

User Involvement in Software Evolution Practice: A Case Study

Interview Questions

The following sections include the interview questions we used to study user involvement in software evolution practice [1]. We conducted semi-structured, open-ended interviews, which allow for improvisation and thus facilitate an exploration of the studied cases. To this end, we provided each subject with the question catalogue one week before the interview. We explicitly called their attention to the semi-structured nature of the interview, and underlined that the questions should be regarded as support. If a subject was working in multiple projects, we asked her to select one project which was developed for a large number of end users.

1 Project Information

1. Of which **type** is the **project**?
 - closed source,
 - open source,
 - other:
2. Which **kind of software** is the project? For instance,
 - mobile application,
 - desktop application,
 - on the shelf-component,
 - customized software,
 - other:
3. Which **user audience** do you target?
 - any user,
 - special group of users:
 - other:
4. **How many active users** does your software have?
 - less than 100: ____
 - between 100 and 500
 - between 500 and 1.000
 - between 1.000 and 5.000
 - between 5.000 and 10.000
 - more than 10.000: _____
 - no answer
5. How frequently do you **release** a new external version?
 - more often than every 2 weeks,
 - every 2 to 4 weeks,
 - every 4 to 8 weeks,

1 *Project Information*

- () every 2 to 6 months,
- () less often than every 6 months

2 User Feedback - Current Landscape

1. How can your users **contribute** to improve your software?
 - by reporting errors
 - by requesting new features
 - by providing feedback on existing features
 - by requesting improvements or enhancements
 - by rating the product
 - other:
2. How often can your users **contribute** to improve your software?
 - at the beginning of a project (e.g. by focus groups)
 - regularly (e.g. before you plan a new major release)
 - continuously (i.e. at any time during the software lifecycle)
 - other:
3. How do your users **report errors**? (**often**, **sometimes**, **rarely**, **never**)
 - by using a public issue tracker,
 - by sending email,
 - by calling on the phone,
 - by writing blog posts,
 - by writing to mailing-lists,
 - by publishing their opinion on other publicly available media,
 - by using a reporting mechanism which is integrated in the software,
 - by using a specialized service (e.g. user-voice, get-satisfaction),
 - other:
4. How do your users **request new features**? (**often**, **sometimes**, **rarely**, **never**)
 - we actively carry out studies with a representative group,
 - by using a public issue tracker,
 - by sending email,
 - by calling on the phone,

2 User Feedback - Current Landscape

- by writing blog posts,
 - by writing to mailing-lists,
 - by publishing their opinion on other publicly available media,
 - by using a feedback mechanism which is integrated in the software,
 - by using a specialized service (e.g. user-voice, get-satisfaction),
 - other:
5. How do your users provide **feedback on existing features**? (often, sometimes, rarely, never)
- we actively carry out studies with a representative group,
 - by using a public issue tracker,
 - by sending email,
 - by calling on the phone,
 - by writing blog posts,
 - by writing to mailing-lists,
 - by publishing their opinion on other publicly available media,
 - by using a feedback mechanism which is integrated in the software,
 - by using a specialized service (e.g. user-voice, get-satisfaction),
 - other:
6. How do your users request **enhancements and improvements** to your software? (often, sometimes, rarely, never)
- we actively carry out studies with a representative group,
 - by using a public issue tracker,
 - by sending email,
 - by calling on the phone,
 - by writing blog posts,
 - by writing to mailing-lists,
 - by publishing their opinion on other publicly available media,
 - by using a feedback mechanism which is integrated in the software,
 - by using a specialized service (e.g. user-voice, get-satisfaction),
 - other:

3 User Feedback - Current Workflow and Problems

1. Why is user feedback **interesting** for you?
 - it helps to improve the software quality (e.g. to fix bugs, to improve features)
 - it helps to find missing features
 - it helps to understand what users want and need
 - it helps to advertise and market the application
 - it helps to understand, if and how the product is accepted (e.g. by ratings)
 - other:
2. How do you **process** user feedback, why, and which problems do you encounter?
3. Which activities **take most time** while processing user feedback?
4. Which activities **are the most difficult** while processing user feedback?
5. Do you **relate single reports** to each other (e.g. to identify duplicates or antipodes)?
 - yes, these get mapped automatically by a tool
 - yes, our users correlate their feedback themselves
 - yes, we do this manually before starting the analysis
 - yes, we do this successively while analyzing the single reports
 - no, we don't. Reason:
6. **How easy** or intuitive is it for you to relate different feedback to each other?
 - very easy
 - somewhat easy
 - undecided
 - somewhat difficult
 - very difficult
7. **According to what** do you relate single reports?
 - if both users had the same experience with the software

3 User Feedback - Current Workflow and Problems

- if the reports are duplicates
 - if the reports are antipodes
 - if one feedback details another
 - if the reports concern the same feature
 - if the type of the reports is the same (e.g. both are feature requests)
 - other:
8. Overall, **how satisfied** are you with your current practice of processing user feedback?
- very satisfied
 - somewhat satisfied
 - undecided
 - somewhat unsatisfied
 - very unsatisfied

4 User Feedback - Potentials and Challenges

1. Would you embrace **tool support** to **relate** user feedback to each other and why?
2. **When** should user feedback be related by a tool
 - the tool should show the user relevant existing feedbacks to allow her to comment on, rate, or vote for them (proactive)
 - the tool should simply collect user feedback. The gathered reports should be analyzed and related later (reactive)
 - other:
3. **According to what** should user feedback be related by a tool? (open question)
 - if the users had the same experience with the software
 - if the reports are duplicates
 - if the reports are antipodes
 - if one feedback details another
 - if the reports concern the same feature
 - if the type of the reports is the same (e.g. both are feature requests)
 - other:
4. Is it important for you to assess the **potential** of a specific user feedback **to improve your software** and why?
 - a) How would you **assess** the **importance** of a specific user feedback for **your project**?
 - i. by the quantity of its occurrence,
 - ii. by assessing the individual user who contributed,
 - iii. only after working on it (retrospective),
 - iv. not at all. Reason:
 - v. other:
 - b) Would you embrace **tool support** for this assessment and why?
5. Is it important for you to assess the **importance** of a specific user feedback **to the user community** and why?

4 User Feedback - Potentials and Challenges

- a) How would you **assess the importance** of a specific user feedback to the **user community**? (multiple answers allowed)
 - i. by the quantity of its occurrence,
 - ii. by assessing the individual user who contributed,
 - iii. only after working on it (retrospective),
 - iv. not at all. Reason:
 - v. other:
 - b) Would you embrace **tool support** for this assessment and why?
6. Can you tell if there are **specific individual users** that provide particularly important or frequent information, and how?
- a) Do you or would you **treat their feedback differently**
 - i. yes, we read it first
 - ii. yes, if we have a lot of similar feedback, we only / first read the feedback of this user
 - iii. no, we don't. Reason:
 - iv. other:
 - b) Do you or would you let the users or the **community know that they are important** and how?
 - i. yes, we let important users know that their feedback is important to us
 - ii. yes, we let a specific user know if a specific feedback was important to us
 - iii. yes, we let the community know which users are important to us
 - iv. yes, we let the community know which feedback is important to us
 - v. no, we don't. Reason:

5 Personal Information

1. How would you describe **your role** in the project?
 - requirements engineer
 - developer
 - tester
 - architect
 - project manager
 - other:
2. How many years of **experience** do you have in your main role?
 - less than 1 year,
 - between 1 and 2 years,
 - between 3 and 5 years,
 - between 6 and 10 years,
 - more than 10 years

Bibliography

- [1] Dennis Pagano and Bernd Bruegge. User Involvement in Software Evolution Practice: A Case Study. In *Proceedings of the 35th International Conference on Software Engineering*, San Francisco, CA, USA, 2013. IEEE.