



F&B AI Purchasing Platform - 12 Sprint Development Plan

Timeline: 24 weeks | **Sprints:** 12 × 2 weeks | **Team:** Cross-functional | **Target:** Full platform launch



Sprint Plan Overview

PHASE 1: FOUNDATION (Sprints 1-3 | Weeks 1-6)

- └ Sprint 1: Architecture & Setup
- └ Sprint 2: Core Backend & APIs
- └ Sprint 3: Database & Integrations

PHASE 2: AI AGENTS (Sprints 4-6 | Weeks 7-12)

- └ Sprint 4: Agent Framework & Query Agent
- └ Sprint 5: Sales & Purchasing Agents
- └ Sprint 6: Agent Optimization & Testing

PHASE 3: USER INTERFACES (Sprints 7-9 | Weeks 13-18)

- └ Sprint 7: Web Dashboard (Restaurant)
- └ Sprint 8: Web Dashboard (Supplier)
- └ Sprint 9: Mobile App (iOS + Android)

PHASE 4: INTEGRATIONS & POLISH (Sprints 10-12 | Weeks 19-24)

- └ Sprint 10: External API Integration
- └ Sprint 11: Testing & QA
- └ Sprint 12: Launch & Monitoring

Total Timeline: 24 weeks (~6 months)

Team Composition Framework

Core Team (All Sprints)

- **Product Owner** (1) - Requirements, prioritization, stakeholder management
- **Scrum Master** (1) - Process, blockers, team health
- **Tech Lead** (1) - Architecture decisions, code quality

Variable Team (Sprint-Specific)

Backend Engineers: 3-4 **Frontend Engineers:** 2-3 **AI/ML Engineers:** 2 **QA/Testing:** 1-2 **DevOps Engineer:** 1 (Part-time to full-time) **Documentation:** 1 (Shared)

Total Core Team: 12-15 people



Sprint Breakdown

PHASE 1: FOUNDATION

Sprint 1: Architecture & Setup (Week 1-2)

Sprint Goal: Establish development infrastructure and architectural foundation

Team Composition (8 people)

Role	Count	Names/Allocation
Product Owner	1	Requirements & vision
Tech Lead	1	Architecture lead
Backend Lead	1	Backend architecture
Backend Engineers	2	Infrastructure setup
DevOps Engineer	1	Infrastructure & CI/CD
Frontend Lead	1	Frontend setup
Scrum Master	1	Process & coordination

Deliverables

- ✔ Development environment setup (local, staging, production)
- ✔ Git repository structure & CI/CD pipeline
- ✔ Docker containerization for all services
- ✔ Kubernetes cluster setup (dev/staging/prod)
- ✔ Architecture documentation (decisions, trade-offs)
- ✔ Database design (PostgreSQL schema draft)
- ✔ API gateway structure (Express.js skeleton)
- ✔ Authentication framework (JWT + RBAC setup)

Key Dependencies

- GitHub/GitLab access
- AWS/Cloud account setup
- Team tools (Slack, Jira, etc.)

Risks & Mitigation

Risk	Probability	Impact	Mitigation
Infrastructure delays	Medium	High	Pre-allocate DevOps 2 weeks early
Architecture churn	Medium	High	Lock architecture in Week 1 review
Tool setup confusion	Low	Medium	Create setup documentation Day 1

Sprint 2: Core Backend & APIs (Week 3-4)

Sprint Goal: Build foundational backend services and API layer

Team Composition (9 people)

Role	Count	Notes
------	-------	-------

Product Owner	1	Requirements
Tech Lead	1	Oversee backend
Backend Engineers	3	Core services
AI/ML Engineer	1	LangGraph setup
Frontend Engineer	1	API contract testing
QA Engineer	1	API testing
DevOps Engineer	1	Infrastructure
Scrum Master	1	Process

Deliverables

- ✔ MedusaJS 2.0 setup & customization
- ✔ Core API endpoints (REST + GraphQL)
- ✔ Product service (SKUs, pricing, inventory)
- ✔ Order service foundation
- ✔ User service (authentication, RBAC)
- ✔ LangGraph framework integration
- ✔ Message queue (BullMQ + Redis)
- ✔ API documentation (Swagger/OpenAPI)

Key Metrics

- API response time: <200ms (p95)
- Test coverage: >80%
- Deployment: 1-click from CI/CD

Dependencies

- Sprint 1 completion (infrastructure)

Sprint 3: Database & Integrations (Week 5-6)

Sprint Goal: Complete data layer and external service integration

Team Composition (10 people)

Role	Count	Notes
Product Owner	1	Requirements
Tech Lead	1	Oversee
Backend Engineers	3	Data + integrations
Database Engineer	1	Schema, optimization
AI/ML Engineer	1	Vector DB (Weaviate)
QA Engineer	1	Integration testing
DevOps Engineer	1	Infrastructure

Frontend Engineer	1	Mock endpoints
Documentation	1	API docs
Scrum Master	1	Process

Deliverables

- ✔ PostgreSQL schema (complete)
- ✔ Weaviate vector database setup
- ✔ Database migrations & versioning
- ✔ WhatsApp Business API integration
- ✔ Poppel (e-invoicing) API integration
- ✔ Telr (payment gateway) integration
- ✔ Foodics POS API connector
- ✔ AWS Textract (OCR) integration
- ✔ Database backup & recovery procedures
- ✔ Performance testing (load benchmarks)

Key Metrics

- Database query: <50ms (p95)
- API integration success rate: >99%
- Data consistency: 100%

Dependencies

- Sprint 1-2 completion

PHASE 2: AI AGENTS

Sprint 4: Agent Framework & Query Agent (Week 7-8)




Sprint Goal: Build LangGraph agent framework and first agent (Query Agent)

Team Composition (8 people)

Role	Count	Notes
Product Owner	1	Requirements
Tech Lead	1	Agent architecture
AI/ML Engineers	2	LangGraph expertise
Backend Engineers	2	Tool implementations
QA Engineer	1	Agent testing
Documentation	1	Agent specs

Deliverables

- ✔ LangGraph state machine framework
- ✔ Tool registration & execution system
- ✔ Query Agent implementation
 - Natural language intent classification

- Supplier search (vector DB)
 - Price comparison logic
 - Ranking algorithm
-  Agent testing framework
-  Prompt engineering & optimization
-  Agent performance metrics dashboard

Agent Specs

Query Agent:

- Input: Natural language query ("Need 50kg apples")
- Output: Ranked list of suppliers (top 3)
- Performance target: <3 seconds
- Accuracy target: >95% intent classification

Key Metrics

- Agent response time: <3s
- Intent accuracy: >95%
- Supplier match quality: >90%

Dependencies

- Sprint 3 completion (databases, integrations)



Sprint 5: Sales & Purchasing Agents (Week 9-10)




Sprint Goal: Build supplier and restaurant AI agents

Team Composition (10 people)

Role	Count	Notes
Product Owner	1	Requirements
Tech Lead	1	Agent lead
AI/ML Engineers	2	Core agent logic
Backend Engineers	2	Tool implementations
QA Engineers	2	Agent testing
Documentation	1	Agent docs
Business Analyst	1	Domain knowledge

Deliverables

-  Autonomous Sales Agent
 - Quote generation (<3s)
 - Margin guardrails enforcement
 - Basket-aware upselling
 - Distressed inventory liquidation
 - Multi-language support (AR, EN)
-  Purchasing Agent (Restaurant)

- Smart cart generation
 - Quantity calculation
 - Supplier preference learning
-  Inventory Agent
 - Stock level monitoring
 - Automatic reorder triggering
 - Expiry tracking
-  Agent reasoning logs
-  A/B testing framework for agents

Key Metrics

- Sales Agent quote generation: 2.3s average
- Sales Agent win rate: >78%
- Cart accuracy: >95%
- Inventory alerts: <5min latency

Agent Performance Targets

Sales Agent:

- └ Quote response time: <3 seconds
- └ Quote-to-order rate: >75%
- └ Upsell conversion: >30%
- └ Margin guardrails: 100% compliance

Purchasing Agent:

- └ Cart generation time: <10 seconds
- └ Cart accuracy: >95%
- └ Cost optimization: >10% savings
- └ Human approval required: Always

Dependencies

- Sprint 4 completion

Sprint 6: Agent Optimization & Testing (Week 11-12)

Sprint Goal: Optimize agents, extensive testing, production readiness

Team Composition (9 people)

Role	Count	Notes
Product Owner	1	Prioritization
Tech Lead	1	Oversight
AI/ML Engineers	2	Optimization
QA Engineers	3	Testing
Backend Engineers	1	Performance fixes
Documentation	1	Test docs

Deliverables

- ✔ Agent performance optimization
- ✔ Load testing (1000+ concurrent agents)
- ✔ Stress testing
- ✔ Chaos engineering
- ✔ Agent monitoring dashboards
- ✔ Error handling & recovery
- ✔ Fallback mechanisms
- ✔ Agent logging & analytics
- ✔ Production deployment checklist

Test Coverage

Unit Tests: >90%
Integration Tests: >80%
Agent Tests: >85%
E2E Tests: >70%
Load Tests: 1000 concurrent users

Key Metrics

- Agent uptime: >99.9%
- Error rate: <0.1%
- Latency p95: <5s
- Throughput: >100 requests/sec

Dependencies

- Sprint 5 completion

PHASE 3: USER INTERFACES

Sprint 7: Web Dashboard - Restaurant (Week 13-14)

Sprint Goal: Build restaurant manager dashboard

Team Composition (9 people)

Role	Count	Notes
Product Owner	1	Requirements
Tech Lead (Frontend)	1	UI architecture
Frontend Engineers	3	Dashboard components
UI/UX Designer	1	Design specs
Backend Engineer	1	API optimization
QA Engineer	1	UI testing
Documentation	1	User guides

Dashboard Features




RESTAURANT DASHBOARD





1. Overview
 - └ Today's spending
 - └ Active orders
 - └ Low stock alerts
 - └ Pending approvals
2. Inventory Management
 - └ Current stock
 - └ Par levels
 - └ Expiry dates
 - └ Auto-reorder suggestions
3. Supplier Chat Center
 - └ Active conversations
 - └ Pending quotes
 - └ Quick reply templates
 - └ Chat history
4. AI Cart Approvals
 - └ Pending carts
 - └ AI reasoning explanation
 - └ Edit capabilities
 - └ Approval history
5. Analytics & Reports
 - └ Food cost %
 - └ Supplier performance
 - └ Price trends
 - └ Savings achieved
 - └ Export reports
6. Settings
 - └ Preferences
 - └ Supplier management
 - └ Par level settings
 - └ Alert thresholds

Tech Stack

- Next.js 14 (React)
- TypeScript
- TailwindCSS + Shadcn/ui
- Zustand (state management)
- React Query (data fetching)
- WebSocket (real-time updates)

Deliverables

-  All dashboard screens (6 main sections)
-  Real-time updates via WebSocket
-  Data visualization (charts, trends)

-  Export functionality (PDF, CSV)
-  Responsive design (mobile, tablet, desktop)
-  Dark mode support
-  Accessibility (WCAG 2.1 AA)

Key Metrics

- Page load: <2s
- Interactivity: <1s
- Accessibility score: >95%
- Mobile responsiveness: 100%

Dependencies

- Sprint 2-6 completion (APIs, agents)

Sprint 8: Web Dashboard - Supplier (Week 15-16)

Sprint Goal: Build supplier sales dashboard

Team Composition (9 people)

Role	Count	Notes
Product Owner	1	Requirements
Tech Lead (Frontend)	1	UI architecture
Frontend Engineers	3	Dashboard components
UI/UX Designer	1	Design specs
Backend Engineer	1	API optimization
QA Engineer	1	UI testing
Documentation	1	User guides

Dashboard Features

 SUPPLIER DASHBOARD

1. Sales Performance

- └ Revenue today/month
- └ Order count
- └ Active chefs
- └ AI agent performance

2. Inventory Management








- └ Stock levels
- └ Expiry alerts
- └ Flash deal budgets
- └ Restock triggers

3. WhatsApp Chat Center

- └ Active conversations

	└ Pending quotes
	└ Response metrics
	└ Auto-response settings
4. AI Agent Performance	
	└ Quote generation rate
	└ Win rate
	└ Upsell conversion
	└ vs human agents
	└ Optimization suggestions
5. Collections & Billing	
	└ Invoices sent
	└ Payments received
	└ DSO (Days Sales Outstanding)
	└ Overdue alerts
	└ Payment tracking
6. Market Intelligence	
	└ Chef preferences
	└ Seasonal trends
	└ Competitor pricing
	└ Demand forecasting

Deliverables

-  All dashboard screens (6 sections)
-  Real-time sales updates
-  Flash deal management
-  Performance analytics
-  Collections management
-  Reporting & insights
-  Export capabilities

Key Metrics

- Page load: <2s
- Real-time latency: <1s
- Chart rendering: <500ms
- Analytics accuracy: >99%

Dependencies

- Sprint 7 completion (frontend framework)

Sprint 9: Mobile App (Week 17-18)

Sprint Goal: Build React Native mobile app (iOS + Android)

Team Composition (10 people)

Role	Count	Notes
Product Owner	1	Requirements

Mobile Lead	1	Architecture
React Native Engineers	3	iOS + Android
UI/UX Designer	1	Mobile design
Backend Engineer	1	API optimization
QA/Mobile Tester	2	Testing
Documentation	1	User guides

Mobile App Features

Restaurant App:

- Push notifications (quotes, alerts, updates)
- Quick actions (1-tap approve, quick order)
- Mini dashboard (today's spend, pending items)
- Approval workflows (carts, invoices)
- Camera tools (GRN photos, invoice scanning)
- Offline mode (cache data, auto-sync)

Supplier App:

- Push notifications (chat requests)
- Active chefs list
- Flash deal management
- Quick order processing
- Performance metrics
- Offline mode

Deliverables

-  React Native app (iOS + Android)
-  Push notification system
-  Offline-first architecture
-  Camera integration
-  OCR integration (Textextract)
-  Biometric authentication
-  App store deployment (TestFlight, Google Play)

Performance Targets

- App launch: <3s
- Screen transition: <500ms
- Offline functionality: 100%
- Crash rate: <0.01%

Dependencies

- Sprint 7-8 (web dashboards for feature parity)

PHASE 4: INTEGRATIONS & POLISH

Sprint 10: External API Integration (Week 19-20)

Sprint Goal: Integrate all external services and ensure data consistency

Team Composition (8 people)

Role	Count	Notes
Product Owner	1	Requirements
Tech Lead	1	Integration lead
Backend Engineers	2	API integrations
Integrations Specialist	1	External APIs
QA Engineer	2	Integration testing
DevOps Engineer	1	Deployment

Integration Checklist

☒

POS Systems

- └ Foodics API (OAuth 2.0)
- └ Oracle Simphony STSG2
- └ Generic CSV/Excel upload
- └ Real-time sync

☒

Payment Gateway

- └ Telr (primary)
- └ 2Checkout (fallback)
- └ Refund handling

☒

E-Invoicing

- └ Poppel Network (FTA compliance)
- └ ZATCA integration
- └ XML + PDF generation
- └ Audit logging

☒

Document Processing

- └ AWS Textract (OCR)
- └ Google Document AI (fallback)
- └ Handwriting recognition
- └ Quality validation

☒

Communication

- └ WhatsApp Business API
- └ SendGrid (email)
- └ Twilio (SMS)
- └ Push notifications

☒

Data & Analytics

- └ Analytics integration (GA4)
- └ Error tracking (Sentry)

- └ Performance monitoring (DataDog)
- └ Logging (ELK stack)

Deliverables

- ☒ All external service connectors
- ☒ Error handling & retries
- ☒ Data validation & reconciliation
- ☒ Audit logging
- ☒ Monitoring dashboards
- ☒ Integration test suite
- ☒ Runbooks for failures

Key Metrics

- Integration uptime: >99.5%
- Error rate: <0.1%
- Data consistency: 100%
- Sync latency: <5 minutes

Dependencies

- All previous sprints

Sprint 11: Testing & QA (Week 21-22)

Sprint Goal: Comprehensive testing, bug fixes, production readiness

Team Composition (10 people)

Role	Count	Notes
Product Owner	1	Requirements
Tech Lead	1	QA lead
QA Engineers	4	Manual + automation
Performance Engineer	1	Load testing
Security Engineer	1	Security testing
Backend Engineers	1	Bug fixes
DevOps Engineer	1	Infrastructure

Testing Checklist

- ☒ Functional Testing
 - └ Restaurant workflows (100%)
 - └ Supplier workflows (100%)
 - └ Admin workflows (100%)
 - └ Edge cases (100%)
- ☒ Integration Testing
 - └ WhatsApp ↔ Backend (100%)

- └ Web ↔ Backend (100%)
- └ Mobile ↔ Backend (100%)
- └ External services (100%)
- └ Data consistency (100%)

- ✓ Performance Testing
 - └ Load: 1,000 concurrent users
 - └ Stress: 5,000+ concurrent users
 - └ Soak: 24-hour runs
 - └ Spike: Sudden traffic increases
 - └ Response time p95: <5s

- ✓ Security Testing
 - └ OWASP Top 10 scan
 - └ SQL injection testing
 - └ XSS prevention
 - └ CSRF protection
 - └ Authentication/Authorization
 - └ Data encryption
 - └ Penetration testing

- ✓ Accessibility Testing
 - └ WCAG 2.1 AA compliance
 - └ Screen reader testing
 - └ Keyboard navigation
 - └ Color contrast
 - └ Mobile accessibility

- ✓ Browser & Device Testing
 - └ Chrome, Firefox, Safari, Edge
 - └ iOS 14+, Android 8+
 - └ Tablets & phones
 - └ Various screen sizes
 - └ Network conditions (4G, 5G, WiFi)

- ✓ UAT (User Acceptance Testing)
 - └ Restaurant managers (10+ users)
 - └ Suppliers (10+ users)
 - └ Admin team (5+ users)
 - └ Feedback incorporation

Deliverables

- ✓ Test automation suite (>1000 tests)
- ✓ Performance benchmarks
- ✓ Security audit report
- ✓ Accessibility audit report
- ✓ Bug tracking & resolution
- ✓ Production deployment guide
- ✓ Runbooks & playbooks
- ✓ Monitoring setup

Test Results Target

Automated Tests: >95% passing
Manual Testing: 100% checklist completion
Performance: All targets met (>99th percentile)
Security: Zero critical/high vulnerabilities
Accessibility: WCAG 2.1 AA compliant

Dependencies

- All previous sprints

Sprint 12: Launch & Monitoring (Week 23-24)

Sprint Goal: Production launch, monitoring setup, post-launch support

Team Composition (10 people)

Role	Count	Notes
Product Owner	1	Launch lead
Tech Lead	1	Technical oversight
DevOps Engineers	2	Deployment & monitoring
Backend Engineers	2	On-call support
Frontend Engineer	1	Production fixes
QA Engineer	1	Smoke testing
Customer Success	1	User onboarding
Documentation	1	Release notes

Launch Activities

Week 23 (Production Deployment)

Day 1-2: Final Checks

- └ All tests passing
- └ Security audit cleared
- └ Performance benchmarks met
- └ Disaster recovery tested
- └ Team trained

Day 3: Blue-Green Deployment

- └ Deploy to production (green)
- └ Run smoke tests
- └ Gradually shift traffic
- └ Monitor metrics
- └ Rollback plan ready

Day 4-5: Stabilization

- └ Monitor all metrics
- └ Fix critical issues only
- └ Customer support training
- └ Success metrics tracking

Week 24 (Post-Launch)

Day 1-3: Early Adopter Support

- └ Active customer support
- └ Bug monitoring
- └ Performance optimization
- └ Documentation updates
- └ User feedback collection

Day 4-5: Full Launch

- └ Open to all customers
- └ Marketing campaign
- └ Sales enablement
- └ User training webinars
- └ Success metrics review

Day 6-10: Stabilization

- └ Monitor for issues
- └ Hotfix deployments as needed
- └ Customer success calls
- └ Performance tuning
- └ Roadmap prioritization

Deliverables

- ☒ Production deployment
- ☒ Monitoring dashboards
- ☒ Alert configuration
- ☒ On-call rotation
- ☒ Runbooks for common issues
- ☒ Customer documentation
- ☒ Release notes
- ☒ Post-launch metrics

Launch Success Criteria

- ☒ System Uptime: >99.5% (Week 1)
- ☒ Error Rate: <0.1%
- ☒ Response Time p95: <5s
- ☒ Customer Satisfaction: >4.5/5
- ☒ Critical Issues: 0 (Week 1)
- ☒ High Issues: <3 (Week 1)
- ☒ User Adoption: >70% (restaurants + suppliers)

Post-Launch Support (Weeks 3-4)

- Bug fixes (critical first)
- Performance optimization

- User feedback incorporation
- Feature requests documentation
- Team retrospective

Dependencies

- Sprint 11 completion (all testing passed)

17

Full Timeline

PHASE 1: FOUNDATION

- └ Sprint 1 (Week 1–2): Architecture & Setup
- └ Sprint 2 (Week 3–4): Core Backend & APIs
- └ Sprint 3 (Week 5–6): Database & Integrations
- Milestone: Backend API complete ✓

PHASE 2: AI AGENTS

- └ Sprint 4 (Week 7–8): Agent Framework & Query Agent
- └ Sprint 5 (Week 9–10): Sales & Purchasing Agents
- └ Sprint 6 (Week 11–12): Agent Optimization & Testing
- Milestone: AI layer complete ✓

PHASE 3: USER INTERFACES

- └ Sprint 7 (Week 13–14): Web Dashboard – Restaurant
- └ Sprint 8 (Week 15–16): Web Dashboard – Supplier
- └ Sprint 9 (Week 17–18): Mobile App (iOS + Android)
- Milestone: All UIs complete ✓

PHASE 4: INTEGRATIONS & POLISH

- └ Sprint 10 (Week 19–20): External API Integration
- └ Sprint 11 (Week 21–22): Testing & QA
- └ Sprint 12 (Week 23–24): Launch & Monitoring
- Milestone: Platform LIVE ✓

Total Duration: 24 weeks (~6 months)



Team Capacity & Allocation

Core Team (Constant)

Product Owner: 1 (100% all sprints)
Scrum Master: 1 (100% all sprints)
Tech Lead: 1 (100% all sprints)
Documentation: 1 (50% all sprints, 100% during Phase 4)

Role Allocation by Phase

Phase 1 (Foundation)

- Backend Engineers: 3-4 FTE
- DevOps Engineer: 1 FTE

- Database Engineer: 0.5 FTE (Sprint 3)
- Total: ~8 people

Phase 2 (AI Agents)

- Backend Engineers: 2-3 FTE
- AI/ML Engineers: 2 FTE
- QA Engineers: 1 FTE
- Total: ~8 people

Phase 3 (User Interfaces)

- Frontend Engineers: 3 FTE
- UI/UX Designers: 1-2 FTE
- Backend Engineers: 1 FTE
- QA Engineers: 1 FTE
- Total: ~8-9 people

Phase 4 (Integration & Polish)






- Backend Engineers: 2 FTE
- QA Engineers: 3-4 FTE
- DevOps Engineers: 1-2 FTE
- Performance Engineer: 1 FTE
- Security Engineer: 1 FTE
- Total: ~9-10 people

Team Ramp-Up/Down Profile

Week 1-6: 8 people (Foundation)
Week 7-12: 10 people (Agents ramp-up)
Week 13-18: 12 people (UI ramp-up, Backend ramp-down)
Week 19-24: 10 people (QA/DevOps ramp-up, Frontend ramp-down)





Success Metrics by Sprint

Sprint Goals (Each Sprint)





1.  Planned work completed (90%+ sprint commitment)
2.  Zero critical bugs in code review
3.  Test coverage maintained (>80%)
4.  Documentation updated
5.  Team velocity consistent

Phase Milestones





Phase 1 Milestone (End of Sprint 3)

-  Backend APIs fully functional
-  All external integrations connected
-  Database tested at scale
-  CI/CD pipeline automated
- **Go/No-Go:** Can we build on this? ✓





Phase 2 Milestone (End of Sprint 6)

-  All agents production-ready
-  Agent performance targets met
-  Agent testing comprehensive
-  Monitoring dashboards live
- **Go/No-Go:** Can we release agents? ✓

Phase 3 Milestone (End of Sprint 9)

-  All UIs feature-complete
-  Web dashboards fully responsive
-  Mobile app in TestFlight/Beta
-  Real-time sync working
- **Go/No-Go:** Can we integrate everything? ✓

Phase 4 Milestone (End of Sprint 12)

-  Platform live in production
-  All integrations working
-  Monitoring & alerts active
-  Customer success team ready
- **Go/No-Go:** LAUNCH! 🚀

 **Risk Management**

High-Risk Items (Proactive Mitigation)

Risk	Probability	Impact	Sprint	Mitigation
Scope Creep	High	High	All	Lock scope in Sprint planning, change control board
AI Agent Performance	Medium	High	4-6	Early optimization, allocate ML engineers 50% from Sprint 1
Integration Delays	Medium	High	10	Pre-integrate external APIs in Sprint 2
Team Turnover	Low	High	All	Knowledge sharing, pair programming, documentation
Infrastructure Issues	Low	High	1	Pre-allocate DevOps resources, chaos testing
Performance Bottlenecks	Medium	Medium	11	Load testing in Sprint 2, continuous profiling

Contingency Plans

- If Phase 1 delayed by 1 week:
 - └ Compress Phase 2 (parallel work on agents)
 - └ Impact: Minor (2-3 day slip to Phase 3)
- If AI agents underperform (Sprint 5):
 - └ Allocate additional AI/ML resources (Sprint 6)

- └ Defer Phase 3 start by 1 week
 - └ Impact: 1 week slip to launch
 - If Phase 3 delayed (UI complexity):
 - └ MVP mode: Ship core dashboards only
 - └ Full dashboards in Sprint 12
 - └ Impact: 2-3 week slip to launch
 - If launch issues (Sprint 12):
 - └ Rollback to previous version
 - └ Fix in production with careful deployment
 - └ Impact: Depends on issue severity
-

Reporting & Communication

Weekly Cadence

- **Monday 9 AM:** Sprint Planning (1 hour)
- **Daily 10 AM:** Standup (15 min)
- **Wednesday 2 PM:** Mid-sprint sync (30 min)
- **Friday 4 PM:** Sprint Review & Demo (1 hour)
- **Friday 5 PM:** Retrospective (1 hour)

Stakeholder Updates

- **Weekly:** Executive summary (15 min)
- **Bi-weekly:** Detailed status review (30 min)
- **Monthly:** Business metrics & roadmap (1 hour)

Documentation

- Sprint backlogs (Jira)
 - Sprint burndown charts
 - Velocity tracking
 - Risk register (updated weekly)
 - Release notes (per sprint)
-

Team Development

Knowledge Transfer

- Pair programming (20% of sprint time)
- Code reviews (all code reviewed by 2+ people)
- Tech talks (Friday 30 min, rotating presenters)
- Documentation (living docs, updated weekly)

Training & Onboarding

- Sprint 1: New team members onboarded
 - Weeks 1-2: Architecture deep-dive
 - Weeks 3-4: System setup training
 - Ongoing: Skill development (2 hours/week per person)
-

✔ Sprint Completion Checklist

Each sprint must have:

- ✔ All planned work completed or carried forward
- ✔ Code reviewed & merged
- ✔ Tests written & passing (>80% coverage)
- ✔ Documentation updated
- ✔ No critical bugs in staging
- ✔ Deployment ready
- ✔ Stakeholders notified
- ✔ Retrospective completed
- ✔ Next sprint planned

Summary: 12-Sprint Roadmap

24 Weeks | 12 Sprints | 12-15 Team Members

- └ Phase 1 (6 weeks): Foundation & Backend
- └ Phase 2 (6 weeks): AI Agents
- └ Phase 3 (6 weeks): User Interfaces
- └ Phase 4 (6 weeks): Integration & Launch

Key Deliverables:

- ✓ Production-ready backend
- ✓ 9 AI agents (7 restaurant + 2 supplier)
- ✓ Web dashboards (restaurant + supplier)
- ✓ Mobile app (iOS + Android)
- ✓ WhatsApp integration
- ✓ Full external integrations
- ✓ Comprehensive testing
- ✓ LIVE platform launch

Target Launch: Week 24 (6-month timeline)

Ready to kickoff? Sprint 1 starts on [Start Date] 🚀