UNDERSTAND DOMAINS IN SOFTWARE ENGINEER: AN EMPIRICAL STUDY ON AGENTIC PULL REQUESTS

10/25/28

Kevin Jie, Xiangrui Ke, Zhiheng Lyu

David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

BACKGROUND

- Agentic-PRs is an important part of open-source development.
- AI models are shaping today's software industry.
- AI code agent is hugely involved in software development

UNIVERSITY OF WATERLOO

PROBLEM

Software domains are not homogeneous: they differ in architecture, testing requirements, and review norms.

Existing studies treat all PRs as homogeneous, overlooking domain-specific factors that may influence their reception.

Lack of systematic understanding across domains

UNIVERSITY OF WATERLOO

PROBLEM

Core tags

dtype, io, regex, concurrency, security, numerical, ml, graph, parsing, file-system, serialization, memory, performance, networking, database, cli, logging, config, datetime, unicode, path, container, build, packaging, versioning, gpu, distributed, caching, algorithm, recursion, state-machine, auth, crypto, api, schema, SQL, noSQL, docstring, l10n, i18n, accessibility, visualization, instrumentation, telemetry, overflow, underflow, NaN, encoding, compression, rand, seed, os-compat, symlink, exception, test-infra

Extended tags

admission-control, audit, batch, cloud, code-generation, conformance, controller, custom-resource, dns, docker, extensibility, federation, ha, hw-accel, images-registry, ingress, ipv6, network-policy, node-lifecycle, observability, provider, release-eng, reliability, secret, swagger, usability, io-avro, io-cloud, io-csv, io-database, io-delta, io-iceberg, io-json, io-parquet, interchange, interop, grpc, rest, ci, cd, devops, rollback, scheduler, metrics, tracing, feature-flag, a11y-aria, spark-sql, spark-streaming, spark-mllib, spark-graphx, comp:core, comp:data, comp:eager, comp:api, comp:dist-strat, comp:gpu, comp:gpu:tensorrt, comp:tpu, comp:xla, comp:keras, comp:grappler, comp:optimizer, module:autograd, module:nn, module:jit, module:onnx, module:cuda, module:quantization, module:distributed, module:fp16, module:dataloader, module:hub, module:lite, component:dom, component:concurrent-features, component:fast-refresh, component:devtools, component:build-infrastructure, browser:ie, browser:safari

UNIVERSITY OF WATERLOO

Motivation

Better know the domains difference within the software development pipeline.

- To Identify where AI coding agents are most effective.
- Developer can gain feedback for domain-specific fine-tuning and improvement (e.g merge rate)
- An advancing responsible AI adoption in software engineering

Related Work & Novel Contribution

- Many works study how AI can help and revolutionize software engineering process (AI4SE).
  - *Code Review:*
    - *Automating Code Review Activities by Large-Scale Pre-training* (Li, et al, 2022)
  - DevOps:
    - *Explaining GitHub Actions Failures with Large Language Models: Challenges, Insights, and Limitations* (Valenzuela-Toledo, et al, 2025)
- Some works focus on Agentic-PRs
  - *Quality gatekeepers: investigating the effects of code review bots on pull request activities* (Wessel, et al, 2022)
    - It shows that automation can increase merge rates and alter collaboration patterns in open-source projects
  - *On the Use of Agentic Coding: An Empirical Study of Pull Requests on GitHub* (Watanebe, 2025)
    - It studies the usefulness of Agentic-PRs and the extent to which their contributions are accepted in real-world projects

Related Work & Novel Contribution

- The closest related work:
  - *On the Use of Agentic Coding: An Empirical Study of Pull Requests on GitHub* (Watanebe, 2025)
    - To what extent are Agentic-PRs rejected and why?
    - How do Agentic-PRs differ from Human PRs in terms of change size and purpose?
    - What proportion of Agentic-PRs are accepted without revisions? If needed, to what extent?
    - What changes are required to revise Agentic-PRs?
- We study form an orthogonal angle: the acceptance/rejection rate among different theme categories the codes belong to

Table 1. PR purposes based on analysis of PR content

Category	Description	% APRs	% HPRs	% Δ
fix	Code changes that fix bugs and faults within the codebase	31.0%	30.8%	0.2% ↑
feat	Code changes that introduce new features to the codebase, encompassing both internal and user-oriented features	26.8%	27.6%	0.8% ↓
refactor	Code restructuring without changing its behavior, aiming to improve maintainability	24.9%	14.9%	10% ↑
docs	Updates to documentation or comments, such as README edits, typo fixes, or API docs improvements	22.1%	14.0%	8.1% ↑
test	Additions or modifications to test files, including new test cases or updates to existing tests	18.8%	4.5%	14.3% ↑
build	Changes to build configurations (e.g., Maven, Gradle, Cargo). Change examples include updating dependencies, configuring build configurations, and adding scripts	10.8%	3.6%	7.2% ↑

Research Questions

RQ.1	RQ.2	RQ.3
How accurately can LLMs identify software domains of Agentic-PRs?	**Which domain exhibit higher Agentic-PR acceptance rates and shorter review times?**	**How do different AI agents behave across domains?**

Tools

We use **DeepSeek-V3** as the primary model for labeling **5,000 (more or less)** representative Agentic-PRs, chosen for its balance between cost and accuracy. A subset of 500 samples will be cross-validated using GPT-5 and Claude 4.5 Sonnet to estimate model consistency and bias.

Reasons to Use of LLM:

- **First**, the large sample size makes manual labeling difficult;
- **Second**, LLM is capable of handling complex classification tasks;
- **Third**, because LLM is pre-trained and has strong performance, we do not need to train a new model.

While an alternative to LLM is to use a traditional machine learning model for labeling (classification task), LLM remains the best choice.

Tools

Fair Sampling Technique:

- **Qwen3 8B-Embedding** to encode commit messages, PR titles, and diffs.
- These embeddings support semantic clustering, retrieval-based few-shot prompting, and visualization of domain relationships.
- Selecting samples based on their embedding similarity rather than random choice or frequency

Tools

Table 1: Prompt: Domain Classification

You are a senior software architect experienced across kernel, cloud-native, data, ML, and front-end stacks. Your job is to read the supplied **code diff** (and, if present, the related issue text) and decide which *technical domain(s)* the change touches.

Domain Tag Vocabulary
Use any tag from the list below **or create one that follows Rule 2**.

Core tags

`dtype, io, regex, concurrency, security, numerical, ml, graph, parsing, file-system, serialization, memory, performance, networking, database, cli, logging, config, datetime, unicode, path, container, build, packaging, versioning, gpu, distributed, caching, algorithm, recursion, state-machine, auth, crypto, api, schema, SQL, noSQL, docstring, l10n, i18n, accessibility, visualization, instrumentation, telemetry, overflow, underflow, NaN, encoding, compression, rand, seed, os-compat, symlink, exception, test-infra`

Extended tags

`admission-control, audit, batch, cloud, code-generation, conformance, controller, custom-resource, dns, docker, extensibility, federation, ha, hw-accel, images-registry, ingress, ipv6, network-policy, node-lifecycle, observability, provider, release-eng, reliability, secret, swagger, usability, io-avro, io-cloud, io-csv, io-database, io-delta, io-iceberg, io-json, io-parquet, interchange, interop, grpc, rest, ci, cd, devops, rollback, scheduler, metrics, tracing, feature-flag, a11y-aria, spark-sql, spark-streaming, spark-mllib, spark-graphx, comp:core, comp:data, comp:eager, comp:api, comp:dist-strat, comp:gpu, comp:gpu:tensorrt, comp:tpu, comp:xla, comp:keras, comp:grappler, comp:optimizer, module:autograd, module:nn, module:jit, module:onnx, module:cuda, module:quantization, module:distributed, module:fp16, module:dataloader, module:hub, module:lite, component:dom, component:concurrent-features, component:fast-refresh, component:devtools, component:build-infrastructure, browser:ie, browser:safari`

Rule 2 - Custom Tags If no existing tag fits, you may invent **one custom tag**. Custom tags must start with X_, contain 3 English words, and must not conflict semantically with the vocabulary above.

Output Guidelines

1. Return **1–3 tags** (comma-separated, no spaces, no quotes).
2. Choose the *most specific* tags; if both a parent and child fit, choose the child.
3. If multiple domains apply, list up to three in relevance order.
4. Provide **no explanations**.

Code Changes
`{code_changes}`
Issue Body
`{issue_body}`

Plan

To answer RQ1,

- Manually label a sample subset from the original dataset as a validation set
- Annotate the domains each Agentic-PR belongs to, use DeepSeek-V3 to classify the same dataset and measure its performance (by precision, recall, and F1 scores)

To answer RQ2,

- Heuristically select a subset of the AgenticPRs as our main study scope
- Whenever possible, PRs will be paired within the same repository but with different acceptance outcomes to support comparative analysis.
- Use DeepSeek-V3 for large-scale labeling and cross-validate with GPT-5 and Claude 4.5 Sonnet to estimate model bias and consistency

Plan

To answer RQ3,

- Analyze the correlation between PR features and outcomes, such as acceptance rate, review latency, comment density.
- A mixed-effects regression model will be used to quantify the effects of domain and agent identity on PR success.

Threats

Sample Bias

- The selected subset of PRs may overrepresent certain repositories or domains. We will perform stratified sampling to balance domain frequency and repository activity.

Manually Labelled Data

- Human labels may vary due to subjective interpretation. Peer review and consensus checking will be applied to improve reliability.

LLM-Assisted Labelling

- Model-based labeling may reflect training-data bias. Cross-validation with multiple models (such as DeepSeek-V3, GPT-5, Claude 4.5) will be used to estimate bias and stability.

DISCUSSION

Reference

Mairieli Wessel, Alexander Serebrenik, Igor Wiese, Igor Steinmacher, and Marco A. Gerosa. Quality gatekeepers: investigating the effects of code review bots on pull request activities. Empirical Softw. Engg., 27(5), September 2022.

Miku Watanabe, Hao Li, Yutaro Kashiwa, Brittany Reid, Hajimu Iida, and Ahmed E. Hassan. On the use of agentic coding: An empirical study of pull requests on github, 2025.

Li, Z., Lu, S., Guo, D., Duan, N., Shailesh Jannu, Jenks, G., Majumder, D., Green, J., Alexey Svyatkovskiy, Fu, S. and Sundaresan, N. 2022. Automating code review activities by large-scale pre-training. Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. (Nov. 2022). DOI:https://doi.org/10.1145/3540250.3549081.

Explaining GitHub Actions Failures with Large Language Models: Challenges, Insights, and Limitations: 2025. https://arxiv.org/abs/2501.16495. Accessed: 2025-09-22.