Development Roadmap & Implementation Guide

MES/SCADA RAG System

SPRINT PLANNING OVERVIEW

Total Estimated Duration: 8-10 týdnů
Team Size: 1-2 developers
Methodology: Agile s týdenními sprinty

PHASE 1: FOUNDATION (Týdny 1-3)

Sprint 1: Infrastructure & Architecture Setup

1.1 Development Environment

Deliverables:

- Docker Compose setup s všemi services
- Local development environment
- Basic CI/CD pipeline (GitHub Actions)
- Environment configuration management

Acceptance Criteria:

- Jeden `docker-compose up` spustí celý stack
- Všechny services jsou healthy
- Hot reload funguje pro development
- Environment variables správně načteny

Technical Tasks:

```yaml
# docker-compose.yml structure
services:
  - nginx (reverse proxy)
  - fastapi (backend API)
  - postgres (metadata DB)
  - qdrant (vector DB)
  - redis (cache/sessions)
  - ollama (embeddings)
  - frontend (static files)
```

1.2 Database Schema Design

Deliverables:

- PostgreSQL schema pro Manufacturing Hierarchy
- Document metadata schema
- User management schema
- Migration system setup

Core Tables:

- hierarchy_nodes (flexibilní ISA-95 struktura)
- documents (metadata + file references)
- document_node_associations (M:N relationship)
- users a roles (authentication/authorization)
- audit_log (compliance tracking)

1.3 API Contract Definition

Deliverables:

- OpenAPI specification
- FastAPI project structure
- Endpoint stubs s dokumentací
- Request/Response models (Pydantic)

API Endpoints (MVP):

```
/api/v1/hierarchy/    # Manufacturing structure CRUD
/api/v1/documents/    # Document management
/api/v1/search/       # RAG search endpoint
/api/v1/auth/         # Authentication
/api/v1/admin/        # System administration
```

Sprint 2: Authentication & Authorization Framework

2.1 Authentication System

Deliverables:

- JWT-based session management
- Local user authentication
- SSO integration framework (SAML/OIDC ready)
- Password security (hashing, policies)

Implementation Priorities:

1. Local authentication (immediate)
2. SSO framework (extensible for customer integration)
3. Session management (Redis-backed)
4. Security middleware (rate limiting, CORS)

2.2 Authorization System

Deliverables:

- Role-based access control (RBAC)
- Hierarchical permissions (děděné přes ISA-95 structure)
- Document-level permissions
- Admin interface pro role management

Permission Model:

- **System Admin:** Full system access
- **Hierarchy Manager:** Can modify manufacturing structure
- **Document Manager:** Can upload/manage documents
- **User:** Read access based on hierarchy assignment

Sprint 3: Core Data Models & Business Logic

3.1 Manufacturing Hierarchy Manager

Deliverables:

- Flexible hierarchy implementation (ne strikt ISA-95)
- CRUD operations s validation
- Tree manipulation utilities
- Hierarchy path resolution

Business Rules:

- Libovolná hloubka hierarchie
- Pojmenované level types (ne jen ISA-95)
- Validace při vytváření/přesouvání nodes
- Soft delete s audit trail

3.2 Document Management Core

Deliverables:

- File upload handling
- Metadata extraction
- Document-hierarchy association logic
- File storage abstraction layer

File Storage Decision: Pro MVP zvolíme filesystem approach s organizovanou strukturou:

```
/uploads/
  /{year}/
    /{month}/
      /{hash}/
        original_filename.ext
        metadata.json
```

PHASE 2: CORE MVP (Týdny 4-7)

Sprint 4: Document Processing Pipeline

4.1 Text Extraction Engine

Deliverables:

- Multi-format text extraction (PDF, DOCX, XLSX, TXT)
- Content cleaning a normalization
- Metadata extraction (author, creation date, etc.)
- Error handling pro corrupt files

Technology Stack:

- **PDF:** PyMuPDF (rychlé, reliable)
- **Word:** python-docx
- **Excel:** pandas + openpyxl
- **Text:** charset detection

4.2 RAG Pipeline Foundation

Deliverables:

- Ollama integration pro embeddings
- Document chunking strategy
- **Qdrant integration**

- Async processing pipeline

Chunking Strategy:

- Semantic chunking (per paragraph/section)
- Chunk size: 512-1024 tokens
- Overlap: 50-100 tokens
- Preserve document structure metadata

Sprint 5: Search Engine Implementation

5.1 RAG Search Core

Deliverables:

- Vector similarity search
- Hybrid search (vector + keyword)
- Result ranking algorithm
- Query preprocessing

Search Algorithm:

1. Query preprocessing (cleaning, expansion)
2. Generate query embeddings
3. Vector similarity search in **Qdrant**
4. Apply hierarchy filters
5. Rank results by relevance + context
6. Format response s citations

5.2 Context-Aware Search

Deliverables:

- Hierarchical context integration
- User permission filtering
- Result aggregation across hierarchy levels
- Search analytics tracking

Sprint 6: Frontend Development

6.1 Core UI Components

Deliverables:

- Chat interface (main user interaction)
- Hierarchy tree browser
- Document upload form
- Search results display

UI Framework:

- Vanilla JavaScript + Tailwind CSS
- Component-based architecture
- Mobile-responsive design
- Accessibility compliance (WCAG 2.1)

6.2 Admin Interface

Deliverables:

- Hierarchy management interface
- Document management dashboard
- User/role administration
- System monitoring dashboard

Sprint 7: Integration & Polish

7.1 End-to-End Integration

Deliverables:

- Complete user workflows testing
- API error handling
- Loading states a progress indicators
- Form validation

7.2 Performance Optimization

Deliverables:

- Database query optimization
- Caching strategy implementation
- Async processing improvements
- Frontend bundle optimization

PHASE 3: ENHANCEMENT & PRODUCTION PREP (Týdny 8-10)

Sprint 8: Advanced Features

8.1 Advanced Search Features

- Faceted search (filter by document type, date, author)
- Search within hierarchy context
- Saved searches
- Search suggestions

8.2 Document Lifecycle

- Document versioning
- Approval workflows (basic)
- Archive/restore functionality
- Bulk operations

Sprint 9: Production Readiness

9.1 Monitoring & Observability

Deliverables:

- Application metrics (Prometheus)
- Health check endpoints
- Error tracking (Sentry integration)
- Performance monitoring

9.2 Security Hardening

Deliverables:

- Security headers implementation
- Input validation hardening
- Rate limiting
- Security audit checklist

Sprint 10: Deployment & Documentation

10.1 Production Deployment

Deliverables:

- Production Docker Compose
- Environment-specific configs

- Backup/restore procedures
- Migration scripts

10.2 Documentation Package

Deliverables:

- Installation guide
- User manual
- API documentation
- Troubleshooting guide

DEVELOPMENT RULES & STANDARDS

Code Quality Standards

```python
# Naming conventions
- snake_case pro variables a functions
- PascalCase pro classes
- UPPER_CASE pro constants
- Descriptive names (ne abbreviations)

# File organization
/src/
  /api/          # FastAPI routes
  /core/         # Business logic
  /models/       # Database models
  /schemas/      # Pydantic models
  /services/     # Business services
  /utils/        # Utilities
```

Git Workflow

- **Feature branches:** feature/sprint-X-feature-name
- **Commit messages:** Conventional commits format
- **Pull requests:** Required pro všechny changes
- **Code review:** Minimálně jeden approve

Testing Strategy

- **Unit tests:** Každá business logic function
- **Integration tests:** API endpoints

- **E2E tests:** Critical user workflows
- **Performance tests:** Load testing před production

Documentation Requirements

- **Code comments:** Složitá business logic
- **API docs:** OpenAPI/Swagger automated
- **Architecture docs:** Updated s changes
- **User docs:** Step-by-step guides

RISK MITIGATION STRATEGIES

Technical Risks

1. **Ollama Performance Issues**
   - Mitigation: Benchmark early, fallback na cloud embeddings
   - Detection: Response time monitoring
2. **Large File Processing**
   - Mitigation: Async processing, chunking strategy
   - Detection: Processing time metrics
3. **Search Relevance**
   - Mitigation: A/B testing different algorithms
   - Detection: User feedback tracking
4. **Qdrant Scaling**
   - Mitigation: Monitor collection size, implement sharding if needed
   - Detection: Query performance metrics

Schedule Risks

1. **Scope Creep**
   - Mitigation: Strict MVP definition, change request process
2. **Technology Learning Curve**
   - Mitigation: Proof of concepts pro unknown tech
3. **Integration Complexity**
   - Mitigation: Early customer SSO testing, fallback options

SUCCESS CRITERIA PER SPRINT

Technical Milestones

- **Sprint 1:** Complete local development environment

- **Sprint 3:** Basic CRUD operations working
- **Sprint 5:** First successful RAG query
- **Sprint 7:** Complete user workflow end-to-end
- **Sprint 10:** Production-ready deployment

Quality Gates

- All tests passing
- Code coverage >80%
- Security scan passing
- Performance benchmarks met
- Documentation complete

NEXT IMMEDIATE ACTION

Začneme Sprint 1 s Docker Compose setup. Chcete pokračovat s konkrétní implementací, nebo potřebujete upřesnit nějaké aspekty návrhu?