Track Selected

Track 1 – Agentic AI (Applied GenAI)

Title

Live Bug Report Analyzer & Prioritization Agent for Small Development Teams

Problem Statement

Modern software teams receive a continuous flow of bug reports from multiple sources such as issue trackers, emails, customer feedback tools, and internal testing platforms. These bug reports are not static; they evolve over time as developers add logs, screenshots, comments, and reproduction steps. In many teams, bug triage is performed manually at fixed intervals, which results in outdated prioritization, missed critical issues, duplicated work, and delayed responses to high-impact bugs.

Existing tools treat bug reports as static entries rather than living documents. As a result, teams often rely on stale information when making prioritization decisions. This creates a gap where important changes in a bug's severity or impact go unnoticed until it is too late.

Proposed Agentic AI Solution

We propose an agentic AI system that continuously monitors bug reports and their associated updates in real time. The system acts as an intelligent triage assistant that maintains contextual memory for each bug and dynamically reassesses its severity and priority whenever new information is added.

Unlike traditional automation pipelines, the agent does not operate in batch mode. Instead, it reacts instantly to incoming updates, reasons over the latest context using an LLM, and updates bug priority labels and summaries accordingly. This ensures that developers always work with the most up-to-date understanding of each issue.

System Architecture Overview

The system is built using Pathway's streaming-first architecture. Bug reports, comments, and attachments are treated as live data streams rather than static inputs. Pathway's engine incrementally processes updates and maintains a stateful representation of each bug report.

An LLM-based reasoning component is invoked whenever meaningful changes occur. The agent evaluates factors such as severity, user impact, duplication signals, and frequency of updates before deciding whether the priority of a bug should be adjusted.

High-Level Architecture Flow

Bug Tracker / Document Stream → Pathway Engine → Live Indexing & Stateful Memory → LLM Reasoning Module → Priority Update & Developer Notification

Agent Flow

1. A new bug report or update (comment, log, screenshot) is submitted. 2. Pathway ingests only the changed data and updates the live index. 3. The agent retrieves historical context for the affected bug. 4. The LLM reasons about severity, urgency, and potential duplication. 5. The agent updates the bug’s priority and generates a concise summary for developers.

Why Pathway

Pathway is well-suited for this system due to its ability to handle continuous data streams and stateful computation. Unlike traditional RAG pipelines that rely on static embeddings, Pathway ensures that the agent always reasons over the most recent version of a document.

This makes it possible to build AI agents that react immediately to changing inputs and maintain long-running context, which is essential for live developer workflows.

Impact

The proposed system reduces manual triage effort for development teams and ensures that critical bugs are identified and addressed promptly. By keeping prioritization aligned with the latest information, teams can respond faster to high-impact issues while avoiding unnecessary work on low-priority bugs.

Future Enhancements

Future improvements could include deeper memory integration to learn from past bug resolutions, improved duplicate detection across projects, and tighter integration with development tools. Advanced memory-driven architectures such as Dragon Hatchling could further enhance continuous learning and long-term context handling.