich gerade die Aufgabe lese. Und B, ich hab dann irgendwas von der Aufgabenstellung erzählt: Guck mal da unten. Und wer war ganz woanders. Also das war schon ein Problem."	2	17:44	siehe Follow-3
Explicit Sync + Focus	Synchronisation des Package Explorers	04	"Soweit ich weiß ging das mit Saros nicht, dass man sieht, wo wer in einem Package ist. Da gab es Probleme, das war komplett gar nicht abgedeckt mit Saros. Man konnte nur erklären und sagen jetzt bin ich in der Datei oder in dem Package mit den Unterklassen und die Klasse macht das und das. Jetzt müsste man als Rolle Client nachverfolgen, was der Host also der Erklärer dort meint. Es war nicht so, wenn man das Package öffnet der Server oder der Host hat jetzt die Utils-Klasse das Package aufgemacht und möchte einem jetzt darüber etwas erklären, dann ist es bei dem Zuschauer nicht passiert. Es hat sich nicht geöffnet, das hat sich nicht synchronisiert. Also die visuelle Rückmeldung, wo er sich befindet, was er gerade macht, das hat komplett gefehlt. Also als Benutzer hat man das auch gar nicht festgestellt. Zu aller erst war es so, dass einer erklärt hat. Jetzt mach ich das und das auf. Und da ist gar nichts passiert und man hat sich gefragt, soll das so sein oder macht der jetzt gar nichts. Dann hat man festgestellt bei mir passiert gar nichts, du erklärst grad was bis man am Ende rausgefunden hat, dass das nicht synchronisiert wird. Wäre vielleicht nicht schlecht, wenn man da eine Botschaft gehabt hätte, das wird jetzt nicht synchronisiert, kann nicht synchronisiert werden. Das hat komplett gefehlt dieses Feedback."	2	06:13	Focus-3
OpenFile Follow	-	04	"Was mir richtig schlecht gefallen hat, dass man nicht XML-Dateien sharen konnte. Dass das nur bestimmte Views, also meistens nur bestimmte Klassen und JavaBestimmte Reiter haben nicht funktioniert."	1	24:11	siehe Follow-4
Getting Started	Anlegen eines XMPP-Kontos	06	"Anmeldung über XMPP-Account Verbindungsprobleme das war eine Hürde. Ok, ich will Saros benutzen, also brauch ich ein XMPP-Account, ich hab keinen, also muss ich einen anlegen. Das hat gestört."	1	36:35	GettingStarted- 1

# Wünsche

Kategorie	Erläuterung	ID	Zitat	Part	Time	Schlüssel
Awareness Focus	Synchronisation der Programm- Ausführung	01	"Was cool wäre, wenn ich sowas wie einen "run-shared" Knopf hätte. So dass ich das Programm nicht nur bei mir ausführe, so dass halt nur das Programm-Fenster und die Konsole und so weiter, also meine GUI zu dem anderen übertragen wird und der das sieht. Per Knopf-Druck auf "run", dann mein Programm starte und der andere gegenüber das auch siehtspezielleres ScreenSharing."	1	29:07	
ExplicitSync	-	03	"er musste das [Java-Programm] separat bei sich starten, das war auch ein Problem. Da hätte ich mir gewünscht, wenn ich das Ding starte, also das Programm ausführe dann wird das bei mir lokal gestartet und die Synchronisation ist weger musste F11 drücken und ich musste F11 drücken ich hätte es gern selbst gemacht, aber das nicht geklappt, weil ich hatte dann eine Konsolenausgabe da und die hatte er nicht"	2	09:39	Focus-2
Color	-	03	"verschiedene Farben hatten wir gemacht, das ist auch ganz nützlich wir würden uns dann einigen, der eine kriegt die eine Farbe der andere kriegt die andere Farbe und dann behalten diese beiden Leute auch immer diese Farbe. Dann ist es einfacher sich zu orientieren, weil wir hatten das vorhin bei der Session, da waren unsere beiden Farben vertauscht und das war komisch und ich bin dann immer durcheinander gekommen, mit dem wo er gerade ist und wo ich gerade binEs wäre glaub ich gut, wenn man sagen könnte Person XY hat die Farbe rot und die Person ABC blau oder so. Dann ist das einfach immer so, man sieht dann sofort an der Farbe, wer das ist."	2	15:54	Session-6
Comm	VoIP-Feature	03	"Wir haben miteinander gesprochen über Skype und es wäre natürlich super, wenn Saros gleich die Möglichkeit integrieren würde, dass zwei Leute mit Sprache sich miteinander unterhalten. Notfalls auch einen Chat, aber das halte ich für kritisch. Also lieber telefonieren, das halte ich für besser."	1	28:26	siehe Comm-Verb-1
GettingStarted	XMPP-Konto anlegen	04	"Das mit dem Jabber-Account fand ich jetzt auch nicht so prickelnd, dass das da so über eine Server da läuft. Das müsste mehr dedicated sein. Das man sagt, ich mach jetzt hier den Server, den Host und connecte dich jetzt einfach auf meine IP oder so."	1	23:50	GettingStarted- 1
GettingStarted	Einladungs- prozess	04	"Die Usability, das Benutzen an sich, schnell einfach. Also dass man sich nicht Gedanken über Saros machen muss. Das es selbstverständlich istAm Besten ist es man schmeißt Eclipse an, sagt Connect, Einladen und ab geht die Luzi. Das es schnell geht"	1	21:51	GettingStarted-
GettingStarted	Einladungs- prozess	06	"Was praktisch wäre, wenn man sehr schnell Verbindungen aufbauen könnte. Was cool wäre, wenn mir jemand einen Link schicken würde und der würde dann eine Session initialisieren und dann könnte man direkt an einem Problem arbeiten."	1	32:38	2

# Usability-Problemkatalog

Follow-Mode (bei Dateien außerhalb der Sitzung)	1	Awareness Follow-Mode	Unklarer Status des Follow-Modus (bei Dateien außerhalb der Sitzung)	3
---	---	--------------------------	---	---

#### Follow-3

Vorbedingung: Bob folgt Alice

Saros liefert Bob unzureichende Informationen darüber, wenn Alice Ressourcen öffnet, die nicht Teil der Sitzung sind.

Bob sieht in der Editor Ansicht weiterhin die zuletzt - in der Sitzung - fokusierte Datei. Wird durch Alice wieder eine Datei geöffnet, die über Saros geteilt wird, dann wird Bob diese auch wieder angezeigt.

Es ist nicht klar, dass der Follow-Modus noch aktiv, aber "pausiert" ist.

Anmerkung: In der Saros Ansicht existiert die Information "(following)" inklusive des Pfades der aktuell fokusierten Ressource. Handelt es sich um eine externe Ressource, wird kein Pfad angezeigt.

Quelle: Interviews + UT[Waldmann2012]

- Anwender wirken verunsichert / fragen

# Lösungsansätze:

 Anzeigen des Status des Follow-Modus (fokusierte ist Datei außerhalb der Sitzung; Follow-Modus pausiert und noch aktiv)

Der Anwender muss über den Zustand informiert werden:

- 1. Mögliche Präsentationform der Informationen
  - 1. Alice: BalloonNotification
  - 2. Bob: in der Saros Ansicht anstelle der leeren Pfadangabe z.B. ("non-Session file"); Erweitern der Information "following" auf "following paused"

Arbeitspaket "Follow Mode".

2	Awareness Follow-Mode	Unklarer Status des Follow-Modus (bei zusammengesetzten Editoren)	3
---	--------------------------	--	---

## Follow-4

Vorbedingung: Bob folgt Alice

Saros kann nur simple Editoren synchronisieren. Zusammengesetzte Editoren mit mehreren internen Reitern können von Saros nicht synchronsiert werden, dazu zählen Dateien wie die plugin.xml oder build.properties.

Es ist nicht klar, dass der Follow-Modus noch aktiv, aber der zusammengesetzte Editor nicht synchronisiert werden kann.

Anmerkung: In der Saros Ansicht existiert die Information "(following)" inklusive des Pfades der aktuell fokusierten Ressource. Handelt es sich um einen zusammengesetzten Editor, wird kein Pfad angezeigt.

Quelle: Interviews + UT[Waldmann2012] genervt zurück

- Anwender fragen unsicher und/ oder

2	Awareness Follow-Mode	Unklarer Status des Follow-Modus (bei zusammengesetzten Editoren)	3

## Lösungsansätze:

- 1. Idee von Waldmann2012:
  - Anzeige eines Dummy-Editors, der auf die fehlende Synchronisationsmöglichkeit hinweist, da die Balloon-Benachrichtigung sich als zu subtil herausgestellt hat.
- 2. Anzeigen des Status des Follow-Modus
- 3. Unterstützung zusammengesetzter Editoren

- 1. erwähnte BalloonNotification nicht vorhanden / nicht gesehen
- 2. Mögliche Informationspräsentation:
  - 1. Alice: BalloonNotification ("beibehalten")
  - 2. Bob: in der Saros Ansicht anstelle der leeren Pfadangabe z.B. ("unsupported file format"); Erweitern der Information "following" auf "following paused"
- 3. Zeitlich nicht umsetzbar.

Arbeitspaket "Follow Mode".

3	Awareness Session	Unauffällige Markierung für Projekte der aktuellen Sitzung	1
---	----------------------	---	---

#### Session-5

Die Markierungen an Projekten und Dateien, die Bestandteil einer Sitzung sind, sind sehr subtil und werden nicht wahrgenommen.

Anmerkung: Bei den Markierungen im Package Explorer handelt es sich um ein 8\*8 Pixel große Icons in Form eines Doppelpfeils.

Quelle: Interviews + UT[Waldmann2012] gefunden werden kann

- Stellt Vermutungen auf, wo Information

## Lösungsansätze:

- 1. Änderung durch Waldmann2012:
  - Hinzufügen einer textuellen Suffix-Dekoration an Projekten: "(share)" bzw. "(partial share)"
- 2. Geteilte und nicht geteilte Projekte getrennt darstellen, z.B. durch Teilen der Projekt Explorer Ansicht. (gesehen bei ACE)
- 3. Erklärung der Markierung
- 1. Die Doppelpfeil-Icons wurden von 2 der 4 Teilnehmern wahrgenommen und wiedergegeben. Exakt die anderen 2 Teilnehmer haben sich an der textuellen Information orientiert.
- 2. Diese Form der Darstellung ist leicht zu verstehen und übersichtlich, ist aber bei nur teilweise geteilten Projekte (PartialSharing) unter Umständen verwirrend.

Durch die Erweiterung der Icons um die textuelle Information von Waldmann2012 werden anscheinend viele (vielleicht sogar alle) Anwender erreicht. Auf eine weiterführende Umsetzung kann derzeit verzichtet werden und sollte erst nach weiteren Usability-Tests vogenommen werden.

4 Awareness
Session Beibehalten der eigenen Farbe über
mehrere Sitzungen

#### Session-6

Anwender äußert den Wunsch eine Farbe beibehalten zu können. Es entstand Verwirrung, dass bei einer neuen Sitzung die Farbverteilung eine andere war als zuvor.

Anmerkung: Zum aktuellen Zeitpunkt, werden die verwendeten Farben lokal beim Host gespeichert und auf die nächste Sitzung angewendet. Wechselt von einer Sitzung zur nächsten der Host, so können sich auch die Farben ändern. Die eigene Farbe wird nicht gespeichert, d.h. bei einer unpassenden/ unerwünschten Farben muss bei jeder Sitzung mit einem neuen Host oder einer anderen Reihenfolge der Teilnehmer die eigene Farbe neu angepasst werden.

Quelle: Interviews (Vpn03) + UT[Kahlert2011] + UT[Waldmann2012] - Feature-Wunsch

# Lösungsansätze:

1. Idee von Waldmann2012

Speichern der Farbkombinationen (eigene+fremde), insbesondere versuchen einem Nutzer, der seine Farbe explizit umgestellt hat, auch in einer folgenden Sitzung wieder diese Farbe zuzuweisen. (gesehen bei SubEthaEdit)

Anwender orientieren sich häufig anhand der Farben, weswegen diese Funktionalität sehr hilfreich wäre. An der Umsetzung des Farbmanagements arbeitete S. Rossbach (siehe Gerrit #152). Die Spezifikation des Farbverhaltens und -speicherns wurde noch einmal in Zusammenarbeit mit Franz Zieris überarbeitet. Das Team des Softwareprojekts "Agile Softwareentwicklung in einem Open-Source-Projekt" versucht beides umzusetzen.

5 Awareness Unauffällige Markierung für 1 OpenFile fokusierte Datei	
---	--

#### OpenFile-1

Anwendern ist nicht klar, welche Datei andere Sitzungsteilnehmer gerade im Fokus haben, d.h. gerade im Editor angezeigt wird.

Anmerkung: Im Package Explorer werden für diese Art der Information sehr kleine Icon-Annotationen (8\*8 Pixel) in Form von grünen Punkten verwendet. Dies ist für Sitzungen mit mehr als zwei Teilnehmern ungeeignet, da nicht codiert ist, wer die Datei fokusiert.

Quelle: Interviews + UT[Solovjev2011] + UT[Waldmann2012]

- 1. Änderung durch Waldmann2012:
  - Ergänzung der Saros-View Je Sitzungsteilnehmer einen Eintrag der anzeigt, ob und wenn ja, welche geteilte Datei im Editor angezeigt wird (Angabe des Dateipfads "Projektname: OrdnerA/OrdnerB/Datei.java"). Ein Doppelklick auf den Eintrag löst die Aktion "Jump to user position" aus.
- 2. Erklärung der Markierung

# 5 Awareness Unauffällige Markierung für 1 OpenFile fokusierte Datei

Die Erweiterung der Icons durch die textuelle Information von Waldmann2012 wurde von Anwendern in den Interviews positiv erwähnt, womit dieses vermeintliche Usability-Problem behoben wäre.

Nichtsdestotrotz wäre eine Erwähnung (siehe Ansatz 2) im Rahmen eines "Getting Started Willkommenspakets" sinnvoll: Arbeitspaket "Getting Started"

6	Awareness OpenFile	Unauffällige Markierung für offene Datei(en)	3
	•	` '	

## OpenFile-2

Anwendern ist nicht klar, welche Datei(en) andere Sitzungsteilnehmer gerade offen haben, d.h. im Editor als Tabs dargestellt werden.

Anmerkung: Die bestehende Awareness-Information wird nicht wahrgenommen. Im Package Explorer werden für diese Art der Information sehr kleine Icon-Annotationen (8\*8 Pixel) in Form von gelben Punkten verwendet. Dies ist für Sitzungen mit mehr als zwei Teilnehmern ungeeignet: Es nicht codiert ist, wer die Datei fokusiert und bei mehreren offenen Dateien je Sitzungsteilnehmer ist dies sehr unübersichtlich.

Quelle: Interviews + UT[Solovjev2011] + UT[Waldmann2012]

# Lösungsansätze:

MM → Saros
 Knappe Erklärung der Markierung
 (bisher auf der GettingStarted Seite ganz unten)

Andere ähnliche Anwendungen zeigen keine Lösungen für dieses Problem auf, da sie diese Information nicht darstellen. Auf die Awareness Informationen muss wenigstens rein informell deutlicher hingewiesen werden: *Arbeitspaket "Getting Started"*.

7	Awareness / MM	Bedeutungsunterschied der zwei	2
	OpenFile	Viewports unklar	

# OpenFile-3 und OpenFile-4

Die zwei Viewports links und rechts am Editor werden teilweise nicht wahrgenommen. Sofern doch, dann lösen sie Verwirrung aus, da deren Unterschied unklar ist. Nichtsdestrotz werden sie verwendet, aber dann gehäuft falsch verstanden oder die Funktion des einen Viewports für den jeweils anderen gehalten.

Anmerkung: Der linke Viewport zeigt den derzeit sichtbaren Bereich (Bildschirmausschnitt) der anderen Sitzungsteilnehmer an. Der rechte Viewport veranschaulicht, welcher Bereich innerhalb kompletten Datei gesehen wird.

Quelle: Interviews + UT[Solovjev2011] + UT[Waldmann2012]

7	Awareness / MM OpenFile	Bedeutungsunterschied der zwei Viewports unklar	2

## Lösungsansätze:

1. Saros → MM

Reduzieren auf einen Viewport innerhalb des eigenen Scrollbereiches (rechts). Betrachtet man den gleichen Bildschirmausschnitt ist kein Viewport zu sehen, guckt man verschiedene Bereiche an, wird dieser ober (oder unter) dem eigenen Scrollbalken angezeigt. So wird deutlich, ob sich andere Teilnehmer weiter oben (oder unten) im Dokument befindet. (gesehen bei SubEthaEdit)

- 2. MM → Saros Hinweis und bessere Erläuterung des Unterschieds zwischen den zwei Viewports.
- 1. Gute Lösung, da so keine Verwirrung entstehen kann. Aber darstellungsweise ist untypisch für Eclipse und zeitlich nicht umsetzbar.
- 2. Bisher werden die Viewports unvollständig und ausschließlich in textueller Form auf der GettingStarted-Seite erklärt.

  Arbeitspaket "Getting Started" mit OpenFile-2, Host-1 und GettingStarted-1.

8	Awareness OpenFile	Zu geringer Farbkontrast zwischen Selection und Contribution Annotation	2
		Ailliotation	

#### OpenFile-6

Der Cursor bzw. Markierungen innerhalb der zuletzt geschriebenen Zeichen ist nur schwer zu erkennen, da sich die Farbtöne zu ähnlich sind.

Anmerkung: Saros hinterlegt die zuletzt geschriebenen Zeichen mit der Farbe des Sitzungsteilnehmers. Die Cursor-/ Markierungs-Farbe ist die gleiche, aber etwas heller.

Quelle: Interviews (Vpn05)

### Lösungsansätze:

- 1. Farbtöne des Cursors/ Markierung und der Hintergrundfarbe des zuletzt geschriebenen deutlicher voneinander abheben (Standardeinstellung verbessern).
- 2. Der Ausprägung der eigenen Farbe in den Saros Einstellungen mittels eines Schiebereglers skalierbar machen. (gesehen bei SubEthaEdit)
- Cursor deutlicher hervorheben Sobald etwas geschrieben wird bzw. der Cursor umgesetzt wird, immer den Namen des Sitzungsteilnehmers und dessen Farbe anzeigen. (gesehen bei GoogleDocs)

Eine Kombination der drei Lösungsansätze wäre wünschenswert, ist aber aus zeitlichen Gründen nicht umsetzbar.

Die Standardeinstellung für die Farbunterscheidung (unter der Verwendung der von Kahlert2011 optimalen Farben) sollte bereits so gut sein, dass eine Skalierung und zusätzliche Hervorhebung des Cursors nicht notwendig ist. *Arbeitspaket "Saros-View" (Lösung 1).* 

# 9 Awareness / MM Unklarheit über Synchronisation von Breakpoints

### OpenFile-7

Dem Anwender ist nicht klar, ob Breakpoints von Saros synchronisiert werden oder nicht. Einerseits befinden sich diese innerhalb der Editor-Ansicht, andererseits sind sie nicht Teil des editierbaren Bereiches.

Anmerkung: Saros überträgt keine Breakpoints.

Quelle: Interviews (Vpn03)

- Feature-Wunsch (Beschwerde)

# Lösungsansätze:

- Saros → MM
  - Unterstützung der Synchronisierung
- 2. MM → Saros

Erklärung der Einschränkung (siehe http://www.saros-project.org/DosDonts).

- 1. Geringe Fatalität und zeitlich innerhalb der Arbeit nicht umsetzbar.
- 2. Erweiterung der Erläuterung, welche Informationen synchronisiert werden und welche nicht.

Mögliches Arbeitspaket "Do's & Don'ts" mit Focus-1, Focus-2 und Focus-3.

10	Awareness	Fehlende Synchronisation	2
	Focus	von Dialog-Fenstern	

# Focus-1

Dialog-Fenster, die z.B. beim Refactoren auftreten, werden nicht synchronisiert. Wird also eine Datei umbenannt oder eine Methode ausgelagert, so wirkt dies für andere Sitzungsteilnehmer sehr plötzlich und sprunghaft, da sie den Änderungsprozess nicht sehen - sondern nur das Ergebnis.

Anmerkung: Saros überträgt nur Änderungen innerhalb des Editors.

Quelle: Interviews (Vpn01)

- Feature-Wunsch (Beschwerde)

# Lösungsansätze:

- 1. Saros → MM
  - Übertragung von Dialog-Fenstern.
- 2. MM → Saros

Visualisierung der Einschränkung.

Idee von Waldmann2012:

Farbiger dicker Rahmen um die synchronisiereten Ansichten. Ähnlich zu Screensharing-Programmen, die den übertragenen Teil des Bildschirms einrahmen.

3. MM → Saros

Erklärung der Einschränkung

mit Hinweis auf die ScreenSharing-Funktionalität

10	Awareness	Fehlende Synchronisation	2
	Focus	von Dialog-Fenstern	

Lösungsansatz 1 ist nur im Kombination mit Follow-Modus empfehlenswert, da sonst die Selbstbestimmung des Anwenders eingeschränkt wird und unter Umständen häufige Änderungen und damit einhergehende Dialog-Fenster durch andere Teilnehmer ablenken. Zudem verdecken die Fenster andere eigene Fenster und/ oder Quelltext an dem man gerade arbeitet (S.50, Ellis1991). Gegen die Übertragung von Änderung anderer in Kontext-Menüs oder von Fenstern spricht eine ähnliche Herangehensweise von GoogleDocs. Hier werden ebenfalls nur die Auswirkungen der Änderungen übertragen. Idee Nummer 2 ist grundsätzlich gut, aber ebenso wie 1. zeitlich nicht umsetzbar. Zudem kommt hinzu, dass solange keine weiteren Features in dieser Richtung zu Saros hinzukommen, sich der farbige Rahmen immer nur auf die Editor Ansicht beschränken würde.

Maßnahme 3: Arbeitspaket "Do's and Don'ts" (siehe www.saros-project.org/DosDonts).

11	Awareness	Fehlende Synchronisation von	2
	Focus / EplicitSync	Programmen bei Ausführung	

#### Focus-2

Es wurde der Wunsch geäußert, dass beim Ausführen von Programmen das Programm-Fenster/ GUI und die Konsole übertragen werden.

Anmerkung: Saros überträgt nur Änderungen innerhalb des Editors. Das Starten eines Programms findet nur lokal statt und wird nicht synchronisiert. Das heißt, dies muss von allen Sitzungsteilnehmer manuell durchgeführt werden.

Quelle: Interviews (Vpn01, Vpn03)

- Feature-Wunsch

# Lösungsansätze:

- Saros → MM
   Unterstützung der Synchronisierung z.B. mittels eines "run-shared" Knopfes
- MM → Saros
   Erklärung der Einschränkung
   mit Hinweis auf die ScreenSharing-Funktion

Eine Umsetzung dieser Funktionalität (1.) ist aufwendig und innerhalb der Arbeit nicht durchführbar.

Möglich ist eine übersichtlichere und verständlichere Darstellung des Saros-Synchronisations-Konzepts auf der Seite www.saros-project.org/DosDonts, um die Einschränkung zu erläuterung und einen Hinweis auf die ScreenSharing-Funktion zu geben. Arbeitspaket "Do's and Don'ts".

12	Awareness Focus	Fehlende Synchronisation der Package Explorer Ansicht	2

#### Focus-3

Es wurde der Wunsch geäußert, die Ansicht des Package Explorers mit zu synchronisieren (vor allem in Kombination mit dem Follow-Modus).

Anmerkung: Saros überträgt nur Änderungen innerhalb des Editors.

Quelle: Interviews (Vpn04) + UT[Solovjev2011] - Feature-Wunsch

# Lösungsansätze:

1. Saros → MM

Unterstützung der Synchronisierung,

z.B. sobald der Follow-Modus aktiv ist, zusätzlich die Baumstruktur der geteilten Projekte mit synchronisieren.

2. MM → Saros

Visualisierung der Einschränkung

Idee von Waldmann2012:

Farbiger dicker Rahmen um die synchronisierten Ansichten. Ähnlich zu Screensharing-Programmen, die den übertragenen Teil des Bildschirms einrahmen.

3. MM → Saros

Erklärung der Einschränkung (siehe http://www.saros-project.org/DosDonts).

- Ansatz nur im Kombination mit Follow-Modus empfehlenswert, da dies sonst die Selbstbestimmung des Anwenders einschränkt und unter Umständen häufige Änderungen durch andere Teilnehmer von der eigenen Arbeit ablenken (S. 50, Ellis1991). Die Lösung ist zeitlich nicht umsetzbar.
- 2. Die Idee ist grundsätzlich gut, aber zeitlich nicht umsetzbar. Zudem kommt hinzu, dass solange keine weiteren Features zu Saros hinzukommen, sich der farbige Rahmen immer nur auf die Editor Ansicht beschränken wird. Es sollte also zuvor eindeutig erklärt werden, dass dies der Fall ist.
- 3. Mindestlösung wäre Arbeitspaket "Do's and Don'ts".

13	Comprehension / MM Roles	Unklarheit über die Bedeutung des Hosts	4
13	·	_	4

#### Roles-1

Anwender wissen zwar, wer sich in der Rolle des Hosts befindet, aber nicht welche Besonderheiten damit einhergehen und welche Auswirkungen dies auf die laufende Sitzung hat. Zudem konnte nicht eindeutig zwischen der Host-Rolle und Funktionen des Follow-Modus differenziert werden.

Anmerkung: Wer die Sitzung initiiert hat und damit die Host-Rolle innehält, ist in der Saros Ansicht zu erkennen: Vor dem Namen des entsprechenden Teilnehmers steht das Wort "Host".

Quelle: Interviews + UT[Kahlert2011]+ UT[Waldmann2012]

13 Comprehension / MM Unklarheit über die Bedeutung des Hosts	13		_	4
---	----	--	---	---

## Lösungsansätze:

1. Saros → MM

Aufheben der Host-Sonderrolle, sodass niemand alleinig für eine Sitzung zuständig ist (Ellis1991).

2. Saros → MM

Ermöglichen der Übertragung der Host-Rolle an andere Sitzungsteilnehmer.

- 3. MM → Saros
  - Erläuterung der Notwendigkeit und damit einhergehender Sonderrechte des Hosts (z.B. visuell in Form eines Comics; Webb2012)
  - 2. Umbenennen des Hosts in "Manager" oder "Coordinator" (S. 46, Ellis1991)

Zeitlich und derzeit grundsätzlich sind 1. und 2. nicht umsetzbar, da die Rolle eines zentralen Sitzungsleiters zu tief in Saros verankert ist. Deswegen sollte mindestens Lösung 3 umgesetzt und die Host-Rolle erläutert werden. Diese Information ist zunächst nur für die Person in der Host-Rolle von Bedeutung, sollte aber grundsätzlich jedem zur Verfügung stehen (z.B. auf der Saros-Webseite).

Arbeitspaket "Getting Started" mit OpenFile-2, OpenFile-3/4 und GettingStarted-1.

14	GettingStarted	Anlegen eines XMPP-Kontos	2
		unbequem	

# GettingStarted-1

Das Anlegen eines XMPP-/ Jabber-Kontos wird als unbequem und Hürde angesehen, um Saros zu verwenden.

Anmerkung: Ein XMPP-Konto ist eine Voraussetzung, um andere Anwender zu Saros-Sitzungen einladen zu können. E-Mail-Adressen von Google, GMX, Web.de oder 1&1 könnnen ebenfalls verwendet werden. Darauf wird durch eine Info-Box innerhalb des Saros Configuration Wizard hingewiesen.

Quelle: Interviews (Vpn04, Vpn06) - Feature-Wunsch (Beschwerde)

# Lösungsansätze:

1. Saros → MM

Deutlicher auf die Möglichkeit der Verwendung von bestehenden Konten hinweisen.

- 1. Vereinfachen des Informationstextes.
- 2. Erweitern durch Symbole der E-Mail-Anbieter.
- 3. Dropdown-Menü mit den verschiedenen Adress-Möglichkeiten füllen.
- 2. MM → Saros

Hinweis auf die unterstützte E-Mail-Anbieter

Umsetzung von Ansatz 2: Arbeitspaket "Getting Started".

15	GettingStarted	langwieriger Einladungsprozess	1
An	GettingStarted-2 Anwender äußern den Wunsch einer schnelleren Einladungsform für Sitzungen, z.B. per Link.		
Quelle: Interviews (Vpn04, Vpn06) + Interviews[Solovjev2011] - Feature (Beschwerde)  Lösungsansätze:  1. MM → Saros Änderung durch Waldmann2012 Verbessern des Feedbacks während des Einladungsprozesses  2. Saros → MM Versenden von Einladungen per Link / E-Mail (gesehen bei GoogleDocs, Collabedit)			-Wunsch
	· ·	Feedbacks während des Einladungsprozess siert und warum die Einladung seine Zeit b	

Bei allen folgenden Maßnahmen für die aufgelisteten Usability-Probleme handelt es sich sofern nicht anders benannt - immer um Maßnahmen der Form Saros → MM. Von den insgesamt 21 folgenden Problemen überschneiden sich 2 mit den vorangegangenen. Zu den 15 genannten, kommen also praktisch nur 19 weitere dazu.

16 Coordination sitzungsunabhängiges Koordinieren von Aufgaben nicht möglich
--

#### Coord-Tasks-1

Saros bietet Anwendern keine Möglichkeit sitzungsunabhängig ihre Aufgaben zu planen, mit Prioritäten zu versehen oder diese jemanden zuzuweisen.

Anmerkung: Alle Kommunikationswege von Saros sind von Sitzungen abhängig, weswegen ein sitzungsunabhängiger Austausch nicht möglich ist.

Quelle: Mentales Modell

# Lösungsansätze:

- 1. Unterstützen eines sitzungsunabhängigen Chats
- 2. Unterstützen einer sitzungsunabhängigen VoIP-Verbindung
- 3. Integration von Mylyn

Mylyn ist ein Eclipse-Plugin zur Aufgabenverwaltung und setzt alle in der Problembeschreibung genannten Punkte um. Bezüglich der Kommmunikation zur Koordination von Aufgaben: Die meisten Anwender umgehen die Saros-Kommunikationswege mit Skype, um sowohl vor als auch nach einer Saros Sitzung sprechen und schreiben zu können.

Lösungsansatz 1 wurde bereits von S. Rossbach unabhängig vom Saros-Hauptbranch entwickelt (IM-Feature in Saros Light). Zum Zeitpunkt dieser Arbeit beschäftigt sich ein Team im Rahmen des Softwareprojekts "Agile Softwareentwicklung in einem Open-Source-Projekt" damit diese Funktionalität in den Saros-Master-Branch zu überführen. Die Umsetzung der Lösungsansätze 2-3 ist aus zeitlichen Gründen nicht machbar.

#### UX-1

Anwender haben unterschiedlichste Annahmen bezüglich Paarprogrammierung: Was für Vorteile und Nachteile die Arbeitsweise mit sich bringt, wann es sich lohnt so zusammenzuarbeiten und unter welchen Voraussetzungen dies überhaupt erst zustande kommen sollte.

Quelle: Mentales Modell

Lösungsansätze:

MM → Saros

- 1. Werben für (verteilte) Paarprogrammierung auf der Saros-Webseite
  - Verweisen auf Studien bezüglich der Vorteile von (verteilter) Paarprogrammierung
  - 2. Verweisen auf "erfolgreiche" Firmen, die (verteilte) Paarprogrammierung einsetzen (z.B. InMediasP)

# 17 UX Bilden einer Meinung zur 1 Zusammenarbeit

Vielleicht könnte man durch zusätzliche Informationen z.B. auf der Saros-Webseite bezüglich der Vorteile von (verteilter) Paarprogrammierung Anwender locken, die "zufällig über diese Seite stolpern". Andererseits kommen vielleicht viele erst auf die Saros-Webseite, wenn sie sich sowieso schon für (verteilte) Paarprogrammierung interessieren.

Umsetzung nur wenn zusätzliche Zeit vorhanden ist.

1	18	Coordination Team	mangelnde Unterstützung bei der Teambildung	3
		Tourn	i odinishading	

#### Coord-Team-1

Saros unterstützt größere Teams nur bedingt beim Bilden von kleineren Arbeitsgruppen.

Anmerkung: Alle Kommunikationswege von Saros sind von Sitzungen abhängig, weswegen ein vorheriges Absprechen/ Schreiben nicht möglich ist, um sich innerhalb des Teams zu koordinieren. Man sieht in der Saros View zwar, ob andere Teammitglieder online sind, aber nicht, ob diese bereits in einer Sitzung zusammenarbeiten.

Quelle: Mentales Modell

## Lösungsansätze:

- 1. Unterstützen eines sitzungsunabhängigen Chats
- 2. Unterstützen einer sitzungsunabhängigen VoIP-Verbindung
- 3. Erweitern der Status-Anzeige um die Information, ob man in einer Saros-Sitzung ist

Bei der eigentlich Team-Bildung kann Saros nicht helfen, aber die Kommunikationswege dafür sollten vor und nach der Sitzung gegeben sein, damit sich Entwickler untereinander austauschen und koordinieren können. Auch hier umgehen die meisten Anwender die Saros-Kommunikationswege mit Skype.

Die Umsetzungen Maßnahmen 1 und 3 werden derzeit von S. Rossbach vorangetrieben (IM-Feature und RemoteSessionTracker).

Lösungsansatz 2 ist aus zeitlichen Gründen nicht umsetzbar.

19	Coordination	Wechseln zwischen Teams nur	2
	Team	bedingt möglich	

## Coord-Team-2

Saros bietet nur eingeschränkte Möglichkeiten zwischen verschiedenen Teams zu wechseln.

Anmerkung: Hat man selbst eine Sitzung gestartet (Host-Rolle), dann schließt man beim Verlassen der Sitzung, diese automatisch für alle Teilnehmer.

Quelle: Mentales Modell

## Lösungsansätze:

1. Abschaffen des Hosts

Die Abschaffung der Sonderrolle Host wird bis auf weiteres nicht in Saros möglich sein, da über diesen derzeit alle Aktivitäten laufen und er zuständig für die Konsistenz der Dokumente aller Teilnehmer ist. Zudem handelt es sich hier eine Einfachnennung. Zeitlich nicht umsetzbar.

20	Coordination	Vergeben von PP-Rollen kann nur	2
	Team	simuliert werden	

## Coord-Team-3

Saros bietet keine direkte Möglichkeit typische Rollen der Paarprogrammierung zu vergeben (Driver, Observer).

Anmerkung: Über die Vergabe von Rechten (read/write, read-only) in Kombination mit dem Follow-Modus können die Rollen simuliert werden. (Follow-Modus + read-only = Observer)

Quelle: Mentales Modell

## Lösungsansätze:

- 1. Abschaffen des Hosts
- 2. Unterstützung der Rollen
  - 1. Driver
  - 2. Observer

(Kombinieren von Follow-Modus und read-only)

Die Rollen von Driver und Observer existierten in einer früheren Version von Saros und wurden im Zuge der Erweiterung auf der Mehrbenutzerfähigkeit (>2) abgeschafft. Die Bezeichnung *Driver* und die damit einhergehende Metapher suggerierte, dass es immer nur eine Person in dieser Rolle geben kann. Dieses Konzept führte zu Verwirrung und wurde deshalb durch das Konzept der Lese- und Schreibrechte ersetzt. Um eine wirkliche Unterstützung der PP-Rollen umzusetzen, müsste zunächst die Sonderrolle des Hosts abgeschafft werden. Dieser bisher der einzige, der Rechte vergeben kann. Zudem besitzt er selbst immer Schreibrechte und kann deshalb nie ein reiner Beobachter sein. Zeitlich nicht umsetzbar.

# 21 Coordination Wechseln zwischen PP-Rollen 2 Team

#### Coord-Team-4

Saros bietet keine direkte Möglichkeit zwischen PP-typischen Rollen zu wechseln (Driver, Observer).

Anmerkung: Über die Vergabe von Rechten (read/write, read-only) in Kombination mit dem Follow-Modus können die Rollen simuliert werden. Der Host kann aber seine Schreibrechte nicht ablegen und ist somit nie in der reinen Beobachter-Rolle.

Quelle: Mentales Modell

#### Lösungsansätze:

- 1. Abschaffen des Hosts
- 2. für 2er-Sitzungen
  - Unterstützen des Rollenwechsels, z.B. mittels eines Buttons in der Saros Toolbar
  - 2. Erinnern an Rollen-Wechsel,
    - z.B. mittels eines Timers

Die Abschaffung der Sonderrolle Host wird bis auf weiteres nicht in Saros möglich sein, da dieser auf technischer Ebene derzeit unabdingbar für die Konsistenz der geteilten Dateien ist. Beide Ansätze sind zeitlich nicht umsetzbar.

22	Communication Verbal	verbale Kommunikationswege auf Sitzungen beschränkt	4

## Comm-Verb-1

Saros bietet nur eingeschränkte Möglichkeiten sich über den gesamten Zeitraum der Zusammenarbeit hinweg zu unterhalten.

Anmerkung: Mit Zusammenarbeit sei hier nicht nur das reine Programmieren innerhalb einer Sitzung gemeint, sondern auch das Sprechen vor und nach Sitzungseinheiten.

Quelle: Mentales Modell

#### Lösungsansätze:

- 1. Erweitern der experimentellen Skype-Integration auf Sprach-Verbindungen
- 2. Unterstützen einer sitzungsunabhängigen VoIP-Verbindung

Viele Anwender umgehen das Problem mit Skype. Nichtsdestotrotz wäre eine vollständige Skype-Integration und für Anwender ohne Skype-Konto eine sitzungsunabhängige VoIP-Verbindung für die Zusammenarbeit unabdingbar. Beides ist leider aus Zeitgründen nicht umsetzbar.

23	Communication	Austausch über projektunabhängige Themen	1
----	---------------	---	---

#### Comm-1

Saros bietet keine explizite Möglichkeit sich auch über projektunabhängige Themen auszutauschen (um so beispielsweise die Zusammenarbeit aufzulockern).

Quelle: Mentales Modell

#### Lösungsansätze:

1. Unterstützen von Emoticons im Chat (gesehen bei Skype)

Auch wenn das Whiteboard durch seine Malfunktion solche Zwecke herhalten könnte, wären zusätzlich Funktionen zum Auflockern der gemeinsamen Arbeit wünschenswert. Zeitlich nicht umsetzbar.

24 Communication MangeInde Unterstützung sich gegenseitig zu helf	
---	--

#### Comm-2

Saros bietet kaum Unterstützung, damit sich Entwickler gegenseitig helfen können.

Anmerkung: Es ist nicht möglich vor einer laufenden Sitzung um Hilfe zu fragen oder jemanden anzuschreiben, noch kann man Hinweise oder Tipps zu entsprechenden Code-Stellen an eben dieser Position schriftlich festhalten.

Quelle: Mentales Modell

#### Lösungsansätze:

- 1. Unterstützen eines sitzungsunabhängigen Chats
- 2. Unterstützen einer sitzungsunabhängigen VoIP-Verbindung
- 3. Schreiben von Kommentaren/ Notizen (gesehen bei GoogleDocs, Adobe X Pro)

Um jemanden zu helfen oder Hilfe zu bekommen, sind sitzungsunabhängige Kommunikationswege unabdingbar. Neben der direkten Kommunikation setzen andere Anwendungen zudem Kommentar-Funktionen um.

Für Details zum Lösungsansatz 1 siehe Problembeschreibung 16.

Die Lösungsansätze 2 und 3 sind zeitlich nicht umsetzbar.

25 Coordination mangelhafte Entscheidungshilfe zum Feststellen einer erfolgreichen Zusammenarbeit	3
---	---

#### Coord-Tasks-2

Saros bietet keine Möglichkeit festzuhalten und dann festzustellen, ob eine Zusammenarbeit (erfolgreich) beendet werden kann.

Quelle: Mentales Modell

25	Coordination Tasks	mangelhafte Entscheidungshilfe zum Feststellen einer erfolgreichen Zusammenarbeit	3
----	-----------------------	---	---

#### Lösungsansätze:

- 1. Sperren von Sitzungen (gesehen bei GoogleDocs)
- 2. Integration von Mylyn

Mylyn ist ein Eclipse-Plugin zur Aufgabenverwaltung. Hält man zu Beginn der Zusammenarbeit alle Aufgaben fest, so kann diese abarbeiten. Dies hilft am Ende bei der Entscheidung über das Ende der Zusammenarbeit. Zudem wäre es denkbar im Anschluss daran die laufende Sitzung (oder ein einzelnes Projekt) zu sperren, so dass sichergestellt wird, dass genau dieser Zustand nicht mehr verändert wird. Die Ansätze sind zeitlich nicht umsetzbar.

26	Cooperation	Fehlende Git-Unterstützung	2

#### Coop-1

Saros unterstützt keine Git-Versionsverwaltung.

Anmerkung: Saros unterstützt das Versionsverwaltungssystem SVN. Unter Verwendung von SVN kann der Einladungsprozess zu einer Sitzung beschleunigt werden, da die Daten nicht von Teilnehmer zu Teilnehmer übertragen, sondern mit dem entsprechenden Server abgeglichen werden.

Quelle: Mentales Modell

Lösungsansätze:

1. Unterstützung von Git

Zeitlich nicht umsetzbar.

_			
27	Cooperation	Einchecken nicht während der	1
		Sitzungen möglich	

#### Coop-2

Es ist nicht möglich während einer Sitzung den aktuellen Projektstatus einzuchecken.

Quelle: Mentales Modell

#### Lösungsansätze:

1. Erweitern der SVN-Unterstützung

Das Einchecken (Commit) des aktuellen Projektstandes ist nicht möglich, da es ungelöste Probleme gibt wie: "Wer ist der Author des Commits?" und SVN akzeptiert nur einen Author. Solange es dafür keine Lösungen gibt, bleibt nur das Einchecken nach der Sitzung.

Austauschfunktion	28	Communication	Eingeschränkte Datei- Austauschfunktion	1
-------------------	----	---------------	--	---

#### Comm-3

Das Austauschen von Dokumenten ist nur mit einzelnen Person möglich.

Quelle: Mentales Modell

#### Lösungsansätze:

- 1. Senden von Dateien an mehrere Personen gleichzeitig
- 2. Versenden von Dateien durch Drag&Drop in den Chat (gesehen bei Skype)

Da die meisten Sitzungen nur mit zwei Personen abgehalten werden, ist dies nur ein minimales Problem.

29	Communication	Umständliches Erstellen	2
		von UML-Diagrammen	

#### Comm-4

In Saros ist es nur bedingt möglich UML-Diagramme zu erstellen.

Anmerkung: Es gibt ein sogenanntes Whiteboard mit Malfunktionen, welches sich noch im Entwicklungsstadium befindet und derzeit nur Paint-ähnliche Funktionen unterstützt. Die neue Version beinhaltet Werkzeuge wie "intelligente Verbinder" (Degener2012), ist aber so noch nicht in Saros integriert.

Quelle: Mentales Modell

#### Lösungsansätze:

1. Speichern und Wiederverwenden von Schablonen ("Shapes") (gesehen bei Visio)

Mit der aktuelleren Version des Whiteboards wäre das Erstellen von UML-Diagrammen wesentlich vereinfacht. Zusätzliche Hilfestellung würden wiederverwendbare Schablonen leisten.

Zeitlich nicht umsetzbar.

30	Awareness	Instanziieren von Objekten	1
	Focus		

#### Focus-4

Muss ein Sitzungsteilnehmer beim Instanziieren von Objekten viel rumprobieren, können ihm andere Teilnehmer nur bedingt helfen, weil das Fenster der verwendeten Autovervollständigung von Saros nicht übertragen wird.

Anmerkung: Saros überträgt nur Änderungen innerhalb des Editors.

Quelle: Mentales Modell

#### Lösungsansätze:

1. Übertragen des Autovervollständigungs-Fensters

30	Awareness Focus	Instanziieren von Objekten	1
Zeitlich nicht umsetzbar.			

Ticoncronicien von Losungen		31	Comm	Fehlende Unterstützung beim Recherchieren von Lösungen	1
-----------------------------	--	----	------	---	---

#### Comm-5

Saros bietet Anwendern keine Unterstützung beim Recherchieren von Lösungen.

Quelle: Mentales Modell

#### Lösungsansätze:

1. Unterstützen ein eingebetteten Suchfunktion (gesehen bei GoogleDocs)

Gewonnene Erkenntnisse aus der Recherche anderen mitzuteilen, wäre eine hilfreiche Erweiterung. Das Problem besteht hierbei die Recherche-Ergebnisse von Eclipse-extern nach -intern zu überführen. Zeitlich nicht umsetzbar.

32	Awareness Focus	Fehlende Synchronisation von Tests bei Ausführung	2
----	--------------------	---	---

#### Focus-5

Saros überträgt keine Informationen beim Testen von Programmen, wie zum Beispiel gesetzte Breakpoints oder die Konsole.

Zudem ist ein gemeinsames Testen nur möglich, wenn alle Sitzungsteilnehmer die Schritte durchführen. Das schließt, den Wechsel in die Debug-Perspektive, das Setzen von Breaktpoints und das Starten des Programms im Debug-Modus mit ein.

Anmerkung: Saros überträgt nur Änderungen innerhalb des Editors.

Quelle: Mentales Modell

### Lösungsansätze:

- Unterstützung der Synchronisierung
   z.B. mittels eines "debug-shared" Knopfes
- MM → Saros
   Erklärung der Einschränkung.

Eine Umsetzung dieser Funktionalität (1.) ist aufwendig und innerhalb der Arbeit nicht durchführbar.

Umformulieren der Information auf der Seite http://www.saros-project.org/DosDonts: Erläuterung der Einschränkung und Hinweis auf die ScreenSharing-Funktion. Mögliches Arbeitspaket mit Focus-2.

33 Kooperation Mangelhafte Review-Unterstützung
---

#### Coop-3

Saros unterstützt Anwender nur mangelhaft beim Durchführen von Reviews.

Anmerkung: Es existiert der Follow-Modus mit den man sich gemeinsam durch die Dateien navigieren kann.

Quelle: Mentales Modell

#### Lösungsansätze:

- 1. Schreiben von Kommentaren/ Notizen (gesehen bei GoogleDocs, Adobe X Pro)
- 2. Integration von Mylyn Reviews
- 1. Eine Umsetzung zum Schreiben von Kommentaren, wie bei Review-Boards üblich ist zu aufwendig und zeitlich nicht umsetzbar.
- 2. *Mylyn Reviews* ist Review-Werkzeug für Mylyn, wodurch Reviews als Aufgaben verstanden werden können. Es exisitert zudem eine Integration von Gerrit. Ansatz ist zeitlich nicht umsetzbar.

34	Coordination Team	Mangelhafte Pause-Resume-Funktion für Sitzungen	2
----	----------------------	---	---

#### Coord-Team-5

In Saros ist es nicht möglich eine Sitzung zu pausieren und dann zu einem späteren Zeitpunkt wieder aufzunehmen.

Anmerkung: Dies ist besonders wünschenswert, wenn es sich um besonders große Projekte handelt. Müsste man die Sitzung beenden und zu einem späteren Zeitpunkt wieder neu beginnen, müssen u.U. alle Daten wieder neu übertragen werden.

Quelle: Mentales Modell

- Änderung durch Waldmann2012
   Ein schneller Neustart der Sitzung ist möglich, durch leichtes wiederherstellen der letzten Projektselektion
- Idee von Waldmann2012
   Weitere Optimierungsmöglichkeiten wäre eine "Resume Last Session" Schaltfläche,
   die automatisch Einladungen für die Teilnehmer der letzten Sitzung auslöst.
   Ideal: Automatische Verwendung der zuletzt zugewiesenen Farben

Vor der Änderung durch Waldmann2012 Fatalität 3. Idee 2 ist im Rahmen dieser Arbeit zeitlich nicht umsetzbar.

# Zusammenfassung der Usability-Probleme in Arbeitspakete

# Arbeitspakete

	# Lösung	Erläuterung	Lösungsansätze	Priorität	Aufwand in h
Getting Started					48
Session-5	L3	Doppelpfeil im PE	- Überarbeitung der GettingStarted-Seite,	4	
OpenFile-1	L2	Datei fokusiert	Erklärung der - blauen Doppelpfeile	4	
OpenFile-2	L1	Datei(en) offen	- grünen Punkte	1	10
	L1	Viewport L + R	<ul> <li>gelben Punkte</li> <li>Viewport / nur einen anzeigen</li> <li>Host-Comic</li> </ul>	3	16
OpenFile-3/-4	L2			3	
GettingStarted-1	L2	Anlegen XMPP-Konto	- XMPP-Konto via E-Mail-Anbieter	3	
GettingStarted-1	L1	Anlegen XMPP-Konto	- Hinweis auf unterstützte E-Mail-Anbieter (verändern des Saros Config Wizards)	3	8
Roles-1	L3	MM Host	- Umsetzen des Host-Comics	1	24
Follow Mode					24
Follow-3	L1	Status: ext Ressource		1	24
Follow-4	L2	Status: zusgs. Editor	fokusierte Datei)	1	24
Website					8
OpenFile-7	L2	sync Breakpoints	- verständlichere Erklärung der Saros	3	ļ
Focus-1	L3	sync Refactor- Dialoge	Synchronisations-Einschränkung auf www.saros-project.org/DosDonts	3	
Focus-2	L2	sync Ausführung von Programmen		3	4
Focus-3	L3	sync PE		3	
Focus-5	L2	Ausführen von Tests		3	
UX-1	L1	Meinung zur Zusammenarbeit	- Werben für DPP auf der Saros-Webseiite	3	4
Editor					
Follow-4	L1	Status: zusgs. Editor	- Dummy-Editor	2	
	L3	Otatus. 2039s. Editor	- Unterstützung des Datei-Formats	2	
Comm-2	L3	Sich ggseitig helfen	- Schreiben von Kommentaren/ Notizen	4	
Coop-3	L1	Durchführen von Reviews		4	
VoIP					
Coord-Tasks-1	L2	Koordinieren von Aufgaben	- Abkapseln der VoIP-Funktionalität	2	
Coord-Team-1	L2	Koordinieren des Teams		4	
Comm-Verb-1	L2	Telefonieren		2	
Comm-2	L2	Sich ggseitig helfen		2	
Skype		_			
Comm-Verb-1	L1	Telefonieren	- Sprach-Erweiterung der Skype-Integration	2	
Chat					SWP-Team
Coord-Tasks-1	L1	Koordinieren von Aufgaben	- Abkapseln der Chat-Funktionalität (siehe IM-Feature von S. Rossbach in Saros Light)	2	^
Coord-Team-1	L1	Koordinieren des Teams		4	^
Comm-2	L1	Sich ggseitig helfen		2	^
Comm-1	L1	Austauschen von projektunabhängigen Themen	- Emoticons	4	-

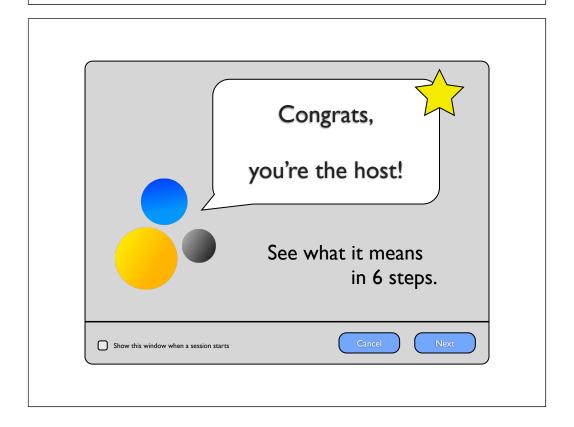
38			1	1		A Intervie
Roster			Vaardiniaran daa	Chatus Appains in Cares Cityunguis/pain		
	Coord-Team-1	L3	Koordinieren des Teams	- Status-Anzeige in Saros-Sitzung: ja/ nein (Prototyp von S. Rossbach in Saros Light)	4	
Host						
	Roles-1	L1	MM Host Koordinieren des	- Abschaffen des Hosts	4	
	Coord-Team-2	L1	Teams	]	4	
	Coord-Team-3	L1	Vergeben von PP- Rollen		4	
	Coord-Team-4	L1	Wechseln zwischen		4	
Roles	Ooord-Team-4		PP-Rollen			
Holes	Roles-1	L2	MM Host	- Übertragen der Host-Rolle	4	
	Coord-Team-3	L2	Vergeben von PP-	- Unterstützung der Rollen Driver, Observer	3	
			Rollen Wechseln zwischen	- Rollenwechsel (Button + Erinnerung)		
	Coord-Team-4	L2	PP-Rollen	3,	3	
SessionIn			langwieriger	Finladung par Link / F Mail		
	GettingStarted-2	L2	Einladungsprozess	- Einladung per Link / E-Mail	4	
Session			1=			
	Coord-Tasks-2	L1	Entscheiden über Ende d. Z.arbeit	- Sperren von Sitzungen	4	
	Coord-Team-5	L2	Regellmäßiges	- Resume Last Session Button (siehe Gerrit	4	
Mylyn			Pausieren	#60 - Prototyp von A. Waldmann)		
	Coord-Tasks-1	L3	Koordinieren von	- Mylyn-Integration (inkl. Mylyn Reviews)	2	
	Coord-Tasks-2		Aufgaben Entscheiden über	-		
	00014-145K5-2	L2	Ende d. Z.arbeit	]	4	
	Coop-3	L2	Durchführen von Reviews		4	
Git						
	Coop-1	L1	Versionieren mit Git	- Git-Unterstützung	4	
SVN			Cin also a long cons	CVAL Hadavadiitavaa		
	Coop-2	L1	Einchecken von Projekten	- SVN-Unterstützung	4	
SendFile						48
	Comm-3	L1 L2	Austauschen von Dokumenten	Senden an mehrere Personen gleichzeitig     Senden per Drag & Drop in den Chat	3	16
Whiteboa	rd	LE		- Senden per brag & brop in den chat		32
	Comm-4	L1	Erstellen von UML-	- wiederverwendbare Visio-Shapes	4	
Research			Diagrammen	(in Progress von H. Degener)		
nesearch	Comm-5	L1	Recherchieren	- eingebettete Suchfunktion (> GoogleDocs)	4	
Color						SWP-Team
	Session-6	L1	Behalten der Farbe	- in Progress (siehe Gerrit #152 S. Rossbach) - Beschränken der Farbwahl auf	2	
			Denaiten der Farbe	Standardfarben		
SarosViev			le u			24
	OpenFile-6	L1 L2	Farbkontrast Farbeskalierung	- Farbunterschied hervorheben	3	8
	NEU		Saros-View	- dezentere Farben	-	16
Curan	NEO			- andere Darstellung der Farbkacheln		10
Cursor	OpenFile-5	L3	Anzeige Cursor	- Cursor hervorheben	4	
SyncPE		25	7 th 201g0 0 di 001		_	
	Session-5	L2	Doppelpfeil im PE	- Erweiterung: Teilen der PE-Ansicht	4	
	Focus-3	L1	sync PE	- sync des Project Explorers	4	
SyncDeb	Ug OpenFile-6	L1	sync Breakpoints	- Unterstützung der Synchronisierung	4	
	Focus-5	L1	Ausführen von Tests	- sync Breakpoints	4	
SyncProg	ram					
	Focus-2	L1	sync Ausführung von Programmen	- sync bei Programmausführung	4	
SyncWind	lows		Togrammen			
	Focus-1	L1	sync Refactor-	- sync Dialog-Fenster in Kombi mit Follow	4	
	Focus-4		Dialoge Instanziieren von	- Synchronisieren des		
		L1	Objekten	Autovervollständigungsfensters	4	
Highlight	Sync		auma Dafa - t - ::	forbiggy Dobroom upp		
	Focus-1	L2	sync Refactor- Dialoge	- farbiger Rahmen um synced Views	4	
	Focus-3	L2	sync PE		4	

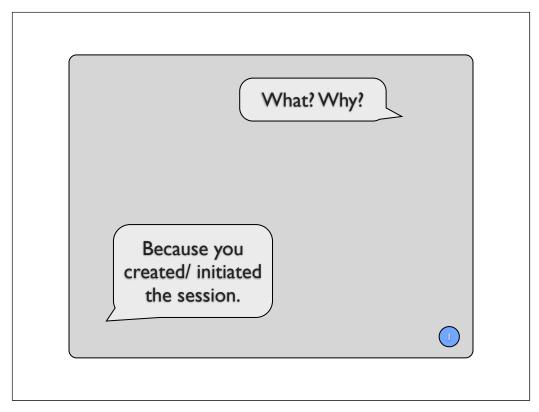
B.1 Arbeitspaket "Getting Started"

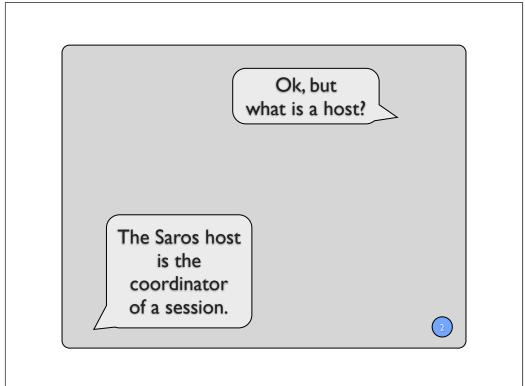
# Das Storyboard zum Comic

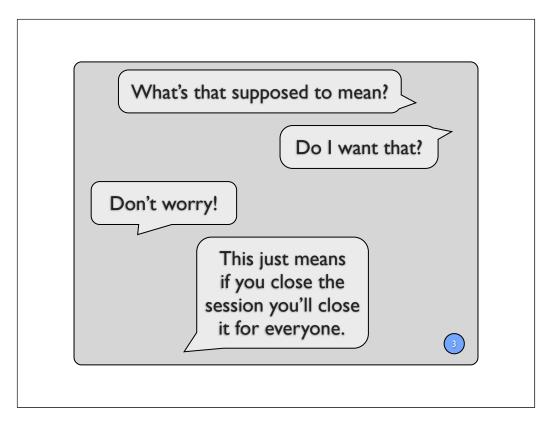
# You're The Host

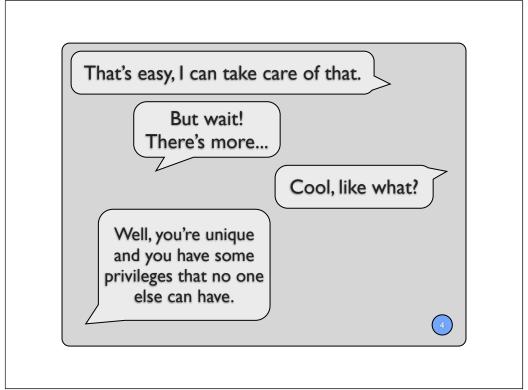
Storyboard

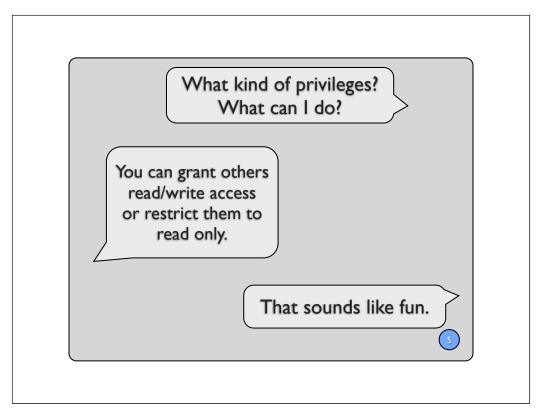


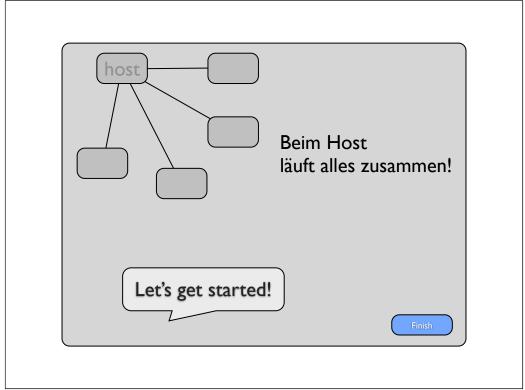










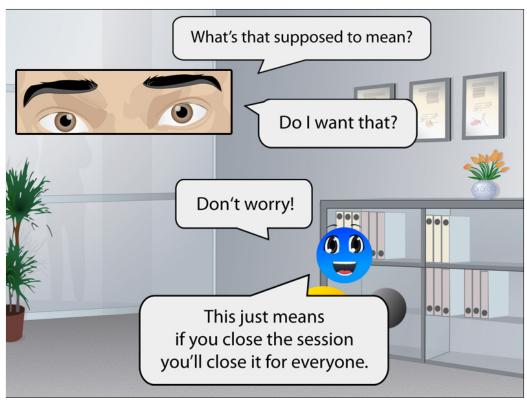


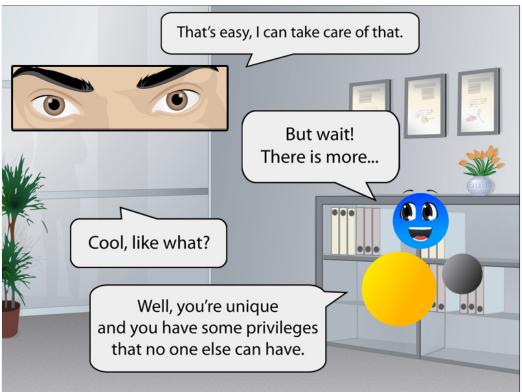
# Der Comic



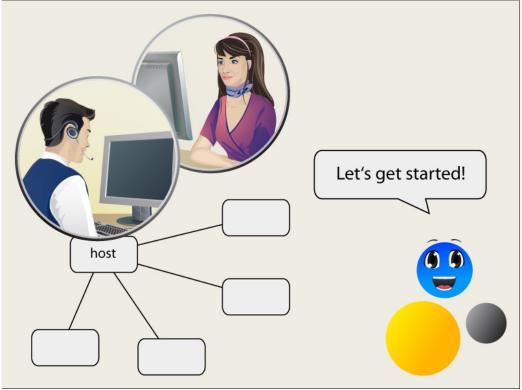












#### B.1.3 Online-Fragebogen zum Host-Comic

#### Das Anschreiben

An: i-studi@mi.fu-berlin.de

Betreff: Umfrage zum SAROS Comic

Liebe InformatikerInnen,

im Rahmen der Masterarbeit zur "Verbesserung der Usability von Sarosïst ein Comic entstanden. Hierzu führe ich eine Umfrage durch und würde mich sehr freuen, wenn Ihr Euch 10 Minuten dafür Zeit nehmen könntet. Die Umfrage besteht aus 7 Bildern und 7 Multiple-Choice-Fragen.

Was ist Saros?

Saros ist ein Open-Source-Plugin für die Entwicklungsumgebung Eclipse zur gemeinsamen und gleichzeitigen Programmierung von verteilten Beteiligten. Diese Form der Zusammenarbeit - unter Verwendung von Eclipse und Saros - findet innerhalb sogenannter Sitzungen statt. In jeder Sitzung existiert eine Person, die die Rolle des "Hostsinnehält. Um diese Rolle wird es in dem Comic und der anschließenden Umfrage gehen.

Schritt 1: Der Comic

https://dl.dropbox.com/u/1259389/SAROS-Host-Comic.pdf

Schritt 2: Die Umfrage

https://docs.google.com/spreadsheet/viewform?formkey=dER1eFpDcnlUNjZWVTJ4YTMOROtmbWc6MQ

Die Umfrage ist bis einschließlich 06.10.2011 freigeschaltet.

Die Teilnahme und Auswertung erfolgt selbstverständlich anonym.

Vielen Dank für die Teilnahme!

Mit freundlichen Grüßen, Maria Spiering

#### Der Fragebogen

#### Situation (1)

Vorbedingung:

- Alice, Bob und Carl wollen zusammenarbeiten.
- Alle haben das Java-Projekt, an dem sie gemeinsam arbeiten wollen, lokal auf ihrem Rechner.
  - 1. Bob wird von Alice zu einer Sitzung eingeladen. Bob lädt Carl in die Sitzung mit ein. Wer ist der Host?
    - (a) Alice
    - (b) Bob
    - (c) Carl
    - (d) Alice und Bob

- 2. Alice und Bob wollen beide, dass Carl über die möglichen Änderungen am Java-Projekt bestimmt. Wer sollte am Besten die Sitzung starten?
  - (a) Alice
  - (b) Bob
  - (c) Carl
  - (d) egal

# Situation (2)

Vorbedingung:

- laufende Sitzung mit den Teilnehmern: Alice, Bob und Carl
- Carl ist Host der Sitzung
  - 3. Was passiert, wenn Alice die Sitzung verlässt?
    - (a) Alice und Carl verlassen die Sitzung
    - (b) die Sitzung wird für alle beendet
    - (c) die Sitzung läuft mit Bob und Carl weiter
  - 4. Was passiert, wenn Carl die Sitzung verlässt?
    - (a) Carl verlässt die Sitzung und kann entscheiden, ob die Sitzung ohne ihn weiter laufen soll
    - (b) die Sitzung wird für alle beendet
    - (c) die Sitzung läuft mit Alice und Bob weiter
  - 5. Alice benutzt den Follow-Modus um Carl zu folgen. Wer ist der Host? Hilfetext: Mit dem Follow-Modus ist es möglich Änderungen und Bewegungen einer Person fortlaufend zu beobachten. Öffnet die Person z.B. eine andere Datei, so wird diese auch bei allen "Followernäutomatisch geöffnet.
    - (a) Alice
    - (b) Bob
    - (c) Carl
    - (d) Alice und Carl
  - 6. Wie oft haben Sie Saros bisher verwendet?
    - (a) gar nicht
    - (b) selten
    - (c) oft
    - (d) regelmäßig
  - 7. Haben Sie schon einmal etwas für Saros entwickelt?

    Hilfetext: z.B. im Rahmen eines Softwareprojektes oder einer Abschlussarbeit

- (a) ja
- (b) nein

# B.2 Arbeitspaket "Saros-View"

### **B.2.1** Berechnung des Farbkontrasts

Für die Berechnung des Farb- und Helligkeitsunterschieds wurde der *Contrast Analyser* der Firma the Paciello Group verwendet<sup>17</sup>.

Bei der Helligkeit sind maximal 255 Einheiten möglich (bei Auswahl der Farben schwarz und weiß) und einem Minimum an 125 durch den Algorithmus gefordert. Bei der Farbe sind maximal 765 Einheiten möglich und 500 gefordert.

Bei der alten Darstellung beläuft sich Helligkeitsunterschied zwischen den zwei Farben auf 20 Punkte und in der neuen Darstellung auf 30. Dieser Unterschied ist nur minimal. Aber der Farb- und Helligkeitsunterschied zum geschriebenen schwarzen Text ist in der neuen Version wesentlich stärker ausgeprägt.

```
Start.java
                                                                                       Bar.java

☑ Start.java

                         J Bar.java
   A.java
                                                                A.java
       package test;
                                                                    package test;
                                                             H
H
       public class FooBar {
                                                                    public class FooBar {
                                                                        public static void main(String[]
                                                                 10
11
12
```

- (a) Alte Darstellung mit einem Helligkeitsunterschied von 20 Einheiten
- (b) Neue Darstellung mit Helligkeitsunterschied von 30 Einheiten

Verbesserung des Farbkontrasts

 $<sup>^{17} \</sup>verb|http://www.paciellogroup.com/resources/contrast-analyser.html|$ 

# Literatur

[1] I. Young, Mental Model: Aligning Design Strategy with Human Behavior, 1st ed. Rosenfeld Media, February 2008.

- [2] L. Prechelt and K. Beecher, "Four generic issues for tools-as-plugins illustrated by the distributed editor saros," in *Proceedings of the 1st Workshop on Developing Tools as Plug-ins*, ser. TOPI '11. New York, NY, USA: ACM, 2011, pp. 9–11. [Online]. Available: http://doi.acm.org/10.1145/1984708.1984712
- [3] L. Prechelt, "Some non-usage data for a distributed editor: the saros outreach," in Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering, ser. CHASE '11. New York, NY, USA: ACM, 2011, pp. 48–48. [Online]. Available: http://doi.acm.org/10.1145/1984642.1984651
- [4] R. Djemili, "Entwicklung einer Eclipse-Erweiterung zur Realisierung und Protokollierung verteilter Paarprogrammierung," Master's thesis, Freie Universität Berlin, August 2006.
- [5] S. Salinger, C. Oezbek, K. Beecher, and J. Schenk, "Saros: an eclipse plug-in for distributed party programming," in *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, ser. CHASE '10. New York, NY, USA: ACM, 2010, pp. 48–55. [Online]. Available: http://doi.acm.org/10.1145/1833310.1833319
- [6] D. E. Kieras and S. Bovair, "The role of a mental model in learning to operate a device," *Cognitive Science*, vol. 8, no. 3, pp. 255 273, 1984.
- [7] C. L. Borgman, "The user's mental model of an information retrieval system: an experiment on a prototype online catalog," *International Journal of Man-Machine Studies*, vol. 24, no. 1, pp. 47 64, 1986.
- [8] N. Staggers and A. Norcio, "Mental models: concepts for human-computer interaction research," *International Journal of Man-Machine Studies*, vol. 38, no. 4, pp. 587 605, 1993.
- [9] B. Kahlert, "Verbesserung der Out-Of-Box-Experience in Saros mittels Heuristischer Evaluation und Usability-Tests," Master's thesis, Freie Universität Berlin, 2011.
- [10] A. Narweleit, "Analyse und Evaluation von Verbesserungsmöglichkeiten der User Experience in Saros," 2011.
- [11] A. Solovjev, "Evaluation der Mechanismen zum Darstellen der Workspace Awareness in Saros," 2011.
- [12] A. Waldmann, "Prüfung und Verbesserung der Usability von Saros im produktiven Einsatz," Master's thesis, Freie Universität Berlin, 2012.
- [13] "About the Eclipse Foundation," 2012. [Online]. Available: http://www.eclipse.org/org/
- [14] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: some issues and experiences," Commun.~ACM,~vol.~34,~no.~1,~pp.~39–58,~1991. [Online]. Available: http://doi.acm.org/10.1145/99977.99987

[15] D. Gallardo, E. Burnette, and R. McGovern, *Eclipse in Action: A Guide for Java Developers*, ser. In Action Series. Manning, 2003.

- [16] N. Edgar, K. Haaland, J. Li, and K. Peter, "User Interface Guidelines," November 2007. [Online]. Available: http://wiki.eclipse.org/User\_Interface\_Guidelines
- [17] T. Groß, M. Koch, and M. Herczeg, *Computer-Supported Cooperative Work*, ser. Interaktive Medien. Oldenbourg, 2007.
- [18] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed. Pearson/Prentice-Hall, 2004.
- [19] S. Teufel, Computerunterstützung für die Gruppenarbeit. Addison-Wesley, 1995.
- [20] P. Dourish and V. Bellotti, "Awareness and coordination in shared workspaces," in Proceedings of the 1992 ACM conference on Computer-supported cooperative work, ser. CSCW '92. New York, NY, USA: ACM, 1992, pp. 107–114. [Online]. Available: http://doi.acm.org/10.1145/143457.143468
- [21] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, vol. 37, no. 1, pp. 32–64, 1995.
- [22] C. Gutwin, "Workspace awareness in real-time distributed groupware," Ph.D. dissertation, Calgary, Alberta, 1997.
- [23] "Offene Abschlussarbeiten im Softwareprojekt Saros." [Online]. Available: http://www.mi.fu-berlin.de/w/SE/ThesesDPP
- [24] R. Klimoski and S. Mohammed, "Team mental model: construct or metaphor?" *Journal of Management*, vol. 20, no. 2, pp. 403 437, 1994, ¡ce:title¿A Special Issue of The Journal of Management¡/ce:title¿.
- [25] A. Cooper, The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity (2nd Edition). Pearson Higher Education, 2004.
- [26] K. S. Moser, "Mentale Modelle und ihre Bedeutung Kognitionspsychologische Grundlagen des Missverstehens," in Sinnbildlich schief: Missgeschicke bei Symbolgenese und Symbolgebrauch, U. Ganz-Blättler, Ed. Peter Lang, 2003, pp. 181–205. [Online]. Available: http://symbolforschung.ch/node/49
- [27] S. Schütze, R. Streule, and D. Läge, "Warum klassische Evaluation oftmals nicht ausreicht eine Studie zur Ermittlung der Bedeutsamkeit Mentaler Modelle als Evaluationsmethode," in Wissensgemeinschaften. Digitale Medien Öffnung und Offenheit in Forschung und Lehre, T. Köhler and N. Neumann, Eds. Münster: Waxmann Verlag, 2011, pp. 273–283.
- [28] K. Potosnak, "Mental models: helping users understand software," IEEE Software, vol. 6, no. 5, 1989.
- [29] D. A. Norman, *Some Observations on Mental Models*. Lawrence Erlbaum Associates, 1983, ch. 1, p. 348.
- [30] N. M. Steiger and D. M. Steiger, "Knowledge management in decision making: Instance-based cognitive mapping," in 40th Hawaii International International Conference on

- Systems Science (HICSS-40 2007), CD-ROM / Abstracts Proceedings, 3-6 January 2007, Waikoloa, Big Island, HI, USA. IEEE Computer Society, 2007, p. 54.
- [31] N. Bevan, "International standards for hei and usability," *International Journal Human-Computer Studies*, vol. 55, pp. 533–552, October 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=565961.565970
- [32] T. Jokela, N. Iivari, J. Matero, and M. Karukka, "The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11," in *Proceedings of the Latin American conference on Human-computer interaction*, ser. CLIHC '03. New York, NY, USA: ACM, 2003, pp. 53–60. [Online]. Available: http://dl.acm.org/citation.cfm?id=944519.944525
- [33] M. J. Davidson, L. Dove, and J. Weltz, "Mental models and usability," November 1999. [Online]. Available: http://www.lauradove.info/reports/mental%20models.htm
- [34] J. Nielsen, Usability Engineering, ser. Morgan Kaufmann Series in Interactive Technologies. AP Professional, 1994.
- [35] C.-M. Karat, R. Campbell, and T. Fiegel, "Comparison of empirical testing and walkthrough methods in user interface evaluation," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '92. New York, NY, USA: ACM, 1992, pp. 397–404. [Online]. Available: http://doi.acm.org/10.1145/142750.142873
- [36] J. Nielsen, "Mental models." [Online]. Available: http://www.useit.com/alertbox/mental-models.html
- [37] R. Menzies, A. Waller, and H. Pain, "Peer interviews: an adapted methodology for contextual understanding in user-centred design," in *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ser. ASSETS '11. New York, NY, USA: ACM, 2011, pp. 273–274. [Online]. Available: http://doi.acm.org/10.1145/2049536.2049603
- [38] J.-Y. Mao, K. Vredenburg, P. W. Smith, and T. Carey, "The state of user-centered design practice," *Communications Of The ACM*, vol. 48, no. 3, pp. 105–109, March 2005.
- [39] M. Maguire, "Methods to support human-centred design," *International Journal Human-Computer Studies*, vol. 55, no. 4, pp. 587–634, 2001.
- [40] N. Bevan and I. Curson, "Planning and implementing user-centred design," in CHI '99 extended abstracts on Human factors in computing systems, ser. CHI EA '99. New York, NY, USA: ACM, 1999, pp. 137–138. [Online]. Available: http://doi.acm.org/10.1145/632716.632800
- [41] Usability Professionals' Association, "What is UCD?" [Online]. Available: http://www.usabilityprofessionals.org/usability\_resources/about\_usability/what\_is\_ucd.html
- [42] L. Prechelt, "Softwaretechnik," 2008, Skript zur Vorlesung.
- [43] A. Brooks and L. Scott, "A methodology from software engineering inspection which supports replicable mental models research," in *Proceedings of the 6th IEEE International Conference on Cognitive Informatics*, ser. COGINF '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 126–133.

[44] N. M. Steiger and D. M. Steiger, "Discovering a decision maker's mental model with instance-based cognitive mining: A theoretical justification and implementation," *Inter*disciplinary Journal of Information, Knowledge, and Management, vol. 4, 2009.

- [45] "Hochschuldidaktik von A-Z der Universität Zürich," 2010. [Online]. Available: http://www.hochschuldidaktik.uzh.ch/hochschuldidaktikaz.html#S
- [46] E. Rosen, "Vorgehen bei der Strukturlegetechnik (SLT)," June 2012, E-Mail vom 01.06.2012.
- [47] J. Rode, M. B. Rosson, and M. A. Pérez-Quiñones, "End-users' mental models of concepts critical to web application development," in *Proceedings of the 2004 IEEE Symposium on Visual Languages Human Centric Computing*, ser. VLHCC '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 215–222. [Online]. Available: http://dx.doi.org/10.1109/VLHCC.2004.25
- [48] C. Bravo-Lillo, L. F. Cranor, J. Downs, and S. Komanduri, "Bridging the gap in computer security warnings: A mental model approach," *IEEE Security and Privacy*, vol. 9, no. 2, pp. 18–26, 2011. [Online]. Available: http://dx.doi.org/10.1109/MSP.2010.198
- [49] C. Kothari, Research Methodology: Methods and Techniques. New Age International Limited, 2009.
- [50] A. Cooper, R. Reimann, and D. Cronin, About Face 3: The Essentials of Interaction Design. John Wiley & Sons, 2007.
- [51] P. M. Institute, A Guide to the Project Management Body of Knowledge: Third Edition, ser. PMI global standard. Project Management Institute, 2004.
- [52] E. Tulving, "How many memory systems are there?" *The American Psychologist*, vol. 40, no. 4, pp. 385–398, 1985.
- [53] H. F. International, "User-centered analysis and conceptual design," 2010, a 3-day course. [Online]. Available: www.humanfactors.com
- [54] R. Brugger, Lean Time Management: Die ultimative Zeitrettung für Gipfelstürmer, Dauerbrenner und andere Ruhelose, 1st ed. Springer, 2011.
- [55] E. N. Webb, G. Balasubramanian, U. OBroin, and J. M. Webb, "WHAM! POW! Comics as User Assistance," *Journal of Usability Studies*, vol. 7, no. 3, p. 13, 2012.
- [56] "Maßnahmenplan Gleichstellung zur Frauenförderung, und Familienfreundlichkeit im Rahmen der Zielvereinbarungen 2010/2011," 2012. [Onlinel. Available: http://www.mi.fu-berlin.de/fb/beauf-aussch/beauf-frauen/ressourcen/ Massnahmenplan\_FBMatheInfo-final.pdf
- [57] S. Kretzschmann, "Verbesserung und Überarbeitung der Saros Homepage unter Verwendung eines User-Centered Design Ansatzes," 2012.
- [58] D. Benyon, P. Turner, and S. Turner, Designing Interactive Systems: People, Activities, Contexts, Technologies. Addison-Wesley, 2005.