Ghosting in the Machine: Predicting Wasted Review Effort in AI-Generated Pull Requests

Anonymous Author(s)

Abstract

The emergence of autonomous coding agents has introduced a new dynamic in software engineering: "AI Teammates" that independently author Pull Requests (PRs). While promising, these agents introduce unique risks, particularly "ghosting"—abandonment after feedback. In this study, we analyze 33,596 Agentic-PRs from the AIDev dataset to characterize this phenomenon. We identify two distinct regimes: "Instant Merges" (32%) which are narrow-scope updates (median 68 lines), and "Normal PRs" where agents face genuine complexity. Our LightGBM models achieve an AUC of 0.84 for identifying high-cost PRs, outperforming a text baseline (AUC 0.57) and generalizing across unseen agents (LOAO AUC 0.66–0.80). Furthermore, we demonstrate that triage policies prioritizing the top 20% of risky PRs can capture 47.4% of total review effort on a repo-disjoint test set. These findings emphasize the importance of structural signals in automated triage and propose actionable human-in-the-loop workflows to mitigate the hidden costs of AI collaboration.

CCS Concepts

• **Software and its engineering → Software evolution**.

Keywords

AI Agents, Triage, Ghosting, Mining Software Repositories

1 Introduction

In the rapidly evolving landscape of Modern Software Engineering, the role of Artificial Intelligence (AI) has shifted from passive assistance to active participation. The emergence of autonomous coding agents—"AI Teammates" capable of independently planning, coding, and submitting Pull Requests (PRs)—marks a paradigm shift in collaborative development [? ? ? ? ? ? ? ?]. Tools like GitHub Copilot Workspace, Devin, and OpenHands promise to accelerate development cycles and reduce the burden of mundane tasks [? ? ?]. However, this autonomy introduces new friction points in the human-AI workflow. Unlike human contributors, who typically adhere to social norms of communication and stewardship [? ? ? ?], early autonomous agents often exhibit erratic follow-through behavior, a phenomenon we term "Ghosting."

Ghosting occurs when an agent submits a PR but fails to respond to human feedback or CI failures, effectively abandoning the contribution. This behavior imposes a significant "Hidden Cost" on open-source maintainers, who must invest time reviewing code, understanding intent, and providing feedback, only to have that effort wasted [? ?]. As Agentic-PRs become ubiquitous, the risk of a "Denial-of-Service" attack on maintainer attention becomes acute. Existing research on Pull Request triage has largely focused on human-centric metrics (e.g., social reputation, prior contributions) [? ? ? ? ? ? ?]. However, AI agents lack social accountability and operate under different constraints—often prioritizing speed and volume over correctness or maintainability [? ? ?]. There is a critical lack of empirical understanding regarding how these agents behave in the wild and what signals predict their reliability.

To address this gap, we present a comprehensive study of 33,596 PRs authored by five prominent AI agents (Claude, Copilot, Cursor, Devin, Codex) from the AIDev dataset [?]. We aim to operationalize the concept of "Agentic Ghosting" and develop predictive mechanisms to triage high-risk contributions before they consume scarce reviewer resources [? ? ? ? ? ?]. Specifically, we investigate:

- **RQ1 (Predictability)**: To what extent can we rely on submission-time signals to predict which Agentic-PRs will incur high review costs or be abandoned?
- **RQ2 (Risk Factors)**: What behavioral and structural cues—such as file complexity or interaction patterns—signal a higher propensity for ghosting?

Our contributions are threefold:

(1) **Operationalization of Ghosting**: We establish a rigorous definition of "True Ghosting" (abandonment after human feedback) and validate it through a manual audit, finding a concerning 64.5% ghosting rate in rejected PRs.
(2) **Predictive Triage Framework**: We propose a LightGBM-based model utilizing 35 features extracted from the initial PR snapshot. Our model achieves an AUC of 0.84 in identifying high-cost PRs, significantly outperforming text-based baselines (AUC 0.57) and demonstrating robustness across unseen agents (LOAO AUC 0.66–0.80).
(3) **Empirical Insights**: We uncover a "Two-Regime" distribution where 32% of agent PRs are "Instant Merges" (trivial updates), while the remaining "Normal Workflow" PRs pose genuine triage challenges. Furthermore, we reveal a counter-intuitive "Interactive Complexity" effect where CI-touching PRs are actually less likely to be ghosted, identifying a key mechanism for human-in-the-loop control.

2 Methodology

2.1 Dataset Curation

We utilize the AIDev dataset [?], a curated collection of fully autonomous PRs. We filtered the dataset to focus on the top five most active agents to ensure statistical significance: Claude, Copilot, Cursor, Devin, and Codex. The final corpus consists of 33,596 PRs. To ensure the validity of our "Ghosting" label, we excluded PRs that were merged without any human interaction or rejected immediately without feedback, isolating the pool where "abandonment" is a meaningful concept. This filtering aligns with best practices in mining software repositories to reduce noise [? ? ? ?]. We also define "Instant Merges" ($<$ 1 min turnaround) as a separate regime from behavioral analysis to avoid skewing latency metrics [?].