

Annotieren von Bildschirmfotos zur Steigerung der Anwenderbeteiligung

Asarnusch Rashid, Clemens van Dinther

FZI Forschungszentrum Informatik

Zusammenfassung

Obwohl zahlreiche Studien belegen, dass die Beteiligung von Anwendern in Softwareprojekten von unschätzbarem Wert ist, wird diese bislang selten praktiziert. Die Gründe dafür liegen unter anderem in dem erheblichen Zeitaufwand für die Durchführung von Anwenderbefragungen. Ferner stellt sich im Dialog mit Anwendern die Herausforderung, sie zur Formulierung von konkreten Anforderungen und Vorschlägen zu befähigen, um nicht ausschließlich Problemlberichte und vage Vorstellungen erfassen zu müssen. Mit der Entwicklung des Werkzeuges OpenProposal soll dieser Problemstellung Rechnung getragen werden. OpenProposal ist ein Programm, mit dem Anwender ihre Kommentare komfortabel als Annotationen auf die Benutzeroberfläche eines Softwareproduktes platzieren und als Bildschirmfoto an die Entwickler senden können. In zwei Fallstudien konnte gezeigt werden, dass sich dieser Ansatz als hilfreich erweist und bei den teilnehmenden Unternehmen erfolgreich angewendet wurde.

1 Einleitung

Zahlreiche Studien (Kujala 2008) belegen, dass die Einbeziehung von Anwendern generell einen positiven Effekt auf den Erfolg eines Systems besitzen kann. Darüber hinaus finden sich einige Hinweise darauf, dass es sich potentiell als eine effektive Methode der Identifizierung von Bedürfnissen erweist, Anwender nicht nur als Informationsquelle sondern auch als aktive Gestalter am Entwicklungsprozess teilnehmen zu lassen (Keil & Carmel 1995).

Etablierte Methoden, die in der Praxis angewendet werden, umfassen u.a. User Experience, Usability Workshops, User Support (bzw. betriebliches Vorschlagswesen) und Benutzerumfragen. Eine für praktische Belange bedeutende Schwachstelle dieser Methoden liegt im hohen Aufwand für Vor- und Nachbereitung. An dieser Stelle soll mit dem Werkzeug OpenProposal ein Beitrag zur Förderung einer frühzeitigeren und häufiger stattfindenden Einbindung von Anwendern in Softwareprojekten geleistet werden. Grundidee hierbei ist es, Anwendern die Möglichkeit zu geben, Verbesserungsvorschläge für eine Software direkt auf dem Bildschirm einer laufenden Softwareanwendung zu annotieren und diese als annotiertes

Bildschirmfoto an den Entwickler zu kommunizieren. Drei Hypothesen bilden den Grundrahmen der OpenProposal Entwicklung:

- H1. Mit OpenProposal werden mehr Verbesserungsvorschläge vom Anwender abgegeben.
- H2. OpenProposal fördert die Kreativität der Anwender und führt zu konkreten Verbesserungsvorschlägen.
- H3. OpenProposal kann auch im Betriebsalltag zur Meldung von Verbesserungsvorschlägen genutzt werden und erleichtert Anwendern die Feedbackabgabe.

2 OpenProposal

Die Entwicklung von OpenProposal fußt auf aktuellen Entwicklungen in den Forschungsfeldern Requirements Engineering (Use Cases, Story Cards, Prototyping, Mock-ups, etc.), Participatory Design (u.a. Distributed Participatory Design (Danielsson et al. 2008), PD in use (Stevens & Draxler 2006)) und Digital Annotation (Fogli et al. 2004). Unmittelbar verwandte Arbeiten finden sich bei Naghsh (2005) mit „Gabbah“ zur Kommentierung von elektronischen Prototypen mit grafischen Annotationen), bei Dörner et al. (2008) mit “Infrastructure Probes” als ethnografische Methode und bei Lohmann et al. (2008) mit “Softfox” zur Kommentierung von Webseiten. Darüber hinaus existiert eine Vielzahl an Werkzeugen zur Erstellung und Bearbeitung von Bildschirmfotos, den so genannten „Snapshot“-Werkzeugen (z.B. Techsmith JING, Annotater Pro).

Basierend auf den Erfahrungsberichten aus verwandten Arbeiten und im Dialog mit Anwendern und Softwareunternehmen wurde das Design von OpenProposal vorgenommen. Folgende Gestaltungsziele bzw. Anforderungen wurden dem Design zugrunde gelegt:

- A1. *Vereinfachung des Einreichungsprozesses.* Das Erstellen und Einreichen von Vorschlägen sollte in wenigen Schritten erfolgen können und so einfach wie möglich sein. Dadurch soll die Akzeptanz der Anwender sichergestellt werden.
- A2. *Formale Strukturen zur besseren Verständlichkeit.* Da Entwickler klare und nachvollziehbare Erklärungen vorziehen, soll Anwendern eine formale Struktur zur Formulierung ihrer Vorschläge angeboten werden. Dies könnte zu leichter verständlichen und konkreteren Vorschlägen führen und Kommunikationskosten senken.
- A3. *Verwaltung der Anwendervorschläge.* Abgegebene Anmerkungen sollten in einer kollaborativen Umgebung gespeichert werden, die allen Teilnehmern Zugang zu den Vorschlägen ermöglicht. Da die Vorschläge der Anwender mitunter nachträglich modifiziert werden müssen (z.B. bei unvollständigen Informationen, unklaren Vorschlägen, etc.) sollte die Möglichkeit gegeben sein Vorschläge auch nach der Abgabe zu bearbeiten.
- A4. *Verringerung von Zeit und Aufwand für die Vor- und Nachbereitung.* Der Zeit- und Arbeitsaufwand für die Vor- und Nachbereitung von Anwenderbefragungen soll verringert werden. So sollen Anwenderbefragungen häufiger durchgeführt werden können.

Dabei sind neben den Rollen „Anwender“ und „Entwickler“ zusätzlich noch die Rollen „Moderator“ und „Entscheider“ zu berücksichtigen. Der Moderator ist verantwortlich dafür, die Bedürfnisse und Erwartungen der Anwender festzuhalten und die „Sprache der Anwender“ in die „Sprache der Entwickler“ zu übersetzen. Die Aufgabe des Entscheiders ist es die Sinnhaftigkeit und Wirtschaftlichkeit der Vorschläge zu bewerten. Er entscheidet, was umgesetzt wird und was nicht.

Abbildung 1 erläutert die Funktionsweise von OpenProposal anhand eines Beispiels: Man stelle sich einen Anwender vor, der einen elektronischen Taschenrechner benutzt. Der Anwender hat mehrere Verbesserungsvorschläge und startet OpenProposal. Zunächst wird der gesamte Anwendungsbildschirm „eingefroren“. Mit (2.B) hat der Anwender die Möglichkeit OpenProposal jederzeit zu deaktivieren, und den Bildschirm wieder freizugeben um auf das Arbeitssystem zugreifen zu können. Um eine Annotation zu setzen, wählt der Anwender ein Annotationswerkzeug (2.C), platziert das Element auf dem eingefrorenen Bildschirm und kann optional einen ergänzenden Text eingeben. (2.1)

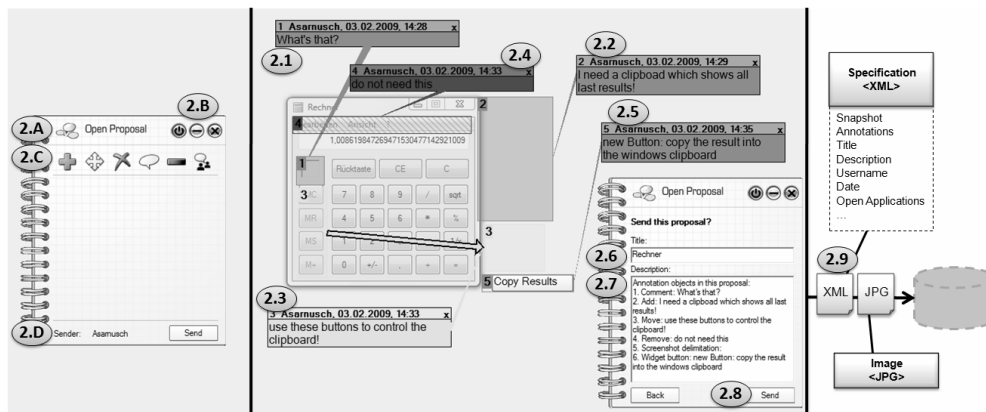


Abbildung 1: Funktionsweise von OpenProposal am Beispiel

Das „Hinzufügen“-Werkzeug (2.2, 2.5) erlaubt es dem Anwender eine Position zu definieren, an der er ein neues Objekt auf dem Bildschirm wünscht. Das „Entfernen“-Werkzeug (2.4) erfüllt die gegenteilige Funktion; ein bereits existierendes Element wird als überflüssig gekennzeichnet. Wird das „Verschieben“-Werkzeug (2.3) verwendet, wählt der Anwender zunächst einen Bereich aus, der an eine andere Position auf der Arbeitsfläche verschoben werden soll, und bestimmt dann die Zielposition. Das Werkzeug „Kommentieren“ (2.1) kann zur Beschreibung von Vorschlägen herangezogen werden, wenn keines der anderen Elemente passt oder man nicht lange nach dem richtigen Element suchen möchte. Zum Schutz der Privatsphäre kann der Anwender mithilfe des „Einschränken“-Werkzeuges das Bildschirmfoto auf einen ausgewählten Bereich des Bildschirms eingrenzen bzw. bestimmte Bereiche schwärzen. Das „Diskutieren“-Modul umfasst mehrere Werkzeuge mit dem beispielsweise Fragen, Fehlermeldungen, Zustimmung, etc. vermerkt werden können. Alle Annotationen

werden als Objekte dargestellt, die bearbeitet, verschoben oder gelöscht werden können, wann immer der Anwender dies wünscht.

Sobald die Annotation beendet ist, kann der Vorschlag durch einen Titel (2.6) und eine generelle Beschreibung (2.7) ergänzt und schließlich durch Anklicken des "Senden"-Buttons (2.8) versendet werden. Abhängig von der vorherigen Einstellung wird die Anfrage entweder (2.9) an einen Issue-Tracker gesendet, auf der Festplatte gespeichert oder als Email an eine spezielle Adresse versendet. OpenProposal erstellt ein XML-Dokument, das die Annotationen, das Bildschirmfoto sowie Metadaten zum Vorschlag und zum Arbeitssystem des Anwenders enthält (z.B. Datum, Titel der annotierten Anwendung, Username).

Von den typischen „Snapshot“-Werkzeugen unterscheidet sich OpenProposal darin, dass direkt auf der Anwendungsoberfläche annotiert wird, dem Anwender strukturierte Annotationswerkzeuge angeboten werden und die Übermittlung seiner annotierten Bildschirmfotos per Knopfdruck ermöglicht wird. Mit diesem Ansatz soll gezeigt werden, dass die Anwenderbeteiligung mit einem auf die Belange der Anwender zugeschnittenen Werkzeug verbessert werden kann.

3 Evaluation

Im Laufe der OpenProposal Entwicklung wurden im Rahmen realer Softwareprojekten zwei Fallstudien durchgeführt. Beide Studien folgten derselben Struktur:

1. Anwender testeten im Rahmen eines realen Softwareprojektes einen Prototyp, entweder im Rahmen eines Usability Workshops oder einer Interview-Reihe.
2. Die Anwender erhielten eine kurze Einführung über die zu testende Software, den Hintergrund der Studie sowie OpenProposal.
3. Die Anwender führten verschiedene Aufgaben durch um die Software zu testen und wurden gebeten Verbesserungsvorschläge mit OpenProposal abzugeben. Eine Kontrollgruppe setzte alternative Werkzeuge ein.
4. Anschließend wurden die zu untersuchende Software und die Nützlichkeit von OpenProposal durch die Anwender diskutiert. Außerdem wurden die erstellten Vorschläge bezüglich ihrer Verständlichkeit sowie Auffälligkeiten untersucht.
5. Zum Schluss wurden die Anwender gebeten einen Fragebogen auszufüllen.
6. In der zweiten Fallstudie wurden Anwender zudem darum gebeten den Prototyp im Rahmen ihrer täglichen Arbeit zu testen und wahlweise OpenProposal oder andere Werkzeuge zu verwenden um ihr Feedback abzugeben.

Tabelle 1 fasst die wesentlichen Merkmale der beiden Fallstudien zusammen. Die Tests wurden von dem jeweiligen Moderator vorbereitet und ausgewertet. In beiden Studien hatten die Entwickler lediglich über Telefon, Email und ein Issue-Trackersystem Kontakt mit Anwendern. Die Entwickler von OpenProposal waren als reine Beobachter vor Ort und weder

an den Softwareprojekten noch an der Organisation der Befragungen beteiligt. In beiden Fällen wurde JIRA, ein internetbasiertes Issue-Tracker-System zur Verwaltung von Aufgaben sowohl von den Teammitgliedern des Softwareprojektes als auch von den Anwendern selbst für Fehlerberichte und die Verwaltung von Aufgaben verwendet. Mit OpenProposal konnten Vorschläge zur weiteren Bearbeitung direkt an JIRA gesendet werden.

Fallstudie	# Unternehmen	# Mitarbeiter	# Teilnehmer	Hintergrund Anwender	Softwaretyp	Umfang
Nr. 1 „Bender“	1	5k – 10k	11 Anwender 1 Moderator 1 Entscheider 1 Entwickler	Training, Vertrieb, Techn. Doku, Support	3D-Konstruktion	2 Tage Usability Workshop
Nr. 2 „WAVES“	4	1-500	16 Anwender 1 Moderator 4 Entscheider 4 Entwickler	Softwareentwickler	Wiki, Suchmaschine	4x 1 Tag Interviews & 3 Monate „in use“

Tabelle 1: Übersicht über die Fallstudien

3.1 Fallstudie 1: Bender

Im September 2007 wurde eine kurze Fallstudie bei der Bender Group (Name geändert) mit Hauptsitz in Deutschland durchgeführt. Bender ist auf die Produktion von Fertigungstechnik spezialisiert, für die auch Konfigurationssoftware angeboten wird. Die Durchführung von Usability Workshops ist Teil des Softwareentwicklungsprozesses bei Bender, und während eines solchen wurde OpenProposal auf Anfrage des Unternehmens erprobt. Bei Bender wurden die Anmerkungen der Anwender bis dahin mittels Stift und Papier festgehalten und später in eine Tabelle übertragen. Dieser Prozess wurde von Praktikanten durchgeführt, dauerte mehrere Tage und war stets mit vielen Übertragungsfehlern behaftet. Am Ende war es somit schwierig den Sinn vieler Anforderungen nachzuvollziehen. Versuche, dieses Problem mit Hilfe computergestützter Werkzeuge wie z.B. einem Textverarbeitungsprogramm oder „Snapshot“-Programmen zu lösen, scheiterten an der mangelnden Akzeptanz der Anwender hierfür von der Testsoftware zu diesen Werkzeugen zu wechseln. Folglich bevorzugten die Anwender Stift und Papier.

Bezug nehmend auf die Hypothesen H1 und H2 wurde in dieser Fallstudie der Frage nachgegangen, ob Anwender dazu motiviert werden können ihr Feedback mittels OpenProposal elektronisch abzugeben und ob dies dazu beitragen kann die Kosten für Usability Workshops bei Bender zu reduzieren und ihre Ergebnisse zu verbessern. Als Vergleichssystem wurde JIRA herangezogen, das neben der Abgabe ausschließlich textlich formulierter Vorschläge auch das Anhängen von Bildschirmfotos bzw. Abbildungen ermöglicht.

Die kurze Studie umfasste elf Testpersonen (fünf weiblich, sechs männlich), einen Moderator, einen Entwickler und einen Entscheider. Der Altersdurchschnitt der Testpersonen lag bei 29,75 (min. 27, max. 47 Jahre). Sie wiesen unterschiedliche Erfahrungen im Umgang mit der Formulierung von Anforderungen vor; fünf waren erfahren und sechs unerfahren. Die Test-

personen hatten zuvor weder die Testsoftware noch OpenProposal gesehen. Da JIRA bereits vor der Studie für das Helpdesk und die Verwaltung von Aufgaben verwendet wurde, waren alle mit JIRA vertraut, ebenso mit der Produktpalette der Bender Software sowie mit der Konfiguration der Maschinen des Unternehmens. Die Testpersonen saßen gemeinsam in einem Raum, jeder an einem eigenen Arbeitsplatz mit einem Computer mit Netzwerkverbindung. Ein Whiteboard konnte für Hinweise und Diskussionen genutzt werden. Bei der getesteten Software handelte es sich um ein komplexes 3D-Planungsinstrument zur Konstruktion von 3D-Modellen von Maschinenteilen. Ziel des Usability Workshop war es die neue Software zu testen und gegebenenfalls neue Anforderungen festzustellen indem Aufgaben gelöst wurden, die an häufig von Kunden des Unternehmens gestellte Fragen angelehnt waren. Sobald ein Fehler oder eine Verbesserungsmöglichkeit entdeckt wurde, sollte dieser per OpenProposal bzw. JIRA gemeldet werden. Bei sehr schweren Programmfehlern wurde darum gebeten diese auf einem Whiteboard sichtbar für alle Tester zu platzieren, um so zu verhindern, dass derselbe Fehler ein zweites Mal angemerkt wurde. Am Ende jedes Workshoptages gab es eine gemeinsame Diskussion über die Eindrücke des vergangenen Tages.

Die Testpersonen wurden zufällig in zwei Gruppen geteilt. Am ersten Tag nutzte die eine Gruppe OpenProposal, die andere JIRA. Am zweiten Tag tauschten die Gruppen. Am Ende jedes Tages erhielten die Testpersonen einen Fragebogen zu dem Werkzeug, das sie an diesem Tag verwendet hatten.

3.2 Fallstudie 2: WAVES

Die zweite Fallstudie wurde von Juli bis Oktober 2008 im Rahmen des BMBF-Forschungsprojektes WAVES durchgeführt – einem Verbund aus vier Softwareunternehmen, der Freien Universität Berlin und dem FZI Forschungszentrum Informatik. Ziel des Forschungsprojektes ist es in verteilten Softwareprojekten den Aufbau und den Austausch von informellem Wissen zu fördern. Am FZI entwickelten hierzu zwei Teams, jedes bestehend aus zwei wissenschaftlichen Mitarbeitern (Entscheider) und zwei studentischen Entwicklern (Entwickler), Werkzeuge zur Unterstützung des Wissensmanagements von Softwareentwicklern. Bei den vier Softwarehäusern handelt es sich um kleine und mittlere Unternehmen aus Deutschland. Ihre Softwareentwickler hatten die Rolle der Anwender inne. Ein wissenschaftlicher Mitarbeiter der Freien Universität Berlin war verantwortlich für die Evaluation der vom FZI entwickelten Werkzeuge und nahm die Rolle des Moderators ein.

Im Rahmen der Fallstudie wurden die beiden Prototypen Wiquila und WavesIS getestet. Wiquila ist ein Wiki-Editor, welcher u.a. an mehrere Wikis angebunden werden kann, über eine WYSISWG-Ansicht verfügt, und eine Autovervollständigen-Funktion anbietet. WavesIS ist eine Suchmaschine, mit der gleichzeitig in verschiedenen Informationsquellen wie z.B. Wikis, Versionsverwaltungssystemen, etc. gesucht werden können. Beide Prototypen wurden in den Unternehmen installiert und in deren Systeme integriert.

Ziel der zweiten Fallstudie war es, die Hypothesen H2 und H3 zu untersuchen. Es wurde der Fragenstellung nachgegangen, ob OpenProposal dazu beitragen kann, die Ergebnisse von Anwenderinterviews zu verbessern. Außerdem soll untersucht werden, ob OpenProposal Anwendern erleichtert ihre Vorschläge auch während ihrer täglichen Arbeit abzugeben

Die Fallstudie erfolgte in zwei Phasen:

Phase 1: In den ersten zwei Wochen führte der Moderator bei den Unternehmen vor Ort Einzel-Interviews mit 16 Anwendern (zwei weiblich, 14 männlich) durch. Während der Interviews testeten die Anwender die Prototypen an einem Einzelarbeitsplatz, wobei sie ihre Anmerkungen laut aussprachen („thinking aloud“) und währenddessen von einer Videokamera gefilmt wurden. Die Anwender hatten stets die Möglichkeit OpenProposal zu benutzen. Der Altersdurchschnitt lag bei 35.36 Jahren (min. 29, max. 54 Jahre). Neun Personen waren mit der Formulierung von Anforderungen vertraut. Alle Testpersonen benutzen regelmäßig Wikis und Suchmaschinen. OpenProposal war keiner Person im Voraus bekannt. Nach jedem Test wurden die Anwender gebeten, einen Fragebogen zu OpenProposal (identisch mit dem aus der ersten Fallstudie) auszufüllen.

Phase 2: In den anschließenden drei Monaten wurden die Anwender gebeten, die beiden Prototypen für ihre tägliche Arbeit zu verwenden und Verbesserungsvorschläge während der Nutzung wahlweise per Email, Telefon, JIRA, OpenProposal oder das in WavesIS integrierte Feedback-Formular abzugeben. Nach Ablauf der drei Monate wurden mehrere Anwender, der Moderator, die Entscheider und die Entwickler in Interviews befragt.

3.3 Beobachtungen in den Fallstudien

Während der beiden Fallstudien konnten sieben Beobachtungen gemacht werden. Da JIRA in der ersten Fallstudie im direkten Vergleich mit OpenProposal stand, stammen Beobachtungen O3 und O4 aus Fallstudie 1. Beobachtungen O6 und O7 beziehen sich auf die zweite Fallstudie, da dies im Untersuchungsrahmen der ersten Fallstudie nicht vorgesehen war.

- O1. OpenProposal wurde von allen Teilnehmern akzeptiert und als hilfreich bewertet.
- O2. Alle Anmerkungen wurden elektronisch festgehalten. Zeit und Aufwand für die Vor- und Nachbereitung konnte verringert werden.
- O3. Im Vergleich zu JIRA wurde OpenProposal von einer größeren Anzahl an Teilnehmern genutzt und es wurden mehr Vorschläge abgegeben.
- O4. Die Vorschläge mit OpenProposal enthielten eine größere Anzahl an Worten als die mit JIRA.
- O5. Eine große Mehrheit der mit OpenProposal erstellten Vorschläge waren konstruktive Verbesserungsvorschläge.
- O6. Die mit OpenProposal erstellten Vorschläge förderten Gruppendiskussionen.
- O7. Während der täglichen Nutzung gaben die Anwender nur wenige Vorschläge mit OpenProposal ab. Mit JIRA wurde kein Vorschlag erstellt.

Zu O1 und O2: Die Testpersonen schienen sich schnell an OpenProposal und das Konzept der Annotation von Bildschirmfotos zu gewöhnen. Sie verwendeten OpenProposal nach unterschiedlichen Mustern; manche gaben eher mehr Text ein, andere weniger, der eine machte eine Annotation pro Vorschlag, der andere vereinte mehrere Ideen in einem Vor-

schlag. Die meisten Testpersonen nutzten die Möglichkeit direkt auf die Arbeitsoberfläche Bezug zu nehmen, manche ergänzten ihre Eingaben durch textbasierte Erläuterungen.

Frage	+++	++	+	o	-	--	---	Enthaltungen
Ich finde OpenProposal leicht zu bedienen	4x	8x	2x	1x	0x	0x	0x	0x
Ich finde die Vorschlagserstellung bei OpenProposal einfacher als bei JIRA	4x	4x	3x	2x	0x	0x	0x	2x
Ich finde die Vorschlagserstellung ist bei OpenProposal mit weniger Aufwand verbunden als bei JIRA	3x	5x	2x	3x	0x	0x	0x	2x

Tabelle 2: Ausschnitt aus den Antworten zu den Fragebögen

(„---“ = „Trifft überhaupt nicht zu“, „--“ = „Trifft größtenteils nicht zu“, „-“ = „Trifft eher nicht zu“, „o“ = „Weder noch“, „+“ = „Trifft eher zu“, „++“ = „Trifft größtenteils zu“, „+++“ = „Trifft voll und ganz zu“)

Tabelle 2 fasst die Antworten auf die wesentlichen Fragen der Fragebögen zusammen. Von 27 ausgegebenen Fragebögen kamen 15 vollständig zurück. Im Vergleich zu JIRA sei OpenProposal besser geeignet, Vorschläge mit geringerem Aufwand abzugeben. Die Testpersonen merkten an, dass sie OpenProposal nicht als Ersatz von JIRA ansahen, sondern ausdrücklich eine Kombination aus OpenProposal und JIRA vorziehen würden.

Die Befragung der Entwickler bei Bender und WAVES ergab, dass die mit OpenProposal erstellten Vorschläge als sehr verständlich bewertet wurden. Für den Entwickler waren die Bildschirmfotos zwar hilfreich, aber nicht unmittelbar notwendig, da sie sich in ihrem Programm auskannten und über die textliche Beschreibung orientieren konnten. Interviews mit den Moderatoren und Entscheidern ergaben, dass die elektronisch abgegebenen Vorschläge für eine weitere Bearbeitung geeignet waren und alle Erwartungen erfüllt wurden. Für sie waren die Bildschirmfotos sehr hilfreich, da sie dadurch schnell überblicken konnten, worum es sich bei den Vorschlägen handelte. Die Kombination aus OpenProposal und JIRA schien für sie die beste Wahl zu sein und soll auch in zukünftigen Workshops und Interviews Anwendung finden. Die Zeit zur Vor- und Nachbereitung der Anwenderbeteiligung konnte aufgrund der rein elektronischen Datenhaltung stark reduziert werden. Direkt nach den Anwenderbefragungen standen die OpenProposal-Vorschläge zur weiteren Verarbeitung zur Verfügung und konnten zeitnah an Entscheider und Entwickler weitergeleitet werden.

Zu O3: Während des Bender-Workshops wurden 30 Vorschläge mit OpenProposal und 14 Vorschläge mit JIRA abgegeben. Auffällig war, dass neun der elf Testpersonen mit OpenProposal Vorschläge erstellten. Mit JIRA gaben lediglich drei Personen Vorschläge ab. Zwei Testpersonen erstellten keinen einzigen Vorschlag.

Zu O4: Die mit OpenProposal abgegebenen Vorschläge enthielten durchschnittlich eine Annotation mit einer aus 12,86 Worten bestehenden erläuternde Beschreibung sowie einen 10,90 Worte umfassenden Titel inklusive allgemeiner Kurzbeschreibung, also im Gesamten 23,76 Worte. Die ausschließlich textbasierten JIRA Vorschläge besaßen einen aus durchschnittlich 18,64 Worten bestehenden Titel und Textkörper. Die in WAVES abgegebenen

Open-Proposal-Vorschläge enthielten durchschnittlich 2,3 Annotationen mit 12,1 Worten pro Annotation, in der Summe 29,86 Worte pro Vorschlag.

Zu O5: Im Bender-Workshop waren 25 der 30 OpenProposal-Vorschläge konstruktive Verbesserungsideen. Im Gegensatz hierzu waren 10 der 14 JIRA-Vorschläge ausschließlich Problembenachrichtigungen. Während der WAVES-Interviews wurden 70 Vorschläge mit OpenProposal erstellt. Bei 89% der Vorschläge handelte es sich um konstruktive Verbesserungsideen. Die restlichen Vorschläge waren Problembenachrichtigungen. Dies wurde von den Entscheidern und Entwicklern positiv bewertet, da ihnen die Arbeit mit den Vorschlägen mehr Spaß bereitete. Einem der Entscheider erschien der Ton der annotierten Bildschirmfotos deutlich angenehmer als die üblichen textbasierten Vorschläge. Der Moderator merkte an, dass die Anwender mit OpenProposal gezwungen wären, ihre Ideen zu konkretisieren. Mit der „thinking aloud“-Methode fiel es den Anwendern hingegen leichter generelle Anmerkungen zu formulieren.

Zu O6: In der zweiten Fallstudie fand sich bei zwei Unternehmen direkt im Anschluss an die Interviews Zeit für einen kurzen Workshop mit den Teilnehmern. Während dieser Workshops wurden die OpenProposal Vorschläge auf einem Bildschirm präsentiert und diskutiert. Es wurde festgehalten, dass die annotierten Bildschirmfotos hilfreich für die Diskussion waren und mit Videoanalysen oder Textdokumenten nicht denkbar gewesen wären.

Zu O7: In der zweiten Phase der WAVES-Studie griffen die Teilnehmer nur fünf Mal auf die Möglichkeit zur Abgabe von Vorschlägen zurück, obwohl alle regelmäßig via Email und Telefon erinnert wurden. Die Anwender begründeten dies mit ihrer knappen Zeit während ihrer täglichen Arbeit. Nur bei schwerwiegenden Fehlern entstand Handlungsbedarf. In diesen Fällen wurde OpenProposal verwendet und als hilfreich empfunden. JIRA und das integrierte Feedbacksystem wurden kein einziges Mal verwendet.

4 Fazit

Die positiven Rückmeldungen und die zahlreichen Beobachtungen aus den beiden Fallstudien unterstützen die Hypothesen H1 und H2. Dies zeigt auch, dass die identifizierten Gestaltungsziele A1-4 bestätigt werden können. Allerdings stößt der OpenProposal Ansatz auch auf zahlreiche Limitierungen: Alle Teilnehmer waren sich einig darin, dass das Annotieren von Bildschirmfotos in manchen Fällen nicht die am besten geeignete Methode sei. Daher sollte dies mit anderen Werkzeugen wie z.B. Whiteboards oder Videoanalysen kombiniert werden können. Ferner wurde OpenProposal nur selten zur Abgabe von Feedback im Rahmen der täglichen Arbeit eingesetzt. Somit kann für Hypothese H3 keine Unterstützung gefunden werden. Hierfür sind weitere Untersuchungen und Anpassungen von OpenProposal notwendig, wobei in Frage gestellt werden muss, ob dies mit dem OpenProposal Ansatz oder einer anderen Art von Werkzeug überhaupt lösbar ist.

Die vorliegende Arbeit zeigt, dass der in OpenProposal gewählte Ansatz den Kommunikationsprozess in Softwareprojekten unterstützen kann. In zwei Fallstudien wurde das Potential von OpenProposal anhand von realen Projekten erfolgreich demonstriert. OpenProposal wurde nach Abschluss der Fallstudien bei Bender und zwei Unternehmen der zweiten Fall-

studie weiterhin eingesetzt. Besonders hilfreich für die Unternehmen war es, dass der Dokumentationsaufwand mit OpenProposal verringert und die Anwender zu konstruktiven Verbesserungsvorschlägen „befähigt“ werden konnten. Allerdings können diese Erkenntnisse nur richtungsweisend sein. Zu einer Generalisierung dieser Aussagen ist ergänzend zur gewählten qualitativen Methodik eine quantitative Untersuchungsmethode notwendig.

Literaturverzeichnis

- Danielsson, K., A. M. Naghsh, et al. (2008). Distributed participatory design. Extended abstracts of the workshop session. Conference on Human Factors in Computing Systems (CHI). Florence, Italy.
- Dörner, C., J. Heß, et al. (2008). Fostering user-developer collaboration with infrastructure probes. In: International workshop on Cooperative and human aspects of software engineering at the International Conference on Software Engineering Leipzig, Germany, ACM.
- Fogli, D., G. Fresta, et al. (2004). "On Electronic Annotation and Its Implementation." ACM AVI: 98-102.
- Keil, M. and E. Carmel (1995). "Customer-Developer Links in Software Development." Communications of the ACM 38(5): 43-51.
- Kujala, S. (2008). "Effective user involvement in product development by improving the analysis of user needs." Behaviour & Information Technology 27(6): 457-473.
- Lohmann, S., J. Ziegler, et al. (2008). Involving End Users in Distributed Requirements Engineering. In: Second Conference on Human-Centered Software Engineering, Pisa, Italy.
- Naghsh, A. M., A. Dearden, et al. (2005). Investigating Annotation in Electronic Paper-Prototypes. In: International Workshop on Design, Specification and Verification of Interactive Systems, Newcastle upon Tyne, England.
- Stevens, G. and S. Draxler (2006). Partizipation im Nutzungskontext. In: Mensch & Computer 2006, Gelsenkirchen, Oldenbourg Verlag.

Kontaktinformationen

Asarnusch Rashid

FZI Forschungszentrum Informatik, Karlsruhe
E-Mail: rashid@fzi.de