A Context-sensitive User Feedback Mechanism for Continuous Software Evolution

Dora Dzvonyar (presented by Lukas Alperowitz)
Problems with User Feedback

communication gap

too many channels

missing structure & information

no clear process

[e.g. ANA14, Dam96, HL09, Kuj03, Pag13, Sch11]
Context-Aware FEedback System

Framework and Process combining approaches from 3 domains:

• Integration into development workflow
• Context-awareness
• Feedback traceability
Integration of Feedback into Development

• Feedback should be considered regularly [Dam96, HL09, Sch11]

• But: existing process models lack concrete integration into dev tasks

• CAFE stores user feedback in the issue tracker
  • Used daily by developers (acceptance)
  • Less hassle to look at feedback
  • Easier transformation into requirements

Context-Awareness

- Compensate for the user’s lack of awareness of what is important to the development team [DSC11, Pag13]
- CAFE stores context data, e.g.
  - Usage environment information
  - User’s location in the application
  - User’s last interactions with the application
  - Screenshot of problem
Feedback Traceability

- Feedback: source or consequence of requirement
- Evolution of requirements through feedback in the same system
Overview over CAFE

Extract Requirement

Including e.g. physical location, current screen, interaction steps

User can enrich feedback with context-based suggestions: “Does your feedback concern...?”

Where does the user see this change?
Evaluation

• User experiments
  • Users prefer CAFE to other means of providing feedback
  • Users are willing to provide additional details if they know about the feature

• Developer interviews
  • Usage context tracking is adequate and useful (granularity, form)
  • Saving feedback in the issue tracker is convenient
  • Expect better communication with the user base and increased user satisfaction
Conclusion & Future Work

• Demonstrated feasibility of CAFE
• Results indicate that Feedback Traceability is a valuable contribution
• Combine CAFE with AppStore —> marketing vs. product improvement
• Long-term evaluation
  • Do developers consider feedback more regularly?
  • More efficient feedback analysis?
  • Value of feedback traceability
Literature #1


Literature #2


Backup
Overview over CAFE

Extract Requirement

“Does your feedback concern…?”

store usage context

Where does the user see this change?
Overview over CAFE

- Extract Requirement

Integration into Dev.

- Developer
  - specify context of changes
  - release application
  - Release
  - install & use application

- User
  - give feedback
  - [has feedback]
  - [has no feedback]
  - [does not want to add details]
  - specify details to feedback
  - send feedback

- WorkItem
  - convert feedback to work item
  - analyze feedback
  - [need to act]
  - [no need to act]
  - prioritize work item
Overview over CAFE

Context-Awareness

“Does your feedback concern…?”

store usage context

Where does the user see this change?
Overview over CAFE

“Does your feedback concern…?”
Evaluation

- User experiments

The following feature of the feedback system was useful to me:
- giving feedback from within the application (instead of e.g. via e-mail or the browser)
- specifying what kind of feedback I was leaving (comment, feature request or bug report)
- receiving suggestions from the application ("Does your feedback concern...?")
Evaluation

• Semi-structured developer interviews