

Benutzerfeedback für die Softwareweiterentwicklung erheben

Melanie Stade¹, Norbert Seyff^{1,2}, Farnaz Fotrousi^{1,3}, Emitza Guzman², Samuel A. Fricker^{1,3}, Martin Glinz², Ronnie Schaniel¹

¹ Fachhochschule Nordwestschweiz, i4Ds Centre for Requirements Engineering (CeRE)

² Universität Zürich, Institut für Informatik

³ Blekinge Institute of Technology, Software Engineering Research Laboratory (SERL-Sweden)

1. Motivation und Problem

Die erfolgreiche Weiterentwicklung von Softwaresystemen beruht auf der Ermittlung von Anforderungen, welche die Bedürfnisse der verschiedenen Stakeholdergruppen widerspiegeln. In diesem Zusammenhang sind Endbenutzer der Software eine wichtige Stakeholdergruppe. Deren Zufriedenheit (Quality of Experience) mit dem Softwaresystem ist ein entscheidender Faktor für den (kontinuierlichen) Erfolg dieses Softwaresystems.

Eine persönliche Kontaktaufnahme von Softwareherstellern oder Entwicklern mit Endbenutzern ist oft schwierig. Herkömmliche Ermittlungsmethoden wie z.B. Interviews und Workshops erlauben es nicht, eine große Anzahl von (unbekannten) Endbenutzern in die Weiterentwicklung von Softwaresystemen einzubinden. Somit bieten herkömmliche Ermittlungsmethoden unzureichend Unterstützung für die kontinuierliche Ermittlung von Endbenutzeranforderungen.

Soziale Medien wie Twitter und Facebook [1][2], aber auch Applikationen, die gezielt für die Kommunikation von Feedback und Anforderungen entwickelt wurden [3][4][5], sind vielversprechende Ansätze, um eine große Anzahl von Endbenutzern in die Softwareweiterentwicklung einzubeziehen [6]. Diese Feedbackmechanismen haben jedoch einen großen Nachteil: Der Endbenutzer muss Zugang zu diesen Kommunikationskanälen haben.

2. Lösungsansatz

Ein Lösungsansatz ist das Ermitteln von Anforderungen mit Hilfe von in Softwaresysteme integrierte Feedbackmechanismen, welche es Endbenutzern ermöglichen, aktiv Feedback zu geben und somit ihre Probleme und Wünsche in Bezug auf ein Softwaresystem zu kommunizieren. Zum Beispiel kann ein Endbenutzer durch die Eingabe einer kurzen textbasierten Stellungnahme kommunizieren, dass Berechnungen, die durch die Software durchgeführt werden, zu langsam sind. Solche Feedbackmechanismen erlauben das Ermitteln von Informationen, aus denen Anforderungen abgeleitet werden können, ohne dass ein Requirements Engineer den Kontakt zu den Endbenutzern aufnehmen muss. Somit können viele Endbenutzer in die Ermittlung von Anforderungen mit einbezogen werden.

Zusätzlich zur Erhebung von Benutzerfeedback können Feedbackmechanismen auch das Softwaresystem

und den Systemkontext überwachen (Monitoring), und durch diese Überwachung weitere relevante Informationen für die erfolgreiche Auswertung der Daten sammeln. In Bezug auf unser Beispiel könnte es hilfreich sein, die Berechnungsgeschwindigkeit der Software auch automatisiert zu überwachen. Somit kann bei der späteren Analyse der Daten das Problem genauer identifiziert und eingegrenzt werden. Stellt sich bei der Analyse heraus, dass z.B. nur wenige Endbenutzer, welche langsame Hardware für die Ausführung der Software nutzen, von dem Leistungsproblem betroffen sind, dann kann der Hersteller entscheiden, das Problem zu ignorieren. Handelt es sich aber um ein generelles Leistungsproblem, dann muss der Hersteller reagieren und das Problem beheben.

3. Ergebnisse

Unserer Forschung im Rahmen der EU Horizon2020 SUPERSEDE (grant agreement no 644018) und Wise-IoT (grant agreement no 723156) Projekte zeigt erste wichtige Erkenntnisse, die bei der Entwicklung und Nutzung von Feedbackmechanismen beachtet werden sollten. Die Ergebnisse beruhen auf Interviews mit Industriepartnern im Projekt, einer derzeit durchgeführten Literaturstudie und der begonnenen Entwicklung und Evaluierung einer Feedback Plattform (Abb. 1).

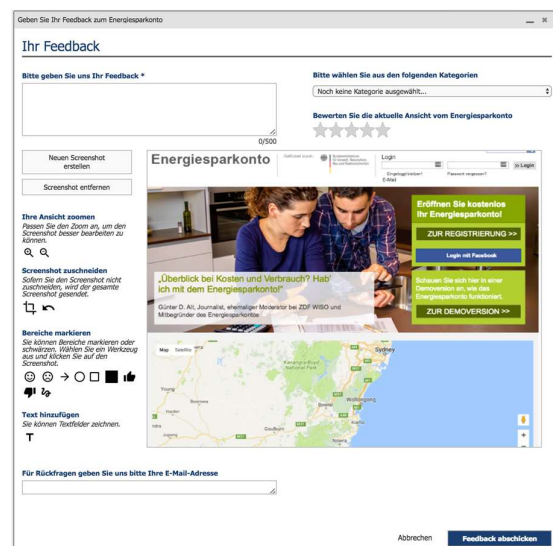


Abb. 1 Bildschirmfoto einer möglichen Konfiguration in der Feedback Plattform des SUPERSEDE Projektes.

Motivation der Endbenutzer: Motivation von Endbenutzern ist der erste wichtige Schritt für die erfolgreiche Erhebung von Benutzerfeedback. Es ist wichtig, Endbenutzern klar zu kommunizieren, dass Feedback erwünscht ist. Ein gut sichtbarer Button und aktive Aufrufe z.B. mittels Newsletter erscheinen hier wirkungsvoll. Ferner sollte dem Endbenutzer kommuniziert werden, dass sein Feedback vom Softwarehersteller ernst genommen wird. Eine Reaktion des Softwareherstellers auf Feedback, beispielsweise in Form einer Dankes-Email, wäre ein Mindestmaß. Zu erforschen bleibt, inwieweit die Art der Software (z.B. Software im Arbeitsalltag versus Unterhaltungssoftware) sich auf die Motivation der Endbenutzer, Feedback zu geben, auswirkt.

Starten des Feedbacks: Die Feedbackkommunikation kann der Endbenutzer („push“) oder der Softwarehersteller („pull“) anstoßen [7]. Im Beispiel oben startet der Endbenutzer die Kommunikation. Eine andere Variante wäre, dass nach dem Berechnungsvorgang automatisch ein Dialog startet, der die Zufriedenheit des Endbenutzers mit dem Ergebnis abfragt. Herauszufinden, wie sich „push“ und „pull“ kombinieren lassen und was in welchem Kontext von Endbenutzern präferiert wird, ist Ziel unserer Forschungsarbeit.

Format der Feedbackkommunikation: Dem Endbenutzer stehen verschiedene sprachliche und nicht-sprachliche Formate zur Kommunikation von Feedback zur Verfügung [8]. Neben einer rein textbasierten Beschreibung kann der Endbenutzer eine Bewertung des Softwaresystems (z.B. auf einer Sterne-Skala) vornehmen oder dem Softwarehersteller eine kurze Sprachaufnahme senden. Auch kann er ein Foto der aktuellen Bildschirmansicht erstellen und relevante Bereiche markieren, in unserem Beispiel den Fortschrittsbalken. Sein Feedback kann der Endbenutzer dann wiederum einer Kategorie zuordnen (z.B. „Problem“). Die von uns entwickelte Feedback Plattform unterstützt diese Formate und bietet Endbenutzern somit maximale Flexibilität bei der Feedbackkommunikation.

Analyse des Feedbacks: Das Format der Feedbackkommunikation beeinflusst den Aufwand bei der Auswertung der Feedbacks. Für textbasierte Formate stehen erste Werkzeuge zur automatischen oder teilautomatischen Auswertung zur Verfügung [9]. Bei einer großen Anzahl von Feedbacks ist eine visuelle Aufbereitung der Ergebnisse sinnvoll [10]. Eine Herausforderung ist die Auswertung von Feedbacks, welche Bildschirmfotos enthalten. Ein erster Lösungsansatz besteht darin, dass der Endbenutzer das Bildschirmfoto mit Icons markiert, die eine klar definierte Bedeutung haben, z.B. „Löschen“, „Verschieben“ [11].

Anpassungsmöglichkeiten: Endbenutzer unterscheiden sich nicht nur in ihren Motiven, sondern auch in ihren Einstellungen und Präferenzen hinsichtlich des Feedbackgebens [12]. Auch der Kontext (z.B. das verfügbare Endgerät) kann Einfluss darauf haben, wie ein Endbenutzer Feedback kommunizieren möchte [13]. Die Entscheidung auf Seiten des Softwareherstellers, welche Feedbackmechanismen eingesetzt werden,

hängt von verfügbaren Ressourcen ab, aber auch von der angestrebten Qualität der Feedbacks.

4. Was ist neu?

Viele Softwareunternehmen verwenden heute traditionelle Feedbackmechanismen wie Telefonhotlines oder Kontaktformulare. Einige Unternehmen nutzen bereits integrierte Feedbackmechanismen, die jedoch oft nur textbasiertes Feedback unterstützen. Das Ziel unserer Forschung ist es eine Feedback Plattform zu entwickeln, welche in Apps und Websites verschiedene Feedbackmechanismen unterstützt und es Entwicklern und Endbenutzern somit erlaubt, eine gewählte Feedbackermittlungsstrategie laufend auf ihre Bedürfnisse anzupassen.

Literatur

- [1] E. Guzman, Alkadhi, R., Seyff, N., „A needle in a haystack: What do Twitter users say about software,” in *RE*, 2016.
- [2] Seyff, N., et al., „Using popular social network sites to support requirements elicitation, prioritization and negotiation,” in *Journal of Internet Services and Applications*, 2015.
- [3] Seyff, N., Bortenschlager, M., Ollmann, G., „iRequire: Gathering end-user requirements for new apps,” in *RE*, 2013.
- [4] Stade, M. J., Wittkugel, H. J., Seyff, N., „Evaluation des iRequire-Ansatzes: Anforderungsermittlung in der Praxis,” in *Mensch und Computer*, 2015.
- [5] Seyff, N., Ollmann, G., Bortenschlager, M., „AppEcho: A user-driven, in situ feedback approach for mobile platforms and applications,” in *MOBILESoft*, 2014.
- [6] Groen, E.C., Doerr, J., Adam, S., „Towards crowd-based Requirements Engineering: A research preview,” in *Requirements Engineering: Foundation for Software Quality*, LNCS 9013, 2015.
- [7] W. Maalej, H.-J. Happel, A. Rashid, „When users become collaborators: towards continuous and context-aware user input,” in *OOPSLA*, 2009.
- [8] I. Morales-Ramirez, A. Perini, R. Guizzardi, „An ontology of online user feedback in software engineering,” in *Applied Ontology*, 2015.
- [9] E. Guzman, W. Maalej, „How do users like this feature? A fine grained sentiment analysis of app reviews,” in *RE*, 2014.
- [10] E. Guzman, P. Bhuvanagiri, B. Bruegge, „FAVe: visualizing user feedback for software evolution,” in *VISSOFT*, 2014.
- [11] A. Rashid, et al. „Bringing developers and users closer together: the open proposal story,” in *Multikonferenz Wirtschaftsinformatik*, 2008.
- [12] M. Almaliki, C. Ncube, R. Ali, „Adaptive software-based feedback acquisition: a persona-based design,” in *RCIS*, 2015,
- [13] J. Hess, L. Wan, B. Ley, V. Wulf, „In-situ everywhere: a qualitative feedback infrastructure for cross platform home-IT,” in *EuroITV*, Berlin, 2012.