Eklavya Product Roadmap

> ***Version**: 1.0 **Last Updated**: January 2026 **Status**: Pre-Implementation Review*

Table of Contents

1. Vision Statement
2. Feature Set (What's Included)
3. What's NOT Included
4. Development Phases
5. Demo Strategy
6. Success Metrics
7. Risk Assessment
8. Dependencies

---

1. Vision Statement

The Problem

Running a software development business today requires constant attention:

- Clients expect fast turnaround
- Managing multiple projects is exhausting
- You can't work while sleeping
- Junior developers need supervision
- Every project starts from scratch

The Solution

Eklavya is an **autonomous agent orchestration platform** that:

- Takes client requirements and works independently
- Builds impressive demos to win contracts BEFORE full development
- Handles multiple projects simultaneously
- Works 24/7 while you sleep
- Learns and improves from every project

Success Looks Like

```
┌─────────────────────────────────────────────────────────────────────────┐
│                          SUCCESS SCENARIO                               │
├─────────────────────────────────────────────────────────────────────────┤
│                                                                         │
│   MONDAY MORNING:                                                       │
│                                                                         │
│   You wake up and check Eklavya:                                        │
│                                                                         │
│   ┌─────────────────────────────────────────────────────────────┐       │
│   │  "Good morning! While you slept:"                           │       │
│   │                                                             │       │
│   │  ✅ Pet Store App - Demo₀ ready for your review             │       │
```

```
│   │  ✅ Invoice System - Full build complete, ready to ship          │  │
│   │  🔄 Blog Platform - 73% complete, continuing...                   │  │
│   │                                                                   │  │
│   │  Total spent overnight: $34.50                                    │  │
│   │  3 client projects progressed without you                         │  │
│   └──────────────────────────────────────────────────────────────────┘  │
│                                                                         │
│   You review the demo, share it with the client, close the deal.        │
│   All before your morning coffee.                                       │
│                                                                         │
└─────────────────────────────────────────────────────────────────────────┘
```

---

2. Feature Set (What's Included)

2.1 Core Features

Project Management

Feature	Description	Priority
New Project Creation	Chat-based project setup, describe what you want	P0
Import Existing Project	Analyze half-baked projects, create recovery plan	P0
Multi-Project Dashboard	See all projects at a glance	P0
Project Status Tracking	Real-time progress with visual indicators	P0
Budget Management	Per-project spending limits and tracking	P0
Project Archival	Archive completed or abandoned projects	P1

Agent System

Feature	Description	Priority
10 Agent Types	Orchestrator, Architect, Developer, Tester, QA, PM, UAT, SRE, Monitor, Mentor	P0
Parallel Execution	Multiple agents work simultaneously	P0
Agent Checkpointing	Save and resume agent state	P0
Agent Monitoring	See what each agent is doing	P0
Smart Task Routing	Orchestrator assigns work intelligently	P1

Demo System (Sales Tool)

Feature	Description	Priority
Demo_0 (Wow Demo)	Initial impressive demo to hook client	P0
Demo_1 (Trust Demo)	Deeper functionality demo	P0

Demo Preview URLs	Shareable links to show clients	P0
Admin Review First	Admin sees demo before client	P0
Skip to Build Option	Admin can skip remaining demos	P0
Demo Scaffolding Reuse	Demo code becomes production foundation	P0

Admin Operations

Feature	Description	Priority
Approval Gates	Required approval before demos and major phases	P0
Smart Notifications	4 levels (Critical/Needs Input/Info/Silent)	P0
Escalation Policy	Configure when agents stop vs proceed	P0
Availability Mode	Set your availability for notification routing	P1
Activity History	See what happened while you were away	P0

Client Flow (Admin-Controlled)

Feature	Description	Priority
Client Feedback Recording	Log client feedback from your calls	P0
Decision Tracking	Record decisions after each demo	P0
Client Information Storage	Store client name, contact, project details	P0

2.2 Technical Features

Development Infrastructure

Feature	Description	Priority
GitHub Integration	Auto-create repos, commit changes	P0
Project Isolation	Docker containers per project	P0
Code Quality Checks	Linting, type checking	P0
Test Automation	Unit and integration tests	P0

Learning System

Feature	Description	Priority
Prompt Versioning	Track and evolve agent prompts	P1
Outcome Tracking	Record success/failure for learning	P1
Thompson Sampling	Statistical prompt selection	P2

Reliability

Feature	Description	Priority
Agent Checkpoints	Save state every 15 minutes	P0
Recovery from Failure	Resume from last checkpoint	P0
Heartbeat Monitoring	Detect stuck agents	P0
Graceful Degradation	Handle API outages gracefully	P1

2.3 Feature Priority Legend

- **P0** = Must have for v1.0 (MVP)
- **P1** = Should have for v1.0 (important)
- **P2** = Nice to have (post-MVP)

---

3. What's NOT Included

3.1 Explicitly Out of Scope for v1.0

```
┌─────────────────────────────────────────────────────────────────────┐
│                        NOT IN VERSION 1.0                           │
├─────────────────────────────────────────────────────────────────────┤
│                                                                     │
│  MULTI-TENANCY / TEAM FEATURES                                      │
│  ✗ Multiple admin accounts                                          │
│  ✗ Role-based access control                                        │
│  ✗ Team collaboration                                               │
│  ✗ Permission management                                            │
│  Reason: Single-admin business model for v1.0                       │
│                                                                     │
│  CLIENT DIRECT ACCESS                                               │
│  ✗ Client portal                                                    │
│  ✗ Client self-service demo viewing                                 │
│  ✗ Client authentication                                            │
│  Reason: Admin controls all client interactions                     │
│                                                                     │
│  PAYMENT / BILLING                                                  │
│  ✗ Stripe integration                                               │
│  ✗ Client invoicing                                                 │
│  ✗ Subscription management                                          │
│  Reason: Use existing billing tools, integrate later                │
│                                                                     │
│  MOBILE APP                                                         │
│  ✗ Native iOS app                                                   │
│  ✗ Native Android app                                               │
│  Reason: Mobile-responsive web app is sufficient                    │
│                                                                     │
│  ADVANCED DEPLOYMENT                                                │
│  ✗ Kubernetes orchestration                                         │
```

```
│  ✗ Multi-region deployment                                                  │
│  ✗ Auto-scaling                                                             │
│  Reason: Single-server deployment handles initial scale                     │
│                                                                             │
│  WHITE-LABELING                                                             │
│  ✗ Custom branding                                                          │
│  ✗ Custom domains per client                                                │
│  ✗ Reseller features                                                        │
│  Reason: Internal tool, not for resale                                      │
│                                                                             │
│  ADVANCED ANALYTICS                                                         │
│  ✗ Revenue forecasting                                                      │
│  ✗ Client profitability analysis                                            │
│  ✗ Agent efficiency dashboards                                              │
│  Reason: Basic stats sufficient for v1.0                                    │
│                                                                             │
│  INTEGRATIONS                                                               │
│  ✗ Jira integration                                                         │
│  ✗ Linear integration                                                       │
│  ✗ Notion integration                                                       │
│  ✗ Slack/Discord bots                                                       │
│  Reason: Direct admin interaction for v1.0                                  │
│                                                                             │
│  AGENT CUSTOMIZATION                                                        │
│  ✗ Custom agent types                                                       │
│  ✗ Visual prompt builder                                                    │
│  ✗ Agent marketplace                                                        │
│  Reason: 10 built-in agents sufficient for v1.0                             │
│                                                                             │
└─────────────────────────────────────────────────────────────────────────────┘
```

3.2 Technical Limitations (v1.0)

Limitation	Why It's Okay
Single admin only	Business is run by one person
~50 concurrent agents	Sufficient for 5-10 projects
Docker-only isolation	No need for Kubernetes yet
PostgreSQL on same host	Can migrate to managed DB later
No HA/failover	Acceptable for initial usage

3.3 Future Roadmap (Post v1.0)

These may be added in future versions based on need:

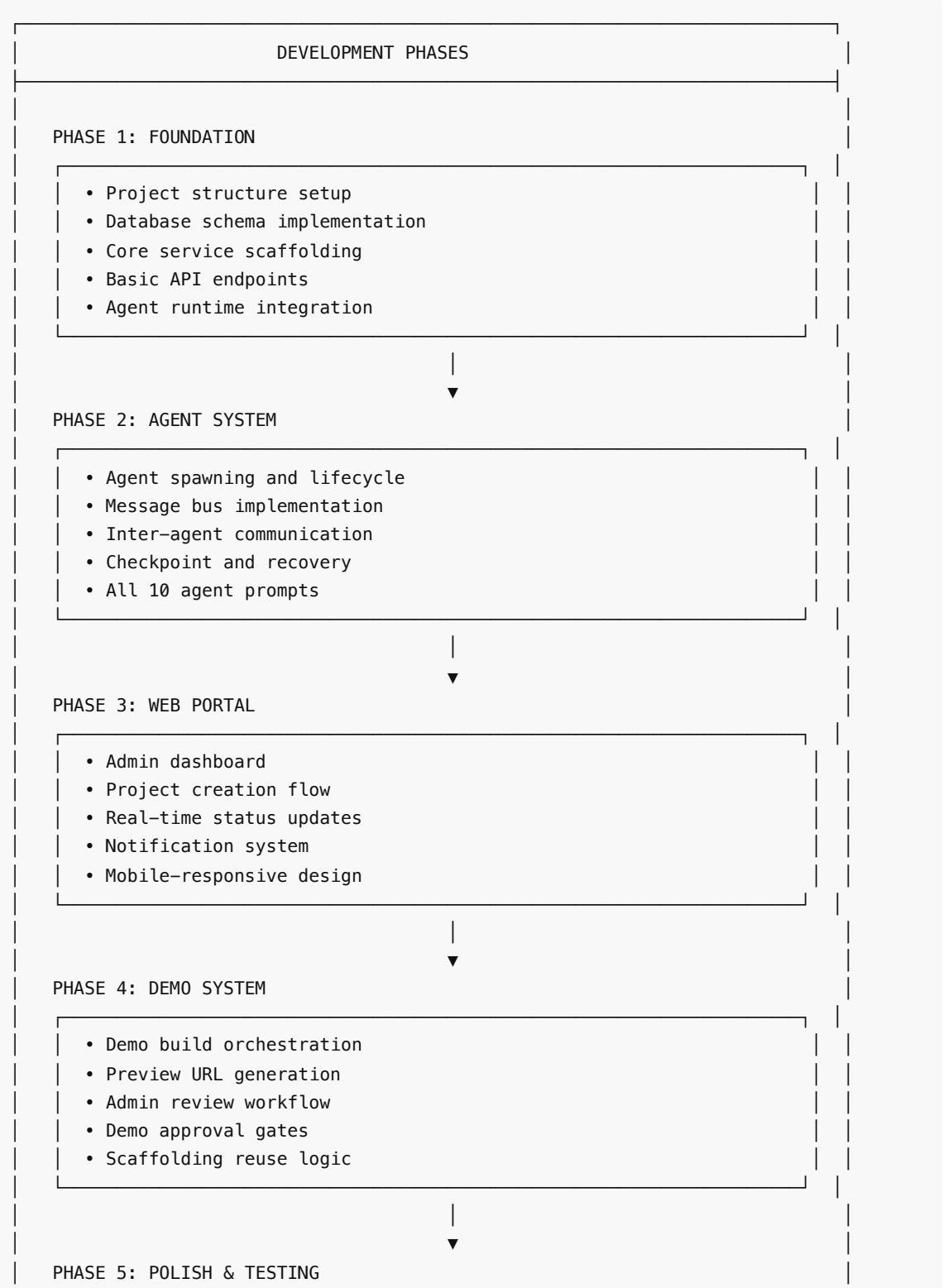
- Multi-admin support (v1.5)
- Advanced analytics dashboard (v1.5)
- Slack notifications (v2.0)
- Client portal option (v2.0)

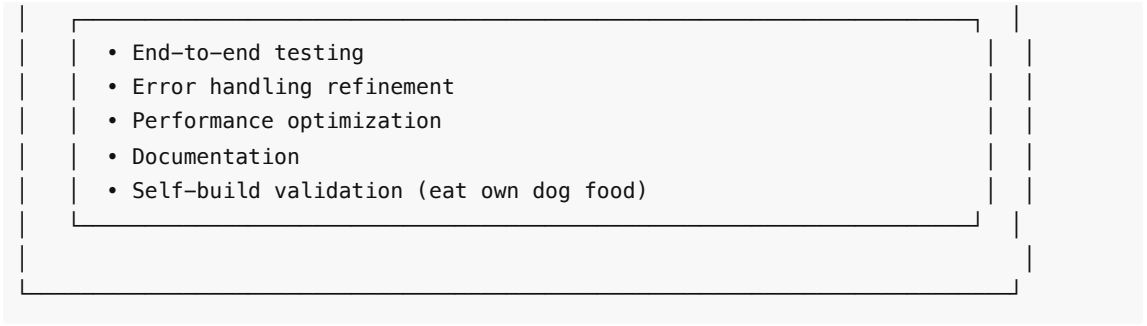
- Kubernetes deployment (v2.0)

---

4. Development Phases

Phase Overview





Phase 1: Foundation

Goal: Set up project structure and core infrastructure

Task	Description
Project scaffolding	Next.js app, TypeScript config, folder structure
Database setup	PostgreSQL schema, migrations, seed data
Redis setup	Connection, pub/sub channels, caching layer
Core services	Project Manager, basic API routes
Docker setup	Development docker-compose, project containers
CI/CD	GitHub Actions for testing and linting

Deliverable: Running Next.js app with database connected

Phase 2: Agent System

Goal: Agents can spawn, communicate, and checkpoint

Task	Description
Agent Manager service	Spawn, monitor, terminate agents
Message Bus	Redis pub/sub, message routing
Agent runtime	Claude Code integration, workspace setup
Agent prompts	All 10 agent type prompts
Checkpointing	State save/restore, file snapshots
Orchestrator agent	Task breakdown, agent coordination

Deliverable: Orchestrator can spawn Developer, assign task, receive result

Phase 3: Web Portal

Goal: Admin can create projects and monitor progress

Task	Description

Dashboard UI	Multi-project view, status cards
Project creation	Chat-based input, requirement capture
Import existing	Project analysis, recovery plan generation
Real-time updates	WebSocket connection, live status
Notifications	Smart alert system, escalation
Mobile responsive	Works on phone and tablet

Deliverable: Admin can create project, see agents working in real-time

Phase 4: Demo System

Goal: Demos can be built, reviewed, and approved

Task	Description
Demo orchestration	Parallel agent builds for demo
Preview URLs	Temporary deployment for demos
Review workflow	Admin sees demo first
Approval gates	Pre-demo and post-demo gates
Decision recording	Track admin decisions, client feedback
Scaffolding reuse	Demo code transitions to production

Deliverable: Full demo workflow from build to client feedback

Phase 5: Polish & Testing

Goal: Production-ready quality

Task	Description
E2E test suite	Playwright tests for critical flows
Error handling	Graceful failures, helpful messages
Performance	Query optimization, caching
Documentation	Setup guide, API docs
Self-build test	Use Eklavya to build a small project

Deliverable: Stable v1.0 ready for daily use

---

5. Demo Strategy

$Demo_0$: Wow Demo (Eklavya Building Itself)

```
┌─────────────────────────────────────────────────────────────────────────┐
│                        DEMO₀ FOR EKLAVYA                                │
├─────────────────────────────────────────────────────────────────────────┤
│                                                                         │
│  PURPOSE: Show that Eklavya works, builds impressive things             │
│                                                                         │
│  WHAT TO BUILD:                                                         │
│  ┌─────────────────────────────────────────────────────────────────┐    │
│  │  ✓ Working admin dashboard with project cards                   │    │
│  │  ✓ Real-time agent activity visualization                       │    │
│  │  ✓ Chat-based project creation (UI only, wired to mock)         │    │
│  │  ✓ Beautiful, professional design                               │    │
│  │  ✓ Mobile-responsive layout                                     │    │
│  └─────────────────────────────────────────────────────────────────┘    │
│                                                                         │
│  WHAT TO SKIP:                                                          │
│  ┌─────────────────────────────────────────────────────────────────┐    │
│  │  ✗ Actual agent spawning (mocked)                               │    │
│  │  ✗ Real project execution                                       │    │
│  │  ✗ Database persistence (in-memory okay)                        │    │
│  │  ✗ Authentication                                               │    │
│  └─────────────────────────────────────────────────────────────────┘    │
│                                                                         │
│  RESULT: ~40% of UI foundation built, looks amazing                     │
│                                                                         │
└─────────────────────────────────────────────────────────────────────────┘
```

Demo₁: Trust Demo (Eklavya Building Itself)

```
┌─────────────────────────────────────────────────────────────────────────┐
│                        DEMO₁ FOR EKLAVYA                                │
├─────────────────────────────────────────────────────────────────────────┤
│                                                                         │
│  PURPOSE: Show that agents actually work, real functionality            │
│                                                                         │
│  WHAT TO ADD:                                                           │
│  ┌─────────────────────────────────────────────────────────────────┐    │
│  │  ✓ Real agent spawning (at least Orchestrator + 1 Developer)    │    │
│  │  ✓ Live task execution visible in UI                            │    │
│  │  ✓ Database persistence working                                 │    │
│  │  ✓ Simple project actually builds                               │    │
│  │  ✓ Checkpoint and recovery demo                                 │    │
│  └─────────────────────────────────────────────────────────────────┘    │
│                                                                         │
│  WHAT TO SKIP:                                                          │
│  ┌─────────────────────────────────────────────────────────────────┐    │
│  │  ✗ All 10 agent types (3-4 enough)                              │    │
│  │  ✗ Learning system                                              │    │
│  │  ✗ Complex project handling                                     │    │
│  │  ✗ Import existing project                                      │    │
│  └─────────────────────────────────────────────────────────────────┘    │
```

```
│                                                                              │
│  RESULT: ~60% complete, core functionality proven                            │
│                                                                              │
└──────────────────────────────────────────────────────────────────────────────┘
```

After Demos → Full Build

```
┌──────────────────────────────────────────────────────────────────────────────┐
│                    FULL BUILD (Remaining 40%)                                │
├──────────────────────────────────────────────────────────────────────────────┤
│                                                                              │
│  COMPLETE THESE:                                                             │
│  ┌──────────────────────────────────────────────────────────────────────┐    │
│  │  ✓ All 10 agent types with full prompts                              │    │
│  │  ✓ Import existing project feature                                   │    │
│  │  ✓ Smart notification system                                         │    │
│  │  ✓ Escalation policy configuration                                   │    │
│  │  ✓ Budget tracking and limits                                        │    │
│  │  ✓ Complete demo workflow with approval gates                        │    │
│  │  ✓ Comprehensive testing                                             │    │
│  │  ✓ Documentation                                                     │    │
│  │  ✓ Error handling and recovery                                       │    │
│  └──────────────────────────────────────────────────────────────────────┘    │
│                                                                              │
│  Demo scaffolding is NOT thrown away - it becomes the foundation             │
│                                                                              │
└──────────────────────────────────────────────────────────────────────────────┘
```

---

6. Success Metrics

v1.0 Launch Criteria

Metric	Target
Can create new project from chat	✓ Working
Can import existing project	✓ Working
Agents spawn and communicate	✓ Working
Demos build and deploy	✓ Working
Admin notifications work	✓ Working
Budget tracking accurate	✓ Working
Checkpoints save/restore	✓ Working
Mobile-responsive UI	✓ Working
Self-build test passes	✓ Builds simple project

Post-Launch Success Metrics

Metric	30-Day Target
Projects completed	≥ 3
Average project success rate	≥ 80%
Admin intervention rate	≤ 20% of tasks
Avg. $Demo_0$ build time	≤ 45 minutes
System uptime	≥ 95%

---

7. Risk Assessment

Technical Risks

Risk	Impact	Likelihood	Mitigation
Claude API rate limits	High	Medium	Implement queuing, backoff
Agent gets stuck in loop	Medium	Medium	Timeout + checkpoint recovery
Container resource exhaustion	Medium	Low	Per-project resource limits
Database performance	Low	Low	Indexing, query optimization

Business Risks

Risk	Impact	Likelihood	Mitigation
API cost overrun	High	Medium	Hard budget limits, alerts
Agents produce poor code	High	Medium	Review gates, testing
Complex projects fail	Medium	Medium	Start with simpler projects
Learning from mistakes	Low	Low	Logging all outcomes

Mitigation Strategies

```
┌─────────────────────────────────────────────────────────────────────┐
│                        RISK MITIGATION                              │
├─────────────────────────────────────────────────────────────────────┤
│                                                                     │
│  COST CONTROL:                                                      │
│  • Hard per-project budget limits                                   │
│  • Warning at 50%, 75%, 90% of budget                               │
│  • Auto-pause at 100%                                               │
│  • Daily spending summary                                           │
│                                                                     │
│  QUALITY CONTROL:                                                   │
```

```
│  • Admin approval before demos                                              │
│  • Admin review after demos                                                 │
│  • Automatic linting and testing                                            │
│  • Checkpoint before risky operations                                       │
│                                                                             │
│  FAILURE RECOVERY:                                                          │
│  • Checkpoint every 15 minutes                                              │
│  • Resume from any checkpoint                                               │
│  • Graceful degradation on API issues                                       │
│  • Agent heartbeat monitoring                                               │
│                                                                             │
└─────────────────────────────────────────────────────────────────────────────┘
```

---

8. Dependencies

External Dependencies

Dependency	Required	Notes
Anthropic API	Yes	Claude models for agents
GitHub Account	Yes	Repository for projects
Node.js 20+	Yes	Runtime
Docker	Yes	Project isolation
PostgreSQL 16	Yes	Primary database
Redis 7	Yes	Message queue, caching

Development Dependencies

Tool	Purpose
pnpm	Package management
TypeScript	Type safety
Vitest	Unit testing
Playwright	E2E testing
Drizzle	Database ORM
TailwindCSS	Styling

Accounts Required

Account	Purpose
Anthropic	API access for Claude
GitHub (ganeshpandeyvns)	Code repositories

(Optional) Vercel	Demo deployments

Summary

What We're Building

Eklavya v1.0 is an autonomous agent platform for running a software development business:

- **Chat-first project creation** - Describe what you want, agents build it
- **Demo-first sales** - Win contracts with impressive demos before full build
- **Autonomous execution** - Agents work while you sleep
- **Admin control** - Approval gates keep you in charge

What We're NOT Building

- Multi-user/team features
- Client direct access
- Payment/billing
- Native mobile apps
- Advanced analytics
- Third-party integrations

Development Approach

1. Foundation → Agent System → Web Portal → Demo System → Polish
2. $Demo_0$ builds 40% (impressive UI) → $Demo_1$ adds 20% (real functionality) → Full build completes 40%
3. Demo code becomes production foundation (no throwaway work)

Document generated for pre-implementation review. Approved architecture and roadmap will guide development.