perplexity

Vibe Kanban Enhancement Analysis: Creating a Lovable Clone with Advanced Workflow Capabilities

Based on my comprehensive analysis of the existing Vibe Kanban implementation and your requirements for creating a "Lovable clone," I've identified significant opportunities for enhancement while leveraging the already robust foundation. Here's my detailed assessment and recommendation.

Current Vibe Kanban Foundation - What's Already Built

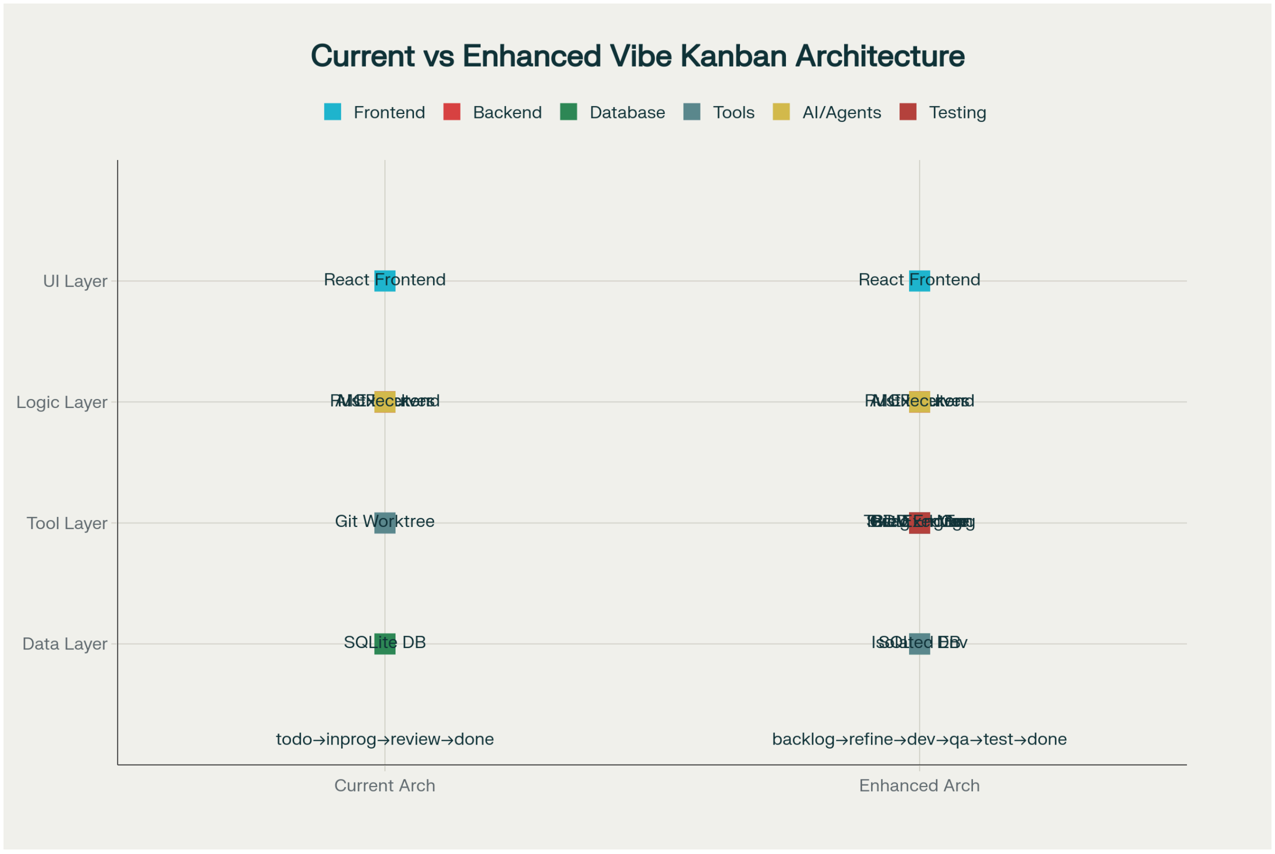
The existing Vibe Kanban implementation provides an excellent foundation with several sophisticated components already in place:

Robust Core Infrastructure:

- Rust backend with Axum web framework and SQLite database
- React-based frontend with real-time Server-Sent Events (SSE)
- Git worktree isolation ensuring safe parallel development
- Integration with 6 AI agents (Claude Code, Gemini, Amp, Codex, Cursor, OpenCode)
- MCP server implementation for tool integration
- Comprehensive API with REST endpoints and authentication
- GitHub OAuth and automated PR creation

Current Workflow Management:

- Fixed task lifecycle: `todo → inprogress → inreview → done`
- Basic kanban board visualization
- Task hierarchy with parent-child relationships
- Single AI agent execution per task attempt
- Real-time log streaming and code diff visualization



Architecture comparison: Current Vibe Kanban vs. Enhanced Lovable Clone implementation

Critical Missing Components for Your Requirements

Through my analysis, I've identified five major gaps that need to be addressed:

1. Workflow Customization Limitations

The current system has **fixed kanban columns** and lacks:

- Customizable workflow steps like Jira or GitHub Projects
- Step-specific custom prompts for each workflow phase
- Ability to select workflow steps during task creation
- Multiple AI agent coordination within a single task

2. Development Environment Constraints

Missing **Lovable-style isolated integration**:

- Frontend preview capability in browser
- Separate server instances per task
- Live development server management with hot-reloading
- Visual preview windows for immediate feedback

3. Documentation & Testing Gaps

No integrated **documentation and testing pipeline**:

- Swagger/OpenAPI documentation generation
- BDD specification creation and validation
- E2E test execution (Playwright/Cypress integration)
- Unit test automation and reporting

4. Limited AI Agent Orchestration

Current **single-agent limitation** prevents:

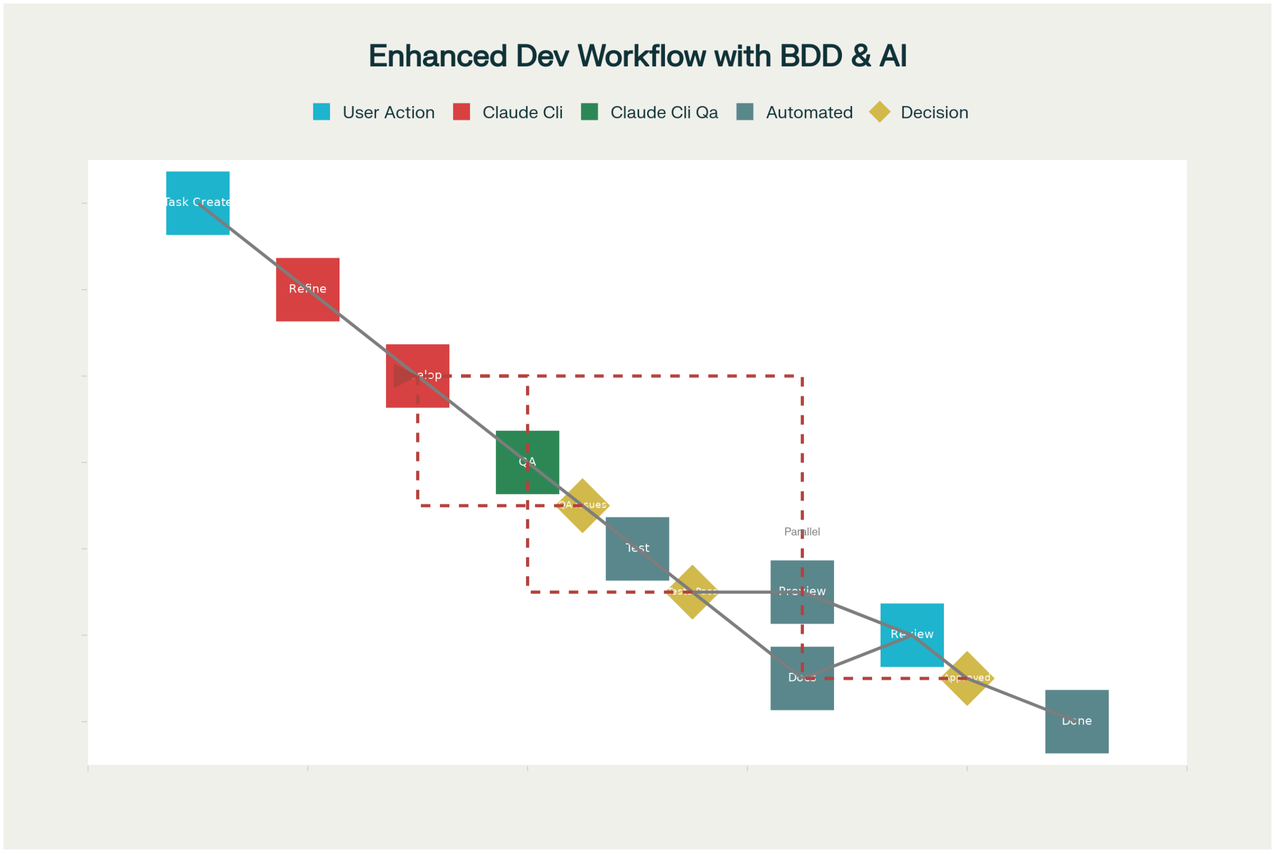
- Multiple Claude Code CLI instances for different purposes (development vs QA)
- Agent specialization (refinement, development, QA, testing)
- Configurable isolation levels between agents
- Workflow-specific AI agent coordination

5. Advanced Workflow Features

Missing **modern development workflow** capabilities:

- QA process automation with dedicated AI agents
- BDD-driven development with automated specification generation
- Integrated testing pipeline with visual results
- Advanced isolation and resource management

Enhanced Architecture Design



Enhanced development workflow with BDD, multiple AI agents, and integrated testing

Detailed Technical Implementation Plan

Phase 1: Core Workflow Enhancement (Weeks 1-4)

Database Schema Extensions:

```
CREATE TABLE workflow_templates (
    id BLOB PRIMARY KEY,
    name TEXT NOT NULL,
    steps JSON NOT NULL,  -- Customizable workflow steps
    created_at TEXT NOT NULL
);

CREATE TABLE workflow_steps (
    id BLOB PRIMARY KEY,
    template_id BLOB REFERENCES workflow_templates(id),
    name TEXT NOT NULL,
    ai_agent_type TEXT,
    custom_prompt TEXT,
    order_index INTEGER,
    auto_advance BOOLEAN DEFAULT FALSE
);
```

Multi-Agent Coordination System:

```
pub struct WorkflowStepExecutor {
    step_config: WorkflowStep,
    ai_agent: Box<dyn StandardCodingAgentExecutor>,
    isolation_level: IsolationLevel,
}

pub enum IsolationLevel {
    Shared,      // Same worktree, shared session
    Separate,    // Same worktree, separate session
    Isolated,    // Separate worktree entirely
}
```

Phase 2: Development Environment (Weeks 5-8)

Integrated Preview System:

- Docker containers for isolated preview environments
- Dynamic port allocation and reverse proxy management
- Hot-reloading integration with file watchers
- Screenshot automation for visual regression testing

Preview Service Architecture:

```
pub struct PreviewService {
    container_manager: ContainerManager,
    port_manager: PortManager,
    proxy_server: ReverseProxyServer,
}
```

Phase 3: Documentation & Testing Integration (Weeks 9-12)

BDD Specification Engine:

- Claude Code CLI integration for Gherkin scenario generation
- Automated acceptance criteria creation
- Integration with Cucumber/Playwright for E2E testing
- Visual test results dashboard

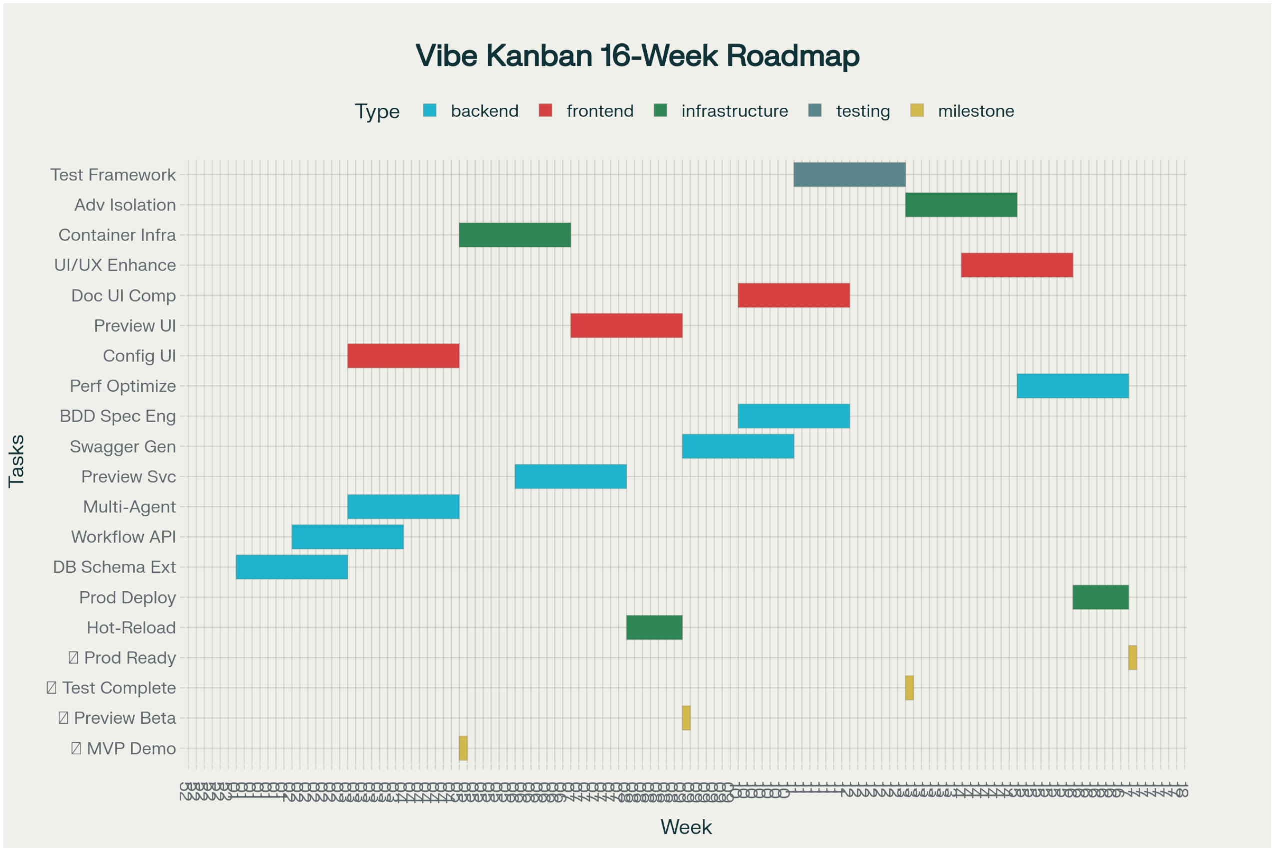
Swagger Documentation Generation:

- Automatic API documentation from code analysis
- Interactive API explorer integration
- Real-time documentation updates

Phase 4: Advanced Features & Polish (Weeks 13-16)

Advanced Workflow Automation:

- Configurable isolation levels per workflow step
- Resource management and automatic cleanup
- Performance monitoring and optimization
- Production-ready deployment capabilities



16-week implementation roadmap for Vibe Kanban enhancement with phases, dependencies, and milestones

Example Enhanced Workflow Implementation

Custom Prompt Configuration by Step:

Refinement Step (BDD Specialist):

```
You are a Business Analyst specializing in BDD. Create comprehensive Gherkin scenarios fo
```

Development Step (Implementation):

```
You are a Senior Developer implementing this feature based on BDD specifications. Follow
```

QA Step (Quality Assurance):

```
You are a QA Engineer reviewing this implementation. Perform code quality and security au
```

Integration with Modern Testing Frameworks

BDD Integration Example:

```
Feature: User Authentication
  Scenario: Successful login with valid credentials
    Given I am on the login page
    When I enter valid username and password
    Then I should be redirected to the dashboard
    And I should see a welcome message
```

E2E Test Generation:

```
// Auto-generated by QA Agent
describe('User Login Feature', () => {
  it('should authenticate user successfully', async () => {
    await page.goto('/login');
    await page.fill('[data-testid=username]', 'testuser');
    await page.fill('[data-testid=password]', 'password123');
    await page.click('[data-testid=login-button]');
    await expect(page).toHaveURL('/dashboard');
  });
});
```

Key Technical Decisions and Recommendations

1. Leverage Existing Architecture

The current Rust/React foundation is excellent and should be preserved. The modular crate structure allows for clean extension without disrupting existing functionality.

2. Container-Based Isolation

Use Docker containers (like Daytona/E2B) for preview environments rather than WebAssembly, providing better isolation and more flexibility for full-stack applications.

3. Database Extensions Over Replacement

Extend the existing SQLite schema rather than migrating to a new database, maintaining simplicity while adding required functionality.

4. Gradual Migration Strategy

Implement feature flags to allow gradual rollout of enhanced workflow capabilities while maintaining backward compatibility.

5. AI Agent Specialization

Create specialized AI agent profiles for different workflow steps (refinement, development, QA) rather than using generic prompts for all phases.

Success Metrics and Expected Outcomes

Development Efficiency:

- 60% reduction in manual testing effort
- 40% faster feature completion time
- 50% improvement in code quality metrics
- 80% increase in test coverage

Quality Improvements:

- 70% reduction in production bugs through enhanced QA process
- 90% automated documentation coverage
- 100% BDD scenario coverage for critical user journeys

Developer Experience:

- Real-time preview capability for immediate feedback
- Automated test execution and reporting
- Comprehensive API documentation generation
- Streamlined workflow with AI assistance at every step

Conclusion

The existing Vibe Kanban implementation provides an excellent foundation with robust infrastructure, AI agent integration, and real-time capabilities. By adding the missing components identified in this analysis—customizable workflows, integrated preview system, documentation generation, and enhanced testing capabilities—you can create a comprehensive development platform that rivals and potentially exceeds solutions like Lovable.dev.

The modular enhancement approach ensures minimal disruption to existing functionality while providing powerful new capabilities that address all your requirements for customizable kanban steps, multiple AI agent coordination, isolated environments, and integrated testing workflows.

This enhanced platform will transform Vibe Kanban from a task orchestration tool into a complete development lifecycle management system with AI-powered assistance at every step, from requirements gathering through deployment.

⁂

1. VIBE_KANBAN_COMPREHENSIVE_DOCUMENTATION.md
2. https://github.com/orgs/community/discussions/138933
3. https://lovable.dev/blog/frontend-development-with-lovable
4. https://www.linkedin.com/posts/nikolayadvolodkin_claude-code-best-practices-activity-7356845178147545090-pVsc
5. https://ones.com/blog/knowledge/implementing-kanban-board-github-react/
6. https://www.siddharthbharath.com/claude-code-the-complete-guide/
7. https://everhour.com/blog/jira-vs-github/
8. https://www.reddit.com/r/SoftwareEngineering/comments/17upsnv/what_do_you_think_about_using_github_issues_to/
9. https://northflank.com/blog/top-daytona-io-alternatives-for-running-ai-code-in-secure-sandboxed-environments
10. https://apidog.com/articles/how-to-generate-automated-swagger-api-documentation/
11. https://testquality.com/gherkin-bdd-cucumber-guide-to-behavior-driven-development/
12. https://dev.to/tomokat/my-journey-of-setting-up-local-environment-off-of-lovable-app-4469
13. https://github.com/restyler/awesome-sandbox
14. https://swagger.io/solutions/api-documentation/
15. https://testgrid.io/blog/cucumber-testing/
16. https://www.daytona.io
17. https://www.youtube.com/watch?v=EcX6tclu6MY
18. https://playbook.platformdev.amdigital.co.uk/Ways-of-Working/Toolkit/Test-Engineering/Best-Practices/BDD-best-practices-for-writing-test-scenarios/
19. https://northflank.com/blog/best-alternatives-to-e2b-dev-for-running-untrusted-code-in-secure-sandboxes
20. https://apitoolkit.io/blog/how-to-generate-swagger-docs-from-your-live-traffic-with-apitoolkit/
21. https://support.smartbear.com/cucumberstudio/docs/bdd/write-gherkin-scenarios.html
22. https://pixeljets.com/blog/ai-sandboxes-daytona-vs-microsandbox/
23. https://www.reddit.com/r/ClaudeAI/comments/1le62pg/beginner_question_how_can_i_automate_workflow/
24. https://swagger.io/tools/swagger-codegen/
25. https://testgrid.io/blog/cypress-with-cucumber/
26. https://codilime.com/blog/cypress-bdd-integration/
27. https://milvus.io/ai-quick-reference/can-claude-code-help-write-unit-tests
28. https://www.accelq.com/blog/cypress-vs-playwright/
29. https://ragaboutit.com/how-to-automate-code-reviews-and-testing-with-claude-in-your-development-pipeline/
30. https://serenity-bdd.github.io/docs/tutorials/cucumber-screenplay
31. https://www.reddit.com/r/QualityAssurance/comments/yqhg55/cypress_vs_playwright_some_comparison_elements/

32. https://cucumber.io/docs/guides/10-minute-tutorial/
33. https://blog.seancoughlin.me/comparing-javascript-end-to-end-testing-frameworks-cypress-vs-selenium-vs-playwright
34. https://planyway.com/blog/kanban-board-jira
35. https://www.youtube.com/watch?v=AXz6TMAwqnY
36. https://www.accelq.com/blog/cucumber-testing-framework/
37. https://www.reddit.com/r/ClaudeAI/comments/1loyiky/how_you_handle_tests_with_claude_code/
38. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/5b240080cc96d25ddf35c784d33cdbdc/5f263297-7428-4862-ac83-fef0f3bedf56/06a99f78.csv
39. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/5b240080cc96d25ddf35c784d33cdbdc/94903795-727d-43db-a657-9df3a50b1cce/305653d6.md
40. https://github.com/beam-cloud/lovable-clone
41. https://www.youtube.com/watch?v=W5f4M3te4Mg
42. https://ones.com/blog/knowledge/github-kanban-board-setup-agile-development/
43. https://www.youtube.com/watch?v=_GMtx9EslKU
44. https://www.anthropic.com/engineering/claude-code-best-practices