

Study Title: Understanding Developers' Perspective on Feature or Enhancement Requests in Moodle

Purpose: In open-source software ecosystems, maintainers and developers play a pivotal role in filtering, evaluating, and translating community input into actionable work. However, many of these discussions begin outside the formal development infrastructure. This part of the study seeks to understand developers' practices and perspectives in discovering, assessing, and integrating community-originated feature requests and to identify barriers and opportunities for improving the flow of ideas across these boundary spaces. Our aim with this study is to understand the traceability of these requests, how ideas evolve from initial discussion to implementation decisions, and how the reasoning for accepting or rejecting suggestions is preserved or lost throughout this process.

Interview Draft:

Introduction and Consent Form (Use the IRB approved form)

Warm-Up and Ice-Breaker Questions

1. What is your name and how old are you?
2. Where do you currently reside?
3. Highest degree of education you have completed?
4. Which areas or roles within a development team are you actively involved in? (e.g., frontend, backend, DevOps, project management, etc.)
5. How long have you been using GitHub? (e.g., Less than 1 year, 1-3 years, 3-5 years, 5+ years)
6. How long have you been contributing to Moodle OSS?
7. Can you tell me a little about your involvement in Moodle as a developer? What kind of contributions have you made? Design changes, core functionality, plugins?

RQ1. How do developers become aware of feature requests originating from community forums?

1. How do you usually find out about feature or enhancement requests that come from community discussions or forums?
2. What channels do you or your team use to monitor user suggestions (e.g., mailing lists, discussion forum, issue tracker)?

3. Is there a formal or informal process for collecting or flagging these ideas for consideration?
4. Who typically brings such ideas to your attention - users, moderators, or other developers?

RQ3. How are ideas from community forums translated into formal issue tracker entries?

1. When a feature idea comes from a community forum, how does it get turned into a formal issue in your tracker?
2. Who usually performs that translation - the original user, a moderator, or a developer?
3. How are discussions reframed or summarized when moving them to the tracker?
4. What information tends to get lost during this process, if any?

RQ2. How do developers evaluate and prioritize user-suggested features?

1. How do you determine whether a user's feature request is valuable or feasible to implement?
2. What factors influence prioritization - technical complexity, alignment with project goals, or community demand?
3. How do you balance community-driven requests against your existing roadmap or technical debt?
4. Who is involved in making final prioritization decisions, and how are these decisions communicated?

RQ4. How do developers communicate with users about feature requests?

1. How often do you or your team engage directly with users about their feature requests?
2. When do you communicate with users/requesters about the feature request?
3. What kinds of messages or responses do you typically provide to manage user expectations?
4. How do you handle situations where requests are infeasible, unclear, or has natural language defects?
5. What communication challenges arise when engaging with non-technical or new users?

RQ5. What mechanisms do developers use to ensure user requests are acknowledged or tracked?

1. What steps do you take to ensure user-suggested features are acknowledged and not lost?
2. Do you use any tagging, linking, or tracking mechanisms between community spaces and issue trackers?
3. How do you notify or update users when progress is made on a feature they suggested?
4. Are there resources or cultural constraints that limit consistent follow-up?

RQ6. What challenges or barriers prevent closer alignment between community discussions and issue tracking?

1. What factors make it difficult to maintain continuity between community discussions and the issue tracker?
2. Do tool limitations or platform separation contribute to these challenges?
3. Are time, labor, or governance constraints part of the problem?
4. How do you view the boundary between community dialogue and formal development work?

RQ7. What improvements or tools could help bridge community forums and issue trackers more effectively?

1. What practices or tools could make it easier to connect user feedback with development workflows?
2. How could better moderation, tagging, or summarization help identify valuable ideas?
3. What role could automation, bots, or AI play in linking forum discussions to existing issues?
4. In your view, what would an ideal feedback loop between users and developers look like?

Closing

[Thank the participant for their time and insights. Reconfirm that their data will remain confidential. End recording and debrief. Send the gift card]

