Designing Fair and Scalable AI-Enhanced Software Engineering Performance Reviews

Aishwarya Babu

babu.aishwarya@gmail.com

Abstract

Performance evaluations in software engineering often struggle with fairness and consistency, particularly in capturing non-code contributions like mentorship and technical leadership. While AI and code analytics offer objectivity and scalability, their over-reliance can reduce nuance. This paper proposes a hybrid framework that integrates AI and natural language processing (NLP) to map traditionally qualitative contributions—such as mentorship impact and design complexity—into measurable signals. By blending conventional code metrics with inferred collaborative behaviors, we introduce a composite metric system designed to ensure fairness across diverse engineering roles and management styles.

Keywords: Performance reviews, software engineering, artificial intelligence, performance metrics, mentorship, code analytics

I. INTRODUCTION

Performance reviews are critical in shaping career progression, compensation, and team dynamics in software engineering organizations. Traditional evaluation methods have struggled to maintain fairness, consistency, and transparency, especially across large-scale, distributed teams. Manual reviews are time-consuming and often reflect inherent biases, such as favoring visibility and charisma over behind-the-scenes contributions.

In recent years, there has been a shift toward data-driven performance reviews powered by code analytics and AI. Despite these advancements, most organizations have yet to strike a balanced approach. The industry continues to grapple with the tension between under-reliance on metrics (subjective, non-replicable feedback) and over-reliance (quantitative metrics that fail to capture nuance).

This paper presents a unique hybrid framework that combines traditional code metrics with AI-derived signals to offer a more holistic, transparent, and fair review process, addressing gaps in current practices. Our contributions include:

- Proposing a composite metric that incorporates both quantitative and qualitative signals.
- Demonstrating how AI and NLP can help surface under-recognized contributions such as mentorship and documentation.
- Highlighting challenges and ethical considerations associated with AI-based assessments.

II. RELATED WORK

Recent literature has explored various facets of AI in performance evaluations. In [1], AI is positioned as a tool for redefining traditional appraisals. Another study [2] emphasizes analytics for evaluating development and reward programs. Platforms such as Pluralsight Flow [5], Haystack, Waydev [4], and CodeScene [6] operationalize metrics like pull request (PR) velocity, deployment frequency, and bug resolution time to

evaluate engineering performance. DORA metrics [3] are widely adopted for assessing delivery performance. NLP techniques are increasingly applied to analyze sentiment in peer feedback, as well as detect communication trends and leadership signals.

However, these tools often miss softer, high-impact contributions — such as mentorship and strategic architectural decisions — due to their qualitative nature. Moreover, research has shown that introducing user-in-the-loop mechanisms — where humans can adjust, annotate, or challenge AI-derived inferences — helps create more trusted and accurate decision-making systems.

In a real-world example, a mid-sized technology company using GitPrime (now Pluralsight Flow) noted that developers began optimizing for PR volume, causing a decline in code quality and increased technical debt [2]. Similarly, a case reported by Waydev highlighted how cultural misinterpretation by sentiment analysis tools led to misjudged peer feedback [1].

These findings highlight the need for a balanced approach that incorporates ethical design, human oversight, and a nuanced understanding of engineering impact.

III. METHODOLOGY

Our methodology combines **existing techniques** with **a proposed AI-driven framework** to enhance fairness and recognition in performance evaluation.

A. Existing Methodologies

1. **Code Quality & Contribution Metrics**: Static analysis and Git-based metrics such as pull request volume, code churn, bug fix ratio, and test coverage are standard in tools like CodeScene [6] and Waydev [4]. These help quantify individual contributions in terms of productivity and code health.
2. **Productivity & Delivery Metrics**: DORA metrics [3]—Deployment Frequency, Lead Time for Changes, Change Failure Rate, and Mean Time to Recovery—are industry-standard for tracking delivery efficiency. Git analytics platforms like Pluralsight Flow [5] extend this with cycle time and PR review participation.
3. **Collaboration & Sentiment Analysis**: NLP techniques are applied to PR comments, documentation, chat logs, and performance reviews to detect sentiment, collaboration tone, and feedback quality. These techniques are discussed in [1], [2].

B. Proposed AI-Enhanced Framework

Building on these existing tools, we propose a composite metric system that integrates the following AI-driven signals:

1. **Mentorship Impact Detection**: Use NLP to analyze Slack/Jira/GitHub interactions between mentor and mentee, tracking behavioral and performance deltas over time. Example indicators include frequency and specificity of code feedback, topic modeling of discussion threads, and mentee progression rate.
2. **Technical Communication Recognition**: Classify design document authorship using large language models (LLMs) to evaluate complexity, originality, and influence. Embedding-based clustering techniques can be used to compare technical proposals over time and across teams.
3. **Resilience Signals**: Identify patterns in how engineers respond after production outages or negative code reviews, using log parsing and sentiment delta analysis. We look for recovery timelines, follow-up PRs, and documented retrospectives as positive indicators.

4. **Technical Foresight Indicators**: Analyze PRs and design artifacts for signs of modularity, backward compatibility, and architectural forward-thinking using semantic embeddings and AI summarization. These are quantified using temporal model predictions on refactorings.
5. **Influence Across Teams**: Track adoption of concepts, tools, or documentation patterns initiated by an engineer. Use graph-based propagation models to visualize how contributions are reused or adapted cross-functionally.
6. **Role-Contextual Adjustments**: Use transformer-based models to synthesize official role guidelines and manager-provided inputs to dynamically calibrate expectations. For instance, expectations for an L5 engineer may focus more on individual delivery, while L7 roles emphasize cross-team influence.
7. **Engineer Feedback Loops**: Introduce an interface for engineers to interact with and annotate AI-generated summaries of their performance. Active feedback becomes part of the training data, increasing transparency and mitigating hallucinations.

IV. IMPLEMENTATION FRAMEWORK

We propose a four-layer implementation framework:

1. **Data Ingestion Layer**: Integrates with platforms like GitHub, Jira, Confluence, Slack, and internal performance systems. Includes ETL pipelines to extract and normalize signals across structured and unstructured data.
2. **Signal Processing & Feature Engineering**: Applies natural language processing, semantic embeddings, and graph analytics to extract behavioral and impact-oriented features. For example, LLMs classify technical writing tone, and graph-based models assess influence propagation.
3. **Composite Metric Engine**: Combines traditional code metrics with AI-inferred qualitative features. Each engineer's profile is compared against role-calibrated expectations. Scoring includes customizable weights by company or team values.
4. **Human-in-the-Loop Feedback Module**: Allows engineers and managers to review, annotate, and challenge auto-generated insights. Supports contextual explanations and revision histories to build trust in AI outputs.

This modular architecture supports phased adoption, starting with metric augmentation and expanding toward full-spectrum AI assistance. Organizations can incrementally adjust transparency, feedback control, and ethical guardrails

V. CHALLENGES AND ETHICAL CONSIDERATIONS

Adopting AI in performance evaluations raises important challenges:

- **Bias & Fairness**: As seen in the Waydev deployment case [1], sentiment models misjudged feedback tone from non-native speakers. Adjustments were needed to accommodate linguistic diversity.
- **Interpretability**: Pluralsight Flow initially generated composite developer health scores without context, leading to distrust [2]. Adding narrative summaries improved trust.
- **Gaming & Misuse**: In GitPrime's case study, developers prioritized PR quantity over value [2]. This led to metric redesign and introduction of review quality measures.
- **Privacy & Consent**: It is essential to implement robust anonymization, opt-in mechanisms, and clear data usage policies to avoid backlash for data without employee consent.
- **Human Oversight**: To avoid over-reliance on AI-driven scoring tools, training modules can help reestablish human context as essential.

VI. BALANCING AI AND HUMAN JUDGMENT

AI-assisted evaluations promise scale and consistency but cannot—and should not—replace human discernment in performance reviews. Over-reliance on AI-generated scores can lead to erosion of trust, diminished context, and cultural misinterpretation. We advocate for a hybrid approach where AI enhances—but does not dominate—the review process.

We propose three guiding principles for effective integration of AI and human judgment:

1. **Companion Model**
   AI-generated insights should function as context-aware, explainable baselines—not as final verdicts. For example, when GitPrime's scoring tools began driving managerial decisions [2], engineers began optimizing for metrics at the cost of long-term code health. We recommend that AI serve as a diagnostic assistant, highlighting areas of impact or concern, while leaving room for human interpretation and override.
2. **Calibration Rounds Anchored in Context**
   Prior to formal review cycles, teams should hold calibration sessions where AI-derived outputs are reviewed alongside team-specific context. This practice mirrors how DORA metrics are contextualized within platform teams versus product teams. In a case from a cloud services company, this helped account for uneven on-call burdens and role ambiguity, preventing misattribution of low deployment frequency as underperformance.
3. **Empowered Human Oversight with Training**
   Managers and reviewers must be empowered—and trained—to identify where AI falls short. This includes recognizing outlier scenarios, interpreting AI-extracted narrative summaries, and correcting for potential biases.

The goal is not perfect objectivity, but better shared understanding. When deployed responsibly, AI can bring hidden contributions into focus, enabling human reviewers to ask better questions and arrive at more nuanced conclusions.

VIII. CONCLUSION AND FUTURE WORK

While this framework addresses a broad range of challenges in performance evaluations, several areas warrant further exploration:

- Investigating the longitudinal effects of AI-generated insights on promotions and retention.
- Expanding mentorship impact detection to asynchronous mentorship models.
- Enhancing explainability of LLM-derived metrics to support review calibration.

By capturing engineering excellence beyond code commits, this framework aspires to democratize recognition and provide richer, more actionable performance insights. AI can serve as a partner—not a judge—helping organizations uphold fairness, transparency, and nuance in evaluating engineering talent.

REFERENCES

[1] M. Ashraf and A. Afzal, “AI in performance management: Redefining performance appraisals in the digital age,” *ResearchGate*, 2023. [Online]. Available: https://www.researchgate.net/publication/376135128
[2] S. Kim and P. Johnson, “Performance Management Analytics: Using AI to Analyze Employee Performance Data and Inform Development and Rewards Programs,” *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/384932994
[3] DORA Metrics, Google Cloud. [Online]. Available: https://cloud.google.com/devops/state-of-devops
[4] Waydev, Git Analytics for Engineering Leaders. [Online]. Available: https://waydev.co/

[5] Pluralsight Flow. [Online]. Available: https://www.pluralsight.com/flow
[6] CodeScene. [Online]. Available: https://codescene.com