Developers' Shared Conversations with ChatGPT in GitHub Pull Requests and Issues

11/12/25

Written by Huizi Hao, Kazi Amit Hasan, Hong Qin, Marcos Macedo, Yuan Tian, Steven H. H. Ding, Ahmed E. Hassan

Presented by Tongwei Zhang

UNIVERSITY OF WATERLOO

Background (from textbook) & Motivation

- Empirical SE now leverages rich repository data; quantitative trends benefit from qualitative context.
- This study observes how developers share ChatGPT conversations within GitHub PRs & issues.
- **Goal: characterize what developers ask, how multi-turn chats flow, and why/where/who shares links.**

Problem To Be Solved by this paper

- Prior work largely benchmarks FM tools or surveys small user groups.
- We lack observational evidence about how developers actually use & share ChatGPT during collaboration on GitHub.
- Key gap: dynamics of what they ask, how multi-turn chats evolve, and how shared links are used across participants and PR/issue surfaces

Study Design

- **Data source & scope**
  - Use **DevGPT** dataset of developer–ChatGPT links embedded across GitHub artifacts; focus on PRs & issues; preprocess to keep English and deduplicate.
  - Final pool: **580** shared conversations; analyze **ALL first prompts** (RQ1), **ALL 189 multi-turn convos** (645 prompts; RQ2), and a **stratified sample of 250 PR/issue threads** for sharing behavior (RQ3).
- **Method overview**
  - RQ1: Manually code ALL 580 initial prompts into a 16-category taxonomy.
  - RQ2: Examine 189 multi-turn conversations (645 prompts) → prompt-role taxonomy + flow patterns.
  - RQ3: Stratified sample of 250 threads for sharing behavior (rationale, location, who).

Key Findings: RQ1 - What types of software engineering inquiries do developers present to ChatGPT in the initial prompt?

- Scope: Manually coded 580 initial prompts (PRs 210, Issues 370) into a 16-type taxonomy.
- Most frequent inquiry types: Code generation, Conceptual, How-to, Issue resolving, Review.
- Why initial prompt? Most conversations are single turn (PRs 66.8%, Issues 63.1%), and follow-up turns don't introduce new inquiry types.

Category	PR	Issue
SE-related:	198(100%)	329(100%)
(C1) Code generation	40 (20%)	90 (27%)
(C2) Conceptual	37 (18%)	45 (14%)
(C3) How-to	26 (13%)	74 (22%)
(C4) Issue resolving	24 (12%)	46 (14%)
(C5) Review	18 (9%)	12 (4%)
(C6) Comprehension	13 (7%)	10 (3%)
(C7) Human language translation	11 (6%)	0 (0%)
(C8) Documentation	10 (5%)	7 (2%)
(C9) Information giving	8 (4%)	8 (2%)
(C10) Data generation	4 (2%)	12 (4%)
(C11) Data formatting	2 (1%)	9 (3%)
(C12) Math problem solving	2 (1%)	9 (3%)
(C13) Verifying capability	2 (1%)	1 (0%)
(C14) Prompt engineering	1 (0%)	0 (0%)
(C15) Execution	0 (0%)	4 (1%)
(C16) Data analysis	0 (0%)	2 (1%)
Others	12	41

Key Findings: RQ2 - What roles do prompts play in multi-turn chats?

- Scope: 189 multi-turn conversations; 645 prompts; 7 roles via open coding.
- Top roles:
  - M1 Iterative follow-up: PR 33%, Issues 40%
  - M2 Reveal initial task: 26% / 29%
  - M3 Refine prompt: 17% / 14%
  - Others: Info giving (8/6), New task (7/4), Negative feedback (6/2), Ask clarification (4/5).

Categories	PR	Issues
(M1) Iterative follow-up	77 (33%)	163 (40%)
(M2) Reveal the initial task	62 (26%)	118 (29%)
(M3) Refine prompt	40 (17%)	56 (14%)
(M4) Information giving	18 (8%)	24 (6%)
(M5) Reveal a new task	16 (7%)	15 (4%)
(M6) Negative feedback	13 (6%)	8 (2%)
(M7) Asking for clarification	9 (4%)	19 (5%)
Others	1	6

Key Findings: RQ2 - What are the flow patterns in multi-turn conversations?

- Typical start: Initial task in first prompt for 81% (PRs) / 90% (Issues); sometimes given in second prompt (PR ~13%, Issues ~3%).
- Six prevalent flows:
  - Start → Initial task → Iterative → End
  - Start → Initial task → Refine → (Iterative) → End
  - Start → Initial task → New task → End
  - Start → Info giving → Initial task → ... → End
  - Start → Initial task → Clarification → End
  - Start → Initial task → Negative feedback → End.

Key Findings: RQ3 - Why do developers share ChatGPT links?

- Top three rationales:
  - P2 Potential solution: Issues 53%; PRs 33%
  - P1 Source of implemented solution: PRs 37%; Issues 22%
  - P3 Support a claim: PRs 24%; Issues 17%
- Less frequent: Answer a question (P4), Illustrate example (P5); some Direct link / Others due to missing context.
- Sampling for RQ3: stratified 250 cases (PR 90, Issues 160).

Category	PR	Issue
With clear purpose:	84 (100%)	139 (100%)
(P1) Reference to a source of solution	31 (37%)	30 (22%)
(P2) Reference to a potential solution	28 (33%)	74 (53%)
(P3) Support a claim	20 (24%)	23 (17%)
(P4) Answer a question	3 (4%)	4 (3%)
(P5) Illustrate an example	2 (2%)	8 (6%)
Direct link	1	15
Others	5	6

Key Findings: RQ3 - Where are links posted & Who shares the links?

- Where?
  - PRs (n=85): Code-review comments 35%, PR comments 34%, PR description 31%.
  - Issues (n=154): Issue comments 63%, Issue description 36%, Title 1%.
- Who?
  - PRs: Authors 53%, Reviewers 47%.
  - Issues: Authors 71%, Collaborators 25%, Assignees 4%.

Positives – Contribution & Rigor

- Novel observational lens on collaboration around LLMs inside PR/issue workflows.
- Clear, reusable taxonomies (16 initial-prompt types; 7 prompt roles).
- Inter-rater agreement reported (substantial to almost perfect).

Positives – Practical Value

- Actionable: Where links appear and who shares informs PR/issue UX and team practices.
- Flow patterns suggest product features (e.g., assist next-turn prompts; provenance surfaces).
- Breadth beyond code generation reflects realistic developer needs.

Negatives / Threats to Validity

- External validity: limited to GitHub PRs/issues and a specific time window of collected links.
- Construct/internal validity: open coding introduces subjectivity despite strong kappa agreements.

Suggested Future Work

- Benchmarks that match reality: Include multi-turn settings and diverse inputs (e.g., error traces without code, code-context generation) to evaluate FMs.
- Role-aware tooling: Surfaces for reviewers vs authors (e.g., attach “chat provenance” to code suggestions).
- Prompt-flow guidance: Leverage observed flow patterns to recommend effective next-turn prompts.

My Rating & Other Comments

- Overall rating: 4/5
  - Strong descriptive contribution and practical insights.
- Clarity in reporting inter-rater reliability and sampling; useful PR vs Issue breakdowns.
- Would be stronger with outcome measures and broader, fresher data.

Discussion Time!

- **Should PR templates add a “ChatGPT provenance” field?** Would that improve transparency or bias reviewers? (Links are already common in descriptions/comments.)
- **When is sharing a chat “evidence” vs “appeal to authority”?** We saw “support a claim” as a common rationale, but hallucinations are possible. What safeguards are needed?
- **Ethics & IP**: Are there cases where sharing the exact conversation is risky (e.g., proprietary snippets pasted into ChatGPT)? Is there anyway to prevent it?

UNIVERSITY OF WATERLOO

YOU+WATERLOO

Our greatest impact happens together.