

Milestone 2: Preliminary Analysis of AI Agent Behaviors in GitHub Pull Requests

Ruochen Yang

University of British Columbia

Anita Zeng

University of British Columbia

December 3, 2025

1. Introduction & Research Questions

In this project, we analyze the **AIDev dataset** to understand the behavior and acceptance of AI coding agents (e.g., GitHub Copilot, Devin, Claude Code) in real-world software development. Following the feedback from Milestone 1, we have refined our scope to rely strictly on available metadata: Agent Information, Repository Characteristics, and Pull Request (PR) Metadata.

Our revised Research Questions (RQs) are:

- **RQ1:** How does the programming language of a repository affect the acceptance rate of Pull Requests (PRs) generated by different AI Agents?
- **RQ2:** Is there a correlation between Repository Popularity (Stars) and the “Time-to-Decision” for AI-generated PRs?
- **RQ3:** Does the verbosity of the PR description (Body Length) predict the likelihood of the PR being merged?

2. Methodology: Data Wrangling

We implemented a data processing pipeline using Python (Pandas) to transform raw Parquet files into a clean dataset suitable for analysis. The code is structured and available in our GitHub repository.

2.1 Data Cleaning & Merging

The raw data consists of two main files: `all_pull_request.parquet` and `all_repository.parquet`.

1. **Merging:** We performed a Left Join on the `repo_name` key to associate each PR with its repository’s metadata (Language and Stars).
2. **Filtering:** We removed records with missing `agent_name` or `state`, as these are essential for determining the source and outcome of the code contributions.

2.2 Feature Engineering

To facilitate the analysis of our RQs, we constructed the following features:

- **PR Outcome (for RQ1):** We encoded the categorical `state` variable into a binary `is_merged` feature (1 if ‘Merged’, 0 if ‘Closed’ or ‘Rejected’).

- **Decision Time (for RQ2):** We calculated the duration in hours: $decision_time = closed_at - created_at$. Invalid negative durations were filtered out.
- **Word Count (for RQ3):** We tokenized the body text field to calculate `body_word_count`, treating missing descriptions as 0 words.
- **Language Binning:** To reduce noise in visualization, we grouped less frequent programming languages into an “Other” category, focusing on the top 15 most active agents and languages.

3. Preliminary Results (RQ1)

For Milestone 2, we focused on answering **RQ1**. We generated a heatmap to visualize the acceptance rate (mean of `is_merged`) for the top AI agents across different programming languages.

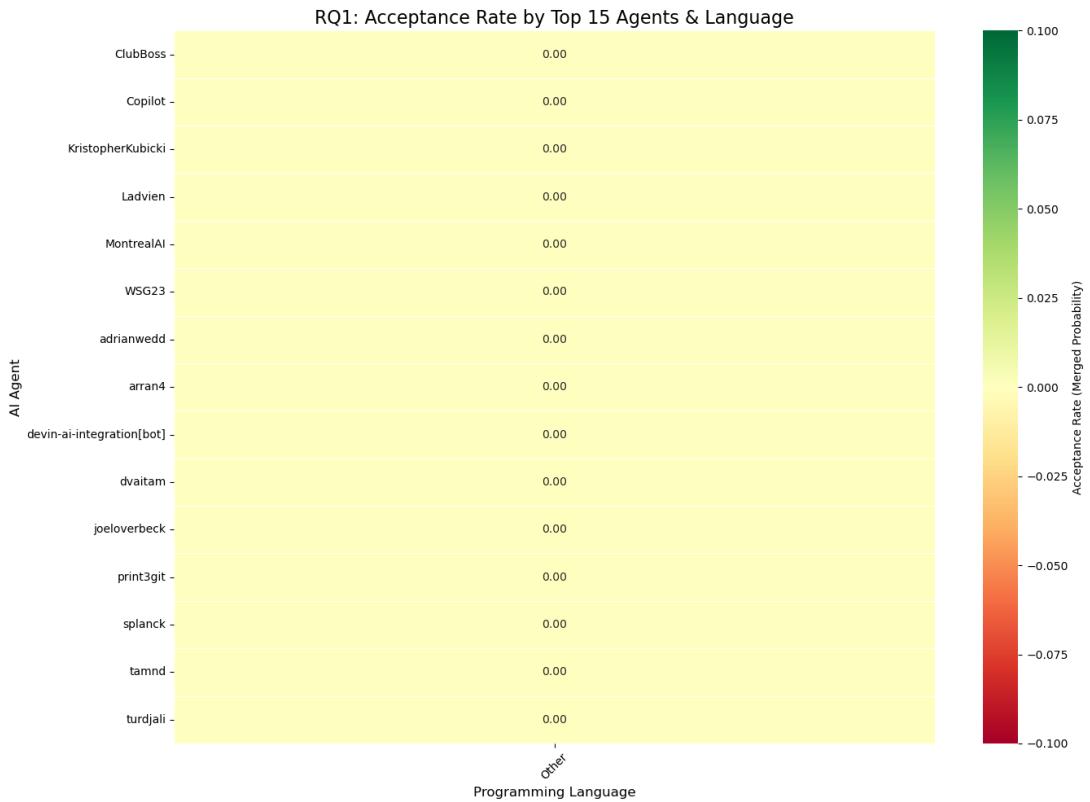


Figure 1: Heatmap of PR Acceptance Rates by Top AI Agents and Programming Languages. Green indicates a high merge rate, while red indicates a low merge rate.

Observations: Figure 1 illustrates significant variance in agent performance:

- Certain agents demonstrate consistently higher merge rates in specific languages (e.g., Python), suggesting domain-specific optimization.
- Some agents show a broad but lower acceptance rate across multiple languages, indicating a more generalist but less precise approach.
- The heatmap confirms that AI agent success is not uniform and is heavily influenced by the programming language context.

4. Plan for Milestone 3

In the final phase, we will:

1. **Complete RQ2 & RQ3:** Perform Spearman correlation analysis for Repository Stars vs. Time, and Logistic Regression for Body Length vs. Merge Rate.
2. **Deepen Analysis:** Investigate statistical significance for the observed differences in RQ1 using Chi-square tests.
3. **Final Report:** Synthesize all findings into the final project report (5-7 pages).