

Agentic PR Risk Auditor

AI Hackathon San Francisco x Hamburg

By Acme

Date
December 2025

The Team



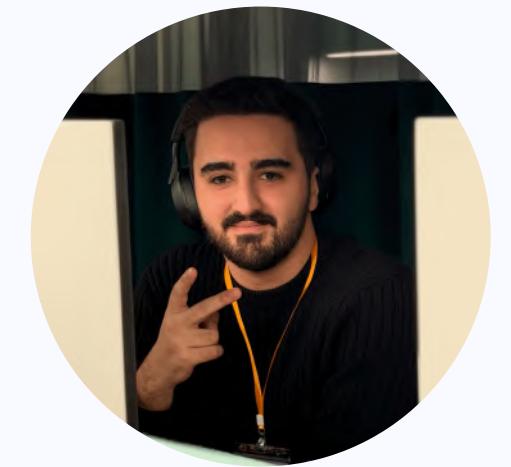
Maiya

Front-End Developer



Amir

Developer



Danial

Front-End Developer



AIRBUS

C>ONSTRUCTOR
UNIVERSITY





Problem

Modern software teams handle **dozens to hundreds of pull requests daily.**

Some PRs introduce **hidden, high-impact risks:**

- Authentication & authorization bugs
- Database integrity issues
- Security regressions
- Broken assumptions not covered by tests

Manual review does not scale:

- Reviewers miss edge cases
- Tests are missing, weak, or outdated
- Risk assessment is subjective and inconsistent
- Running AI generated tests can introduce new risks

PR Risk Intelligence

Prioritize risk • explain why • fix first

The screenshot shows a dark-themed dashboard for managing pull requests. At the top, there's a navigation bar with 'Pull Requests', a checkbox for 'High Risk only', a 'Sort' dropdown set to 'Risk ↓', and a search bar. Below this, three pull requests are listed:

- Fix auth bypass in login**: Risk Score: 78% (High Risk), Confidence: 40%. Status: **X 1**, **⚠ 1**, **✓ 2**. A progress bar is mostly green.
- Optimize query + caching layer**: Risk Score: 66% (Medium Risk), Confidence: 52%. Status: **X 0**, **⚠ 2**, **✓ 1**. A progress bar is mostly yellow.
- Refactor dashboard UI components**: Risk Score: 28% (Low Risk), Confidence: 66%. Status: **X 0**, **⚠ 1**, **✓ 2**. A progress bar is mostly green.

Below these, a specific pull request for "Fix auth bypass in login" is expanded:

- Why this PR is High Risk**:
 - Fails critical check: "SQL Injection test" (85%)
 - Touches authentication/session logic
 - Interacts with data layer / query logic
- Fix this first**:
 - ✗ SQL Injection test**: Potential unsafe query detected in data access layer. Risk: 85% (High Risk).
 - ⚠ Auth edge case test**: Token expiry edge case not fully covered. Risk: 55% (Medium Risk).
- Generated Tests**:
 - Expired JWT validation**: Reason: Auth middleware did not handle expired tokens. Risk: 85% (High Risk).
 - Malformed payload test**: Reason: Missing schema validation for login payload. Risk: 55% (Medium Risk).
- Errors**: **✗ SQL Injection test**: Potential unsafe query detected in data access layer. Risk: 85% (High Risk).
- Warnings**: **⚠ Auth edge case test**: Token expiry edge case not fully covered. Risk: 55% (Medium Risk).
- Passed Tests**: **✓ Login happy path**: Basic login flow passed. Risk: 0% (Low Risk).

At the bottom, there are buttons for "Open PR →" and "Try Pitch".

Solution

An **Agentic PR Risk Auditor** that enforces **evidence-based validation before merge** with CodeRabbit and Daytona technology.

The system **automatically**:

- Analyzes what a PR actually changes
- Identifies its risk surface
- Generates targeted tests
- Runs the tests in an isolated Daytona sandbox
- Summarizes risk clearly for reviewers

Value

- ✓ Safer merges**
- ⚡ Faster reviews**
- 📊 Objective, consistent risk assessment**
- 🔒 Security & correctness enforced **before** production**