perplexity

Product Requirements Document: AI-Powered PR Management System

I've created a comprehensive 31-page Product Requirements Document for your AI-powered GitHub PR management application. This document is ready to send to your engineering team (Cursor) to start development.

Key Highlights of the PRD:

Strategic Positioning

Your idea targets a **real and critical problem**: 45% of open source organizations cite maintainer burnout as their #1 challenge in 2025. However, the market is crowded with AI code reviewers. This PRD positions your product to differentiate by focusing on **PR triage, workflow automation, and maintainer wellbeing** rather than just code quality review.[1] [2] [3]

Core Differentiation from Competitors

Unlike GitHub Copilot, CodeRabbit, and PR-Agent which focus on code review quality, your tool will:

- **Intelligently triage and prioritize PRs** with 6-category classification system
- **Route PRs to the right reviewers** based on expertise, workload, and availability
- **Detect and prevent maintainer burnout** through proactive monitoring
- **Provide full-repository context** via RAG (Retrieval-Augmented Generation)
- **Support progressive review workflows** allowing partial PR merges

Technical Architecture Specifications

Multi-Agent System:

- Coordinator agent orchestrating 5 specialized sub-agents
- Uses Claude 3.7 Sonnet (200K context) for main analysis
- GPT-4o Mini for fast classification tasks
- Asynchronous parallel processing to reduce latency[4] [5]

RAG Implementation:

- Vector database: Pinecone (managed) or Weaviate (self-hosted)[6] [7]
- 768-dimension embeddings using CodeBERT

- Hybrid search (vector + full-text) for context retrieval
- Weekly re-indexing with incremental updates on push events

Technology Stack:

- Backend: Python 3.11+ with FastAPI
- Database: PostgreSQL 15+ for metadata, Redis for caching
- Infrastructure: Docker + Kubernetes for scalable deployment
- GitHub App with JWT authentication[8] [9]

Feature Set Breakdown

Phase 1 MVP (Months 1-3):

- PR classification into 6 categories
- Priority scoring (0-100 based on 6 weighted factors)
- GitHub bot command interface (`@PRCoPilot /triage`)
- Basic RAG context retrieval
- Daily digest emails

Phase 2 Intelligence (Months 4-6):

- Multi-agent architecture
- Enhanced full-repo RAG context
- Smart reviewer routing
- Bulk actions for maintainers

Phase 3 Wellbeing (Months 7-9):

- Burnout detection and prevention system
- Progressive review workflow (partial PR merges)
- Advanced analytics dashboard
- Cross-repository intelligence

Phase 4 Enterprise (Months 10-12):

- Custom agent training per repository
- Compliance reporting
- API for third-party integrations
- On-premise deployment packages

Business Model

```
Free Tier: 75 PR analyses/month (open source)
Pro Tier: $30/user/month (unlimited)
Enterprise: Custom pricing for >100 users
```

Target: 1,000 active repositories by month 6, $50K+ MRR by month 12

Security & Privacy

- **Zero data retention** commitment (code analyzed in-memory only)
- Self-hosted option available for strict data residency requirements
- SOC 2, GDPR, CCPA compliant
- Users provide their own LLM API keys (OpenAI, Anthropic)

Risk Mitigation

The document addresses 7 major risks including:

- LLM hallucinations → Confidence thresholds + human validation
- Competition from GitHub Copilot → Focus on differentiated triage features
- Maintainer skepticism of AI → Transparent reasoning + "co-pilot" positioning

What Makes This Different from Existing Tools

The competitive analysis shows that while GitHub Copilot ($39/mo), CodeRabbit ($12-24/mo), and PR-Agent (free) focus heavily on **code review quality**, your tool is the **only one focused on maintainer workflow optimization**:

- ✅ **PR Triage & Prioritization** (competitors: ❌)
- ✅ **Burnout Prevention** (competitors: ❌)
- ✅ **Smart Reviewer Routing** (competitors: ❌)
- ✅ **Multi-Agent Architecture** (competitors: ❌)
- ✅ **Full-Repo RAG Context** (competitors: PR-diff only)

Ready for Development

The PRD includes everything Cursor needs to start building:

- Database schemas (PostgreSQL tables)
- API endpoint specifications
- GitHub App permissions manifest
- Prompt engineering templates
- Infrastructure deployment guides (Docker, Kubernetes)

- Complete tech stack recommendations with specific versions

The document is structured for iterative development with clear success metrics for each phase and specific milestones to validate product-market fit before scaling.

Your next step: Share this PRD with your engineering team and schedule a kickoff meeting to align on Phase 1 MVP scope and timeline. The document recommends starting with 100 beta testers from mid-size open source projects (1K-10K stars) in Python/JavaScript ecosystems.

⁂

1. https://www.openlogic.com/system/files/2025-05/report-openlogic-2025-state-of-open-source-support.pdf
2. https://opensource.org/blog/key-insights-from-the-2025-state-of-open-source-report
3. https://www.linkedin.com/pulse/maintainer-burnout-real-why-open-source-stagnating-peter-smulovics-9bgze
4. https://cloud.google.com/architecture/multiagent-ai-system
5. https://www.anthropic.com/engineering/built-multi-agent-research-system
6. https://aloa.co/ai/comparisons/vector-database-comparison/pinecone-vs-weaviate
7. https://xenoss.io/blog/vector-database-comparison-pinecone-qdrant-weaviate
8. https://docs.github.com/en/apps/creating-github-apps/registering-a-github-app/using-webhooks-with-github-apps
9. https://notes.kodekloud.com/docs/AZ-400/Design-and-Implement-Authentication-and-Authorization-Methods/Implement-and-manage-GitHub-Authentication
10. https://docs.github.com/en/rest/apps/webhooks
11. https://www.keyreply.com/blog/conversational-design-virtual-assistants
12. https://dl.acm.org/doi/10.1145/3663533.3664041
13. https://www.anaconda.com/blog/my-open-source-journey-from-graduation-to-maintainership
14. https://determ.com/blog/pr-automation-best-tools-which-do-the-work-for-you/
15. https://docs.github.com/en/organizations/keeping-your-organization-secure/managing-two-factor-authentication-for-your-organization/managing-bots-and-service-accounts-with-two-factor-authentication
16. https://www.meltwater.com/en/blog/what-does-pr-automation-mean-for-pr-pros
17. https://docs.github.com/en/actions/tutorials/authenticate-with-github_token
18. https://docs.github.com/en/webhooks/webhook-events-and-payloads
19. https://thechigroup.co/articles/2025/1/10/pr-automation-explained-how-to-save-time-and-maximize-your-pr-impact
20. https://docs.github.com/actions/security-guides/automatic-token-authentication
21. https://github.blog/changelog/2025-07-01-enterprise-level-access-for-github-apps-and-installation-automation-apis/
22. https://www.agilitypr.com/pr-news/pr-skills-profession/how-automation-helps-pr-teams-maintain-consistent-messaging-across-channels/
23. https://docs.github.com/en/rest/apps
24. https://www.axiapr.com/blog/5-ways-to-automate-pr-processes-for-streamlined-lead-generation

25. https://github.com/orgs/community/discussions/67874
26. https://docs.github.com/en/apps/creating-github-apps/about-creating-github-apps/about-creating-github-apps
27. https://prprofessionals.in/blog/how-pr-automation-tools-maximize-client-engagement
28. https://docs.github.com/en/code-security/supply-chain-security/end-to-end-supply-chain/securing-accounts
29. https://docs.github.com/en/rest
30. https://wprd.app/the-role-of-ai-and-automation-in-public-relations-pr-hr-implications-and-opportunities-by-adedeji-adeniyi/
31. https://marutitech.com/how-to-choose-right-llm/
32. https://nexla.com/ai-infrastructure/retrieval-augmented-generation/
33. https://himalayas.app/companies/github/tech-stack
34. https://huggingface.co/blog/dvilasuero/choosing-best-open-source-ai-models
35. https://www.merge.dev/blog/rag-best-practices
36. https://github.com/stackshareio/awesome-stacks
37. https://www.linkgraph.com/blog/best-llm-model/
38. https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/rag/rag-solution-design-and-evaluation-guide
39. https://www.imaginarycloud.com/blog/techstack-mobile-app
40. https://arxiv.org/html/2508.00083v1
41. https://orkes.io/blog/rag-best-practices/
42. https://github.com/orgs/community/discussions/167234
43. https://www.flowhunt.io/blog/best-llms-for-coding-june-2025/
44. https://python.langchain.com/docs/tutorials/rag/
45. https://syndicode.com/blog/how-to-choose-tech-stack/
46. https://www.instaclustr.com/education/open-source-ai/top-10-open-source-llms-for-2025/
47. https://cloud.google.com/blog/products/ai-machine-learning/optimizing-rag-retrieval
48. https://github.com/topics/tech-stack
49. https://dextralabs.com/blog/best-llm-for-coding/
50. https://aws.amazon.com/what-is/retrieval-augmented-generation/
51. https://aws.amazon.com/blogs/modernizing-with-aws/automate-microsoft-web-application-deployments-with-github-actions-and-terraform/
52. https://www.koyeb.com/docs/build-and-deploy/deploy-with-git
53. https://www.cognizant.com/us/en/ai-lab/blog/multi-agent-evaluation-system
54. https://www.howtobuysaas.com/blog/pinecone-vs-weaviate-vs-chroma/
55. https://docs.github.com/actions/deployment/about-deployments/deploying-with-github-actions
56. https://research.aimultiple.com/vector-database-for-rag/
57. https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/ai-agent-design-patterns
58. https://lakefs.io/blog/12-vector-databases-2023/
59. https://learn.microsoft.com/en-us/azure/app-service/deploy-github-actions

60. https://www.docker.com/blog/how-to-build-a-multi-agent-system/
61. https://liveblocks.io/blog/whats-the-best-vector-database-for-building-ai-products
62. https://docs.github.com/en/apps/creating-github-apps/about-creating-github-apps/deciding-when-to-build-a-github-app
63. https://www.nitorinfotech.com/blog/multi-agent-collaboration-how-ai-agents-work-together/
64. https://dev.to/dandv/how-to-choose-a-vector-database-pinecone-weaviate-mongodb-atlas-semadb-a09
65. https://codefresh.io/learn/github-actions/deployment-with-github-actions/
66. https://productschool.com/blog/product-strategy/product-template-requirements-document-prd
67. https://docs.github.com/en/apps/creating-github-apps/about-creating-github-apps/best-practices-for-creating-a-github-app
68. https://www.qodo.ai/blog/automated-code-review/
69. https://www.atlassian.com/agile/product-management/requirements
70. https://github.com/security/advanced-security
71. https://coderabbit.ai
72. https://www.notion.com/templates/category/product-requirements-doc
73. https://www.checkpoint.com/cyber-hub/cloud-security/what-is-developer-security/21-security-best-practices-for-github/
74. https://research.aimultiple.com/ai-code-review-tools/
75. https://www.aha.io/roadmapping/guide/requirements-management/what-is-a-good-product-requirements-document-template
76. https://github.blog/changelog/2025-06-24-security-updates-for-apps-and-api-access/
77. https://github.com/resources/articles/ai/ai-code-reviews
78. https://www.figma.com/resource-library/product-requirements-document/
79. https://github.blog/changelog/2025-03-04-introducing-github-secret-protection-and-github-code-security/
80. https://getdx.com/blog/ai-coding-assistant-pricing/
81. https://www.atlassian.com/software/confluence/templates/product-requirements
82. https://www.gitguardian.com/glossary/what-are-github-security-best-practices
83. https://www.greptile.com
84. https://www.hustlebadger.com/what-do-product-teams-do/prd-template-examples/
85. https://docs.github.com/en/apps/github-marketplace/creating-apps-for-github-marketplace/security-best-practices-for-apps-on-github-marketplace
86. https://docs.github.com/en/code-security/dependabot/dependabot-auto-triage-rules/customizing-auto-triage-rules-to-prioritize-dependabot-alerts
87. https://www.reddit.com/r/opensource/comments/1ciq1xo/open_source_maintainers_tell_me_about_your/
88. https://www.conversationdesigninstitute.com/topics/best-practices
89. https://docs.github.com/en/code-security/securing-your-organization/understanding-your-organizations-exposure-to-vulnerabilities/prioritizing-dependabot-alerts-using-metrics
90. https://www.jlowin.dev/blog/oss-maintainers-guide-to-saying-no
91. https://botpress.com/blog/conversation-design

92. https://docs.github.com/en/code-security/code-scanning/managing-code-scanning-alerts/triaging-code-scanning-alerts-in-pull-requests
93. https://nolanlawson.com/2017/03/05/what-it-feels-like-to-be-an-open-source-maintainer/
94. https://developers.google.com/assistant/conversation-design/learn-about-conversation
95. https://james-vu.com/blog/f/step-by-step-guide-to-triage-security-findings-in-github
96. https://antirez.com/news/129
97. https://www.salesforce.com/blog/what-is-conversation-design/
98. https://docs.github.com/en/code-security/dependabot/dependabot-auto-triage-rules
99. https://opensauced.pizza/blog/when-open-source-maintainers-leave
100. https://rasa.com/blog/how-to-design-chatbot-conversation/
101. https://docs.github.com/en/code-security/security-overview/viewing-metrics-for-pull-request-alerts
102. https://openssf.org/blog/2024/01/31/maintainer-motivations-challenges-and-best-practices-on-open-source-software-security/