Test Environment

> **Source File:** generated-documents\quality-assurance\test-environment.md
> **Generated:** 30/07/2025 at 07:00:34
> **Generated by:** Requirements Gathering Agent - PDF Converter

Test Environment

Generated by adpa-enterprise-framework-automation v3.2.0
Category: quality-assurance
Generated: 2025-07-14T21:15:36.363Z
Description: Test environment setup and configuration

Test Environment Setup and Management

=== PROJECT README ===

ADPA - Advanced Document Processing & Automation Framework

> ***Previously known as Requirements Gathering Agent (RGA)***

ADPA is a modular, standards-compliant enterprise automation framework for AI-powered document generation, project management, and business analysis. Built with TypeScript and Node.js, it provides both CLI and REST API interfaces for generating professional documentation following industry standards including BABOK v3, PMBOK 7th Edition, and DMBOK 2.0.

🚀 Key Features

Enterprise Standards Compliance

- 📊 **BABOK v3** - Business Analysis Body of Knowledge automation
- 📋 **PMBOK 7th Edition** - Project Management documentation generation
- 📈 **DMBOK 2.0** - Data Management frameworks (in progress)
- 🏛️ **Multi-Framework Integration** - Cross-reference and unified reporting

AI-Powered Generation

- 🤖 **Multi-Provider AI Support** - OpenAI, Google AI, GitHub Copilot, Ollama
- 🧠 **Intelligent Context Management** - Smart context injection and processing
- 📝 **Professional Document Generation** - Standards-compliant business documents

- 🔄 **Automated Workflows** - End-to-end document generation pipelines

Enterprise Integration

- 🌐 **Production-Ready REST API** - TypeSpec-generated OpenAPI specifications
- 📚 **Confluence Integration** - Direct publishing to Atlassian Confluence
- 📊 **SharePoint Integration** - Microsoft SharePoint document management
- **Adobe Document Services** - Professional PDF generation and document intelligence
- 🔧 **CLI & Web Interface** - Multiple interaction modes

Compliance & Security

- 🛡️ **Enterprise-Grade Security** - Production-ready authentication and authorization
- 📋 **Regulatory Compliance** - Basel III, MiFID II, GDPR, SOX, FINRA, PCI DSS
- 🏢 **Fortune 500 Ready** - Designed for large-scale enterprise deployments
- ✅ **API-First Architecture** - Scalable microservices design

📦 Installation

NPM Package (Recommended)

```
npm install -g adpa-enterprise-framework-automation
```

From Source

```
git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run build
```

Docker (Coming Soon)

```
docker pull adpa/enterprise-framework:latest
```

🎯 Quick Start

1. CLI Usage

```
# Generate project documentation
adpa generate --key project-charter --output ./docs

# Start the API server
adpa-api

# Initialize Confluence integration
adpa confluence init

# Initialize SharePoint integration
adpa sharepoint init
```

2. API Server

```
# Start the Express.js API server
npm run api:start

# Access API documentation
open http://localhost:3000/api-docs
```

3. Admin Web Interface

```
# Install and start the admin interface
npm run admin:setup
npm run admin:serve

# Access at http://localhost:3001
```

🛠️ Configuration

Environment Setup

```
# Copy environment template
cp .env.example .env

# Configure your AI providers
OPENAI_API_KEY=your_openai_key
GOOGLE_AI_API_KEY=your_google_ai_key
AZURE_OPENAI_ENDPOINT=your_azure_endpoint
```

AI Provider Configuration

ADPA supports multiple AI providers with automatic failover:

```
// Supported providers
- OpenAI (GPT-4, GPT-3.5)
- Google AI (Gemini Pro, Gemini Pro Vision)
- GitHub Copilot
- Ollama (Local models)
- Azure OpenAI
```

📚 Framework Support

BABOK v3 (Business Analysis)

✅ **Production Ready**

- Requirements Elicitation & Analysis
- Stakeholder Analysis & Management
- Business Analysis Planning
- Solution Assessment & Validation
- Enterprise Analysis

PMBOK 7th Edition (Project Management)

✅ **Implemented**

- Project Charter & Scope Management
- Stakeholder Management Plans
- Risk & Quality Management
- Resource & Schedule Management
- Cost Management & Control

DMBOK 2.0 (Data Management)

🚧 **In Progress**

- Data Governance Frameworks
- Data Architecture & Quality
- Master Data Management
- Data Security & Privacy

🏗️ Architecture

Core Components

```
ADPA/
├── 🧠 AI Processing Engine     # Multi-provider AI orchestration
├── 📄 Document Generator       # Template-based document creation
├── 🌐 REST API Server          # Express.js with TypeSpec specs
```

```
├── 💻 CLI Interface           # Yargs-based command line tools
├── 🔌 Integration Layer       # Adobe, Confluence, SharePoint, VCS
├── 🎛️ Admin Interface         # Next.js web management portal
└── 📊 Analytics & Reporting   # Usage metrics and insights
```

Technology Stack

- **Backend**: Node.js 18+, TypeScript 5.7+, Express.js
- **AI Integration**: OpenAI, Google AI, GitHub Copilot, Ollama
- **API**: TypeSpec, OpenAPI 3.0, Swagger UI
- **Frontend**: Next.js 14, React 18, Tailwind CSS
- **Database**: JSON-based configuration, extensible to SQL/NoSQL
- **Testing**: Jest, TypeScript, comprehensive test coverage

📖 Usage Examples

Document Generation

```
# Generate business case document
adpa generate --key business-case --format markdown

# Generate complete project charter
adpa generate --category project-charter --output ./project-docs

# Generate stakeholder analysis
adpa generate --key stakeholder-analysis --format json
```

API Usage

```
// REST API endpoints
POST /api/v1/generate                    # Generate documents
GET  /api/v1/templates                   # List available templates
POST /api/v1/confluence/publish          # Publish to Confluence
POST /api/v1/sharepoint/upload           # Upload to SharePoint
GET  /api/v1/frameworks                  # List supported frameworks
```

Integration Examples

```
# Adobe Document Services integration
npm run adobe:setup                    # Configure Adobe credentials
npm run adobe:demo-generation          # Run document generation demo
npm run adobe:example-basic            # Basic PDF generation example

# Confluence integration
adpa confluence oauth2 login
adpa confluence publish --document ./docs/project-charter.md

# SharePoint integration
adpa sharepoint oauth2 login
adpa sharepoint upload --folder "Project Documents" --file ./docs/

# Version control integration
adpa vcs commit --message "Generated project documentation"
adpa vcs push --remote origin
```

🧪 Testing

```
# Run all tests
npm test

# Test specific providers
npm run test:azure
npm run test:github
npm run test:ollama

# Performance testing
npm run test:performance

# Integration testing
npm run test:integration
```

🏢 Enterprise Features

Compliance Standards

- **Financial**: Basel III, MiFID II, FINRA, CFTC, FCA, BaFin
- **Security**: GDPR, SOX, PCI DSS, ISO 27001, ISO 9001
- **Industry**: Healthcare (HIPAA), Government (FedRAMP)

Enterprise Integration

- **Identity Management**: Active Directory, SAML, OAuth2
- **Document Management**: SharePoint, Confluence, FileNet
- **Project Management**: Jira, Azure DevOps, ServiceNow
- **Version Control**: GitHub Enterprise, GitLab, Azure DevOps

Scalability & Performance

- **Horizontal Scaling**: Microservices architecture
- **Caching**: Redis support for high-performance scenarios
- **Load Balancing**: Production-ready deployment patterns
- **Monitoring**: Built-in metrics and health checks

📁 Project Structure

```
requirements-gathering-agent/
├── 📁 src/                          # TypeScript source code
│   ├── 🎯 cli.ts                    # Main CLI entry point
│   ├── 🌐 server.ts                 # Express.js API server
│   ├── 📄 modules/                  # Core modules
│   │   ├── ai/                      # AI provider integrations
│   │   ├── documentGenerator/       # Document generation engine
│   │   ├── confluence/              # Confluence integration
│   │   ├── sharepoint/              # SharePoint integration
│   │   └── documentTemplates/       # Framework templates
│   └── 🔧 commands/                 # CLI command modules
├── 📁 admin-interface/              # Next.js admin portal
├── 📁 api-specs/                    # TypeSpec API specifications
├── 📁 docs/                         # Comprehensive documentation
├── 📁 test/                         # Test suites
```

```
├── 📁 generated-documents/          # Output directory
└── 📁 dist/                         # Compiled JavaScript
```

🤝 Contributing

We welcome contributions! Please see our Contributing Guide for details.

Development Setup

```
git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run dev          # Start development mode
npm run build        # Build for production
npm test           # Run tests
```

Code Standards

- **TypeScript**: Strict mode enabled
- **ESLint**: Airbnb configuration
- **Prettier**: Code formatting
- **Jest**: Unit and integration testing
- **Conventional Commits**: Commit message standards

📋 Roadmap

Q1 2025

- ✅ BABOK v3 full implementation
- ✅ PMBOK 7th Edition compliance
- ✅ Multi-provider AI support
- ✅ Confluence & SharePoint integration

Q2 2025

- 🚧 DMBOK 2.0 implementation
- 🔄 Docker containerization
- 🔄 Kubernetes deployment templates
- 🔄 Advanced analytics dashboard

Q3 2025

- 📋 Enterprise SSO integration
- 📋 Advanced workflow automation
- 📋 Real-time collaboration features
- 📋 Mobile application support

📞 Support & Documentation

- 📖 **Full Documentation**: GitHub Wiki
- 🦎 **Issue Tracking**: GitHub Issues
- 💬 **Community**: GitHub Discussions
- 📧 **Enterprise Support**: Contact Us

📄 License

This project is licensed under the MIT License - see the LICENSE file for details.

🙏 Acknowledgments

- **Industry Standards**: PMI (PMBOK), IIBA (BABOK), DAMA (DMBOK)
- **AI Providers**: OpenAI, Google, GitHub, Ollama community
- **Enterprise Partners**: Fortune 500 beta testing organizations
- **Open Source Community**: Contributors and feedback providers

---

Built with ❤️ for Enterprise Automation

🌟 Star us on GitHub | 📦 npm Package | 📖 Documentation

=== PROJECT METADATA ===

Name: adpa-enterprise-framework-automation

Description: Modular, standards-compliant Node.js/TypeScript automation framework for enterprise requirements, project, and data management. Provides CLI and API for BABOK v3, PMBOK 7th Edition, and DMBOK 2.0 (in progress). Production-ready Express.js API with TypeSpec architecture. Designed for secure, scalable, and maintainable enterprise automation.

Version: 3.2.0

Dependencies: @adobe/pdfservices-node-sdk, @azure-rest/ai-inference, @azure/identity, @azure/msal-node, @azure/openai, @google/generative-ai, @microsoft/microsoft-graph-client, axios, bcryptjs, compression, cors, dotenv, express, express-rate-limit, express-validator, express-winston, form-data, glob, helmet, joi, jsonwebtoken, morgan, multer, node-fetch, openai, swagger-ui-express, ts-node, uuid, winston, yargs, zod

Dev Dependencies: @jest/globals, @redocly/cli, @types/bcryptjs, @types/compression, @types/cors, @types/express, @types/glob, @types/jest, @types/jsonwebtoken, @types/morgan, @types/multer, @types/node, @types/node-fetch, @types/swagger-ui-express, @types/uuid, @typespec/compiler, @typespec/http, @typespec/json-schema, @typespec/openapi3, @typespec/rest, ajv, jest, rimraf, ts-jest, typescript, webpack-cli

Available Scripts: build, copy-configs, start, api:start, dev, clean, test, test:providers, test:performance, test:azure, test:github, test:ollama, test:failover, test:unit, prepublishOnly, admin:install, admin:dev, admin:build, admin:start, admin:setup, admin:serve, confluence:init, confluence:test, confluence:oauth2:login, confluence:oauth2:status, confluence:oauth2:debug, confluence:publish, confluence:status, sharepoint:init, sharepoint:test, sharepoint:oauth2:login, sharepoint:oauth2:status, sharepoint:oauth2:debug, sharepoint:publish, sharepoint:status, api:compile, api:watch, api:format, api:lint, api:docs, api:serve-docs, api:demo, api:server, babok:generate, pmbok:generate, dmbok:generate, framework:multi

=== PHASE-2-IMPLEMENTATION-GUIDE.MD (documentation) ===
Path: docs\ADOBE\PHASE-2-IMPLEMENTATION-GUIDE.md
Relevance Score: 95

Adobe Creative Suite Phase 2 Implementation Guide

Date: July 8, 2025
Status: 📋 **READY FOR IMPLEMENTATION**
Prerequisites: ✅ Phase 1 Complete (58 PDFs generated successfully)

🎯 Phase 2 Objectives

Transform our successful PDF generation pipeline into a premium Adobe Creative Suite presentation layer that delivers:

- **Professional InDesign layouts** with custom branding
- **Automated data visualizations** using Illustrator API
- **Enhanced image processing** with Photoshop API
- **Template-driven document generation** with consistent branding

🏗️ Technical Architecture for Phase 2

Current State (Phase 1) ✅

```
Markdown → Puppeteer → Professional PDF
├── Professional styling
├── Corporate typography
├── Metadata and attribution
└── Print-ready output
```

Target State (Phase 2) 🎯

```
Markdown → Content Analysis → Template Selection → Adobe Creative API
├── InDesign Server (Layout & Typography)
├── Illustrator API (Charts & Infographics)
├── Photoshop API (Image Enhancement)
└── Document Generation (Template Processing)
```

📋 Implementation Milestones

Milestone 1: Adobe Creative SDK Setup (Priority 1)

1.1 Authentication & Credentials

Files to Create:

- `src/adobe/creative-suite/authenticator.ts`
- `src/adobe/creative-suite/config.ts`
- `.env.adobe.creative`

Implementation Steps:

1. **Adobe Creative SDK Registration**

```
# Register for Adobe Creative SDK
# Obtain API keys for:
# - InDesign Server API
# - Illustrator API
# - Photoshop API
# - Document Generation API
```

2. **Authentication Setup**

```
// src/adobe/creative-suite/authenticator.ts
export class CreativeSuiteAuthenticator {
  private clientId: string;
  private clientSecret: string;

```

```
  async authenticate(): Promise<string> {
    // Implement OAuth 2.0 flow for Creative Suite APIs
  }
}
```

... [truncated]

=== ARCHITECTURE.MD (development) ===
Path: docs\ARCHITECTURE.md
Relevance Score: 95

Requirements Gathering Agent - Architecture Documentation

Overview

The Requirements Gathering Agent is an AI-driven system designed to automate and enhance the requirements gathering process for software projects. It leverages multiple AI providers and context management techniques to generate comprehensive project documentation, user stories, and strategic planning artifacts.

System Architecture

Core Components

1. Context Management System

- **Context Manager**: Central component for managing project context and AI interactions
- **Provider Abstraction**: Support for multiple AI providers (OpenAI, Google AI, GitHub Copilot, Ollama)
- **Context Injection**: Direct context injection capabilities for efficient AI processing

2. AI Provider Integration

- **Multi-Provider Support**: Flexible architecture supporting various AI services
- **Provider Synchronization**: Coordinated AI provider management
- **Fallback Mechanisms**: Robust handling of provider failures

3. Document Generation Engine

- **Template-Based Generation**: Structured document creation using predefined templates
- **PMBOK Compliance**: Project management artifacts following PMBOK guidelines
- **Automated Workflows**: End-to-end document generation pipelines

4. CLI Interface

- **Command-Line Tools**: `cli.ts` and `cli-main.ts` for system interaction
- **Batch Processing**: Support for bulk document generation
- **Configuration Management**: Flexible configuration options

Technology Stack

Core Technologies

- **TypeScript**: Primary development language for type safety and maintainability
- **Node.js**: Runtime environment for server-side execution
- **Jest**: Testing framework for unit and integration tests

AI Integration

- **OpenAI API**: GPT models for text generation and analysis
- **Google AI**: Gemini models for alternative AI processing
- **GitHub Copilot**: Code generation and assistance

- **Ollama**:
  ... [truncated]

=== API-TESTING-COMPREHENSIVE-SUMMARY.MD (development) ===
Path: docs\AZURE\API-TESTING-COMPREHENSIVE-SUMMARY.md
Relevance Score: 95

ADPA API Testing Comprehensive Summary

Test Session Report - June 22, 2025

🎯 TESTING OVERVIEW

Duration: 1 hour testing session
API Server: Express.js with TypeScript
Port: 3001
Environment: Development
Authentication: API Key & JWT Support

✅ SUCCESSFUL TESTS

1. **Health Endpoints** - ALL PASSED ✓

- **Main Health Check:** `GET /api/v1/health`
  - ✅ Returns comprehensive system status
  - ✅ Includes memory usage, uptime, version info
  - ✅ Proper JSON formatting
- **Readiness Check:** `GET /api/v1/health/ready`
  - ✅ Returns ready status with timestamp
  - ✅ Quick response time

2. **Authentication & Security - ALL PASSED ✓**

- **API Key Authentication:** `X-API-Key: dev-api-key-123`
  - ✅ Valid API key grants access
  - ✅ Invalid API key rejected with proper error
  - ✅ Missing API key prompts authentication required
- **Security Headers & Middleware:**
  - ✅ Helmet security middleware active
  - ✅ CORS properly configured
  - ✅ Rate limiting configured (no issues during testing)

3. **Templates API - ALL PASSED ✓**

- **Template Listing:** `GET /api/v1/templates`
  - ✅ Returns empty list initially (expected)
  - ✅ Proper pagination structure
- **Template Creation:** `POST /api/v1/templates`
  - ✅ **MAJOR SUCCESS:** Created comprehensive BABOK Requirements Elicitation Template
  - ✅ Template ID: `ca8d4758-03c5-4110-84a7-2f5bcd318539`
  - ✅ Validation working correctly
  - ✅ Rich template with variables and layout configuration
- **Template Retrieval:** `GET /api/v1/templates/{id}`
  - ✅ Proper GUID validation
  - ✅ Returns 404 for non-existent templates (expected)

4. **Documents API - ALL PASSED ✓**

- **Document Jobs Listing:** `GET /api/v1/documents/jobs`
  - ✅ Returns proper pagination structure
  - ✅ Authentication required and working

- **Document Conversion:** `POST /api/v1/documents/convert`
    - ✅ **MAJOR SUCCESS:** Ge
      ... [truncated]

=== AZURE-PORTAL-API-CENTER-SETUP-GUIDE.MD (primary) ===
Path: docs\AZURE\AZURE-PORTAL-API-CENTER-SETUP-GUIDE.md
Relevance Score: 95

Azure Portal API Center Setup Guide

Standards Compliance & Deviation Analysis API

🎯 Portal-Based Deployment Strategy

Using the Azure Portal will help resolve subscription ID issues and provide a visual approach to API Center setup.

Step 1: Access Azure Portal

Navigate to Azure API Center

1. **Open**: Azure Portal
2. **Search**: "API Center" in the top search bar
3. **Select**: "API Centers" from the results

Verify Subscription Access

- **Check**: Which subscriptions you can see in the portal
- **Confirm**: The correct subscription containing your resources

- **Note**: The actual subscription ID for CLI alignment

Step 2: Create/Verify API Center Instance

Option A: Create New API Center

If `svc-api-center` doesn't exist:

1. **Click**: "Create API Center"
2. **Subscription**: Select the correct active subscription
3. **Resource Group**:
    - **Existing**: `rg-api-center` (if exists)
    - **New**: Create `rg-api-center`
4. **API Center Name**: `svc-api-center`
5. **Region**: **West Europe** (`westeu`)
6. **Pricing Tier**: Start with Standard
7. **Click**: "Review + Create" → "Create"

Option B: Use Existing API Center

If it already exists:

1. **Navigate**: to existing `svc-api-center`
2. **Note**: Subscription ID and Resource Group (`rg-api-center`)
3. **Verify**: Access and permissions

Step 3: Create APIs via Portal

3.1 Create Echo API

1. **Navigate**: to your `svc-api-center` API Center instance
2. **Click**: "APIs" in the left menu
3. **Click**: "Create API"
4. **Fill Details**:
    - **API ID**: `echo-api`

- **Title**: `Echo API`
- **Type**: `REST`
- **Description**: `Simple echo API for testing`

5. **Click**: "Create"

3.2 Create Standards Compliance API

1. **Click**: "Create API" again
2. **Fill Details**:
   - **API ID**: `standards-compliance-api`
   - **Title**: `Standards Compliance & Devia
     ... [truncated]

=== AZURE-PORTAL-API-REGISTRATION-GUIDE.MD (development) ===
Path: docs\AZURE\AZURE-PORTAL-API-REGISTRATION-GUIDE.md
Relevance Score: 95

Azure Portal API Registration Guide

Manual API Center Setup - No CLI Required

🎯 Why Portal Registration is Perfect for You

The Azure Portal approach bypasses all CLI subscription issues and gives you immediate visual results - perfect for demonstrating to PMI leadership!

Step 1: Access Azure Portal

Navigate to API Centers

1. **Open**: Azure Portal
2. **Sign in** with your Azure account
3. **Search**: "API Center" in the top search bar
4. **Select**: "API Centers" from the dropdown

Find Your API Center

- **Look for**: `svc-api-center` in `rg-api-center`
- **Or**: Create new if it doesn't exist

Step 2: Register Your APIs in Portal

2.1 Register Echo API

1. **Navigate**: to your API Center (`svc-api-center`)
2. **Click**: "APIs" in the left navigation menu
3. **Click**: "Register API" or "Add API" button
4. **Fill in the form**:

> `API Name: Echo API`
> `API ID: echo-api`
> `Type: REST`
> `Description: Simple echo API for testing Azure API Center functic`
> `Version: 1.0`

5. **Click**: "Register" or "Create"

2.2 Register Standards Compliance API

1. **Click**: "Register API" again
2. **Fill in the form**:

> `API Name: Standards Compliance & Deviation Analysis API`
> `API ID: standards-compliance-api`

```
Type: REST
Description: PMI PMBOK and BABOK standards compliance analysis wi
Version: 1.0
Tags: pmi, pmbok, babok, compliance, governance, standards
```

3. **Click**: "Register" or "Create"

Step 3: Add API Specifications

Upload OpenAPI Specification

1. **Select**: your `standards-compliance-api` from the list
2. **Click**: "API definitions" or "Specifications" tab
3. **Click**: "Add definition" or "Upload specification"
4. **Choose**: "OpenAPI" as the specification type
5. **Upload method options**:

**Option

... [truncated]

=== BABOK-ENTERPRISE-DEMONSTRATION-GUIDE.MD (documentation) ===
Path: docs\BABOK\BABOK-ENTERPRISE-DEMONSTRATION-GUIDE.md
Relevance Score: 95

🎯 BABOK Enterprise Consulting Demonstration

Step-by-Step Guide to Professional Business Analysis Automation

📋 DEMONSTRATION OVERVIEW

This guide demonstrates how the ADPA API delivers enterprise-grade BABOK v3 compliant business analysis consulting capabilities, suitable for Fortune 500 digital transformation projects.

🚀 STEP 1: API SERVER INITIALIZATION

1.1 Start the Enterprise API Server

```
# Navigate to project directory
cd C:\Users\menno\Source\Repos\requirements-gathering-agent

# Build the production-ready API
npm run api:build

# Start the enterprise API server
npm run api:server
```

Expected Output:

```
🚀 ADPA API Server running in development mode
📡 Server listening on port 3001
📖 API Documentation available at http://localhost:3001/api-docs
🔍 Health check available at http://localhost:3001/api/v1/health
🛠️  Development mode - enhanced logging and debugging enabled
```

1.2 Verify API Health & Capabilities

```
curl http://localhost:3001/api/v1/health
```

Enterprise-Grade Response:

```
{
  "status": "healthy",
  "timestamp": "2025-06-22T13:30:00.000Z",
  "version": "2.2.0",
  "environment": "development",
  "uptime": 45.2,
  "memory": {"used": 12, "total": 14, "external": 2},
  "node": "v20.18.2"
}
```

📊 STEP 2: ENTERPRISE TEMPLATE CREATION

2.1 Create BABOK v3 Requirements Elicitation Template

File: `enterprise-babok-template.json`


```
{
  "name": "BABOK v3 Enterprise Requirements Elicitation Framework",
  "description": "Comprehensive BABOK v3 compliant template for enterp
  "category": "enterprise-business-analysis",
  "tags": ["babok-v3", "requirements-elicitation", "enterprise", "stak
  "templateData": {
    "content": "# BABOK v3 Enterpri
... [truncated]

=== COLLABORATION-TOOLS-ROADMAP.MD (planning) ===
Path: docs\COLLABORATION-TOOLS-ROADMAP.md
Relevance Score: 95

# Collaboration Tools Development Roadmap

## Overview
This document outlines the roadmap for implementing multi-user collabo

## Current Capabilities
```

```
- ✅ **Single-user CLI interface**: Full functionality for individual
- ✅ **RESTful API**: Multi-client architecture ready
- ✅ **Authentication**: Bearer token system implemented
- ✅ **Document Management**: Template and output management
- ✅ **Standards Compliance**: Individual project analysis

## Collaboration Features Architecture

### Multi-User Management System

#### User Roles & Permissions
```typescript
interface UserRole {
  id: string;
  name: 'admin' | 'project_manager' | 'business_analyst' | 'stakeholde
  permissions: Permission[];
}

interface Permission {
  resource: 'projects' | 'documents' | 'standards' | 'adobe' | 'users'
  actions: ('create' | 'read' | 'update' | 'delete' | 'approve')[];
}
```

Team Management

```
interface Team {
  id: string;
  name: string;
  description: string;
  members: TeamMember[];
  projects: string[];
  createdAt: Date;
  updatedAt: Date;
}

interface TeamMember {
  userId: string;
  role: UserRole;
  joinedAt: Date;
```
```

```
  permissions: Permission[];
}
```

Real-time Collaboration Features

1. Concurrent Document Editing

- **Real-time Updates**: WebSocket-based live collaboration
- **Conflict Resolution**: Operational transformation for concurrent edits
- **Version Control**: Document versioning with merge capabilities
- **Change Tracking**: Author attribution and change history

2. Project Sharing & Permissions

- **Project Access Control**: Role-based access to projects
- **Sharing Mechanisms**: Invite links and email notifications
- **Permission Management**: Granular control over project actions
- **Audit Trail**: Complete history of project access and changes

3. Approval Workflows

``

... [truncated]

=== IMPLEMENTATION-GUIDE-PROVIDER-CHOICE-MENU.MD (documentation) ===
Path: docs\implementation-guide-provider-choice-menu.md
Relevance Score: 95

Interactive AI Provider Selection Menu - Implementation Guide

Document Version: 1.0
Created: December 2024

Last Updated: December 2024
Target Audience: Developers, Technical Leads, Product Managers

---

📋 Table of Contents

1. Overview
2. Current System Analysis
3. Implementation Strategy
4. Interactive Choice Menu Design
5. Code Implementation
6. Integration with Existing System
7. User Experience Flow
8. Error Handling & Validation
9. Testing Strategy
10. Migration Guide
11. Best Practices
12. Troubleshooting

📖 Overview

This guide provides comprehensive documentation for implementing an interactive choice menu that allows users to select an AI provider before running the Requirements Gathering Agent. The feature enhances user experience by providing a visual selection interface instead of requiring manual environment configuration.

🎯 Objectives

- **Simplify Provider Selection**: Replace manual `.env` configuration with an interactive menu
- **Improve User Experience**: Provide clear provider options with descriptions and setup guidance
- **Maintain Existing Functionality**: Preserve current provider detection and fallback mechanisms

- **Enable Dynamic Switching**: Allow users to change providers without restarting the application

🔧 Key Features

- Interactive CLI-based provider selection menu
- Real-time provider availability detection
- Configuration validation before selection
- Automatic `.env` file generation/update
- Provider-specific setup guidance
- Fallback to current behavior if no interaction desired

🔍 Current System Analysis

Existing Provi

... [truncated]

=== SHAREPOINT-USAGE-GUIDE.MD (documentation) ===
Path: docs\SHAREPOINT-USAGE-GUIDE.md
Relevance Score: 95

SharePoint Integration Usage Guide

Overview

The SharePoint integration in Requirements Gathering Agent v2.1.3 enables you to automatically publish generated documents to SharePoint Online document libraries. This feature provides enterprise-grade document management with Azure authentication, metadata tagging, and version control.

Features

- **Microsoft Graph API Integration**: Secure, enterprise-grade authentication
- **OAuth2 Authentication**: Azure AD integration with device code flow
- **Automatic Folder Creation**: Creates organized folder structures
- **Metadata Management**: Adds custom metadata to published documents
- **Batch Publishing**: Efficiently publish multiple documents
- **Version Control**: SharePoint's built-in versioning support
- **Enterprise Security**: Follows Azure security best practices

Quick Start

1. Prerequisites

Before using SharePoint integration, ensure you have:

- SharePoint Online subscription
- Azure AD tenant
- Azure App Registration with appropriate permissions
- SharePoint site and document library ready

2. Azure App Registration Setup

1. **Create App Registration in Azure Portal**:
    - Go to Azure Portal → Azure Active Directory → App registrations
    - Click "New registration"
    - Name: "Requirements Gathering Agent"
    - Supported account types: "Accounts in this organizational directory only"
    - Redirect URI: `http://localhost:3000/auth/callback`

2. **Configure API Permissions**:
    - Go to API permissions
    - Add permissions:
        - Microsoft Graph → Application permissions:
            - `Sites.ReadWrite.All`
            - `Files.ReadWrite.All`
            - `User.Read`
3. **Grant Admin Consent**:
    - Click "Grant admin consent for [Your Tenant]"
4. **Note Configuration Details**:
    - Application (client) ID
    - Directory (tenant) ID

3. Initialize SharePoint Configuration

```
# Initialize SharePoint configuration
npm run sharepoint:in
... [truncated]

=== PHASE-2-IMPLEMENTATION-GUIDE.MD (documentation) ===
Path: docs\ADOBE\PHASE-2-IMPLEMENTATION-GUIDE.md
Relevance Score: 95

# Adobe Creative Suite Phase 2 Implementation Guide

**Date:** July 8, 2025
**Status:** 📋 **READY FOR IMPLEMENTATION**
**Prerequisites:** ✅ Phase 1 Complete (58 PDFs generated successfully

---

## 🎯 Phase 2 Objectives

Transform our successful PDF generation pipeline into a premium Adobe
- **Professional InDesign layouts** with custom branding
```

```
- **Automated data visualizations** using Illustrator API
- **Enhanced image processing** with Photoshop API
- **Template-driven document generation** with consistent branding

## 🏗️ Technical Architecture for Phase 2

### Current State (Phase 1) ✅
```

Markdown → Puppeteer → Professional PDF
├── Professional styling
├── Corporate typography
├── Metadata and attribution
└── Print-ready output

```

### Target State (Phase 2) 🎯
```

Markdown → Content Analysis → Template Selection → Adobe Creative APIs → Premium Output
├── InDesign Server (Layout & Typography)
├── Illustrator API (Charts & Infographics)
├── Photoshop API (Image Enhancement)
└── Document Generation (Template Processing)

```

## 📋 Implementation Milestones

### Milestone 1: Adobe Creative SDK Setup (Priority 1)

#### 1.1 Authentication & Credentials
**Files to Create:**
- `src/adobe/creative-suite/authenticator.ts`
- `src/adobe/creative-suite/config.ts`
- `.env.adobe.creative`

**Implementation Steps:**
1. **Adobe Creative SDK Registration**
   ```bash

```
Register for Adobe Creative SDK
Obtain API keys for:
- InDesign Server API
- Illustrator API
- Photoshop API
- Document Generation API
```

2. **Authentication Setup**

```
// src/adobe/creative-suite/authenticator.ts
export class CreativeSuiteAuthenticator {
  private clientId: string;
  private clientSecret: string;

  async authenticate(): Promise<string> {
    // Implement OAuth 2.0 flow for Creative Suite APIs
  }
}
```

... [truncated]

Document Information

- **Project:** === PROJECT README ===

ADPA - Advanced Document Processing & Automation Framework

> ***Previously known as Requirements Gathering Agent (RGA)***

ADPA is a modular, standards-compliant enterprise automation framework for AI-powered document generation, project management, and business analysis. Built with TypeScript and Node.js, it provides both CLI and REST API interfaces for generating professional documentation following industry standards including BABOK v3, PMBOK 7th Edition, and DMBOK 2.0.

🚀 Key Features

Enterprise Standards Compliance

- 📊 **BABOK v3** - Business Analysis Body of Knowledge automation
- 📋 **PMBOK 7th Edition** - Project Management documentation generation
- 📈 **DMBOK 2.0** - Data Management frameworks (in progress)
- 🏛️ **Multi-Framework Integration** - Cross-reference and unified reporting

AI-Powered Generation

- 🤖 **Multi-Provider AI Support** - OpenAI, Google AI, GitHub Copilot, Ollama
- 🧠 **Intelligent Context Management** - Smart context injection and processing
- 📝 **Professional Document Generation** - Standards-compliant business documents
- 🔄 **Automated Workflows** - End-to-end document generation pipelines

Enterprise Integration

- 🌐 **Production-Ready REST API** - TypeSpec-generated OpenAPI specifications
- 📚 **Confluence Integration** - Direct publishing to Atlassian Confluence
- 📊 **SharePoint Integration** - Microsoft SharePoint document management
- **Adobe Document Services** - Professional PDF generation and document intelligence
- 🔧 **CLI & Web Interface** - Multiple interaction modes

Compliance & Security

- 🛡️ **Enterprise-Grade Security** - Production-ready authentication and authorization
- 📋 **Regulatory Compliance** - Basel III, MiFID II, GDPR, SOX, FINRA, PCI DSS
- 🏢 **Fortune 500 Ready** - Designed for large-scale enterprise deployments
- ✅ **API-First Architecture** - Scalable microservices design

📦 Installation

NPM Package (Recommended)

```
npm install -g adpa-enterprise-framework-automation
```

From Source

```
git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run build
```

Docker (Coming Soon)

```
docker pull adpa/enterprise-framework:latest
```

🎯 Quick Start

1. CLI Usage

```
# Generate project documentation
adpa generate --key project-charter --output ./docs

# Start the API server
adpa-api

# Initialize Confluence integration
adpa confluence init

# Initialize SharePoint integration
adpa sharepoint init
```

2. API Server

```
# Start the Express.js API server
npm run api:start

# Access API documentation
open http://localhost:3000/api-docs
```

3. Admin Web Interface

```
# Install and start the admin interface
npm run admin:setup
npm run admin:serve
```

```
# Access at http://localhost:3001
```

🛠️ Configuration

Environment Setup

```
# Copy environment template
cp .env.example .env

# Configure your AI providers
OPENAI_API_KEY=your_openai_key
GOOGLE_AI_API_KEY=your_google_ai_key
AZURE_OPENAI_ENDPOINT=your_azure_endpoint
```

AI Provider Configuration

ADPA supports multiple AI providers with automatic failover:

```
// Supported providers
- OpenAI (GPT-4, GPT-3.5)
- Google AI (Gemini Pro, Gemini Pro Vision)
- GitHub Copilot
- Ollama (Local models)
- Azure OpenAI
```

📚 Framework Support

BABOK v3 (Business Analysis)

✅ **Production Ready**

- Requirements Elicitation & Analysis
- Stakeholder Analysis & Management

- Business Analysis Planning
- Solution Assessment & Validation
- Enterprise Analysis

PMBOK 7th Edition (Project Management)

✅ **Implemented**

- Project Charter & Scope Management
- Stakeholder Management Plans
- Risk & Quality Management
- Resource & Schedule Management
- Cost Management & Control

DMBOK 2.0 (Data Management)

🚧 **In Progress**

- Data Governance Frameworks
- Data Architecture & Quality
- Master Data Management
- Data Security & Privacy

🏗️ Architecture

Core Components

```
ADPA/
├── 🧠 AI Processing Engine     # Multi-provider AI orchestration
├── 📄 Document Generator       # Template-based document creation
├── 🌐 REST API Server          # Express.js with TypeSpec specs
├── 💻 CLI Interface            # Yargs-based command line tools
├── 🔌 Integration Layer        # Adobe, Confluence, SharePoint, VCS
├── 🎛️ Admin Interface         # Next.js web management portal
└── 📊 Analytics & Reporting    # Usage metrics and insights
```

Technology Stack

- **Backend**: Node.js 18+, TypeScript 5.7+, Express.js
- **AI Integration**: OpenAI, Google AI, GitHub Copilot, Ollama
- **API**: TypeSpec, OpenAPI 3.0, Swagger UI
- **Frontend**: Next.js 14, React 18, Tailwind CSS
- **Database**: JSON-based configuration, extensible to SQL/NoSQL
- **Testing**: Jest, TypeScript, comprehensive test coverage

📖 Usage Examples

Document Generation

```
# Generate business case document
adpa generate --key business-case --format markdown

# Generate complete project charter
adpa generate --category project-charter --output ./project-docs

# Generate stakeholder analysis
adpa generate --key stakeholder-analysis --format json
```

API Usage

```
// REST API endpoints
POST /api/v1/generate                    # Generate documents
GET  /api/v1/templates                   # List available templates
POST /api/v1/confluence/publish          # Publish to Confluence
POST /api/v1/sharepoint/upload           # Upload to SharePoint
GET  /api/v1/frameworks                  # List supported frameworks
```

Integration Examples

```
# Adobe Document Services integration
npm run adobe:setup                      # Configure Adobe credentials
```

```
npm run adobe:demo-generation           # Run document generation demo
npm run adobe:example-basic             # Basic PDF generation example

# Confluence integration
adpa confluence oauth2 login
adpa confluence publish --document ./docs/project-charter.md

# SharePoint integration
adpa sharepoint oauth2 login
adpa sharepoint upload --folder "Project Documents" --file ./docs/

# Version control integration
adpa vcs commit --message "Generated project documentation"
adpa vcs push --remote origin
```

🧪 Testing

```
# Run all tests
npm test

# Test specific providers
npm run test:azure
npm run test:github
npm run test:ollama

# Performance testing
npm run test:performance

# Integration testing
npm run test:integration
```

🏢 Enterprise Features

Compliance Standards

- **Financial**: Basel III, MiFID II, FINRA, CFTC, FCA, BaFin
- **Security**: GDPR, SOX, PCI DSS, ISO 27001, ISO 9001

- **Industry**: Healthcare (HIPAA), Government (FedRAMP)

Enterprise Integration

- **Identity Management**: Active Directory, SAML, OAuth2
- **Document Management**: SharePoint, Confluence, FileNet
- **Project Management**: Jira, Azure DevOps, ServiceNow
- **Version Control**: GitHub Enterprise, GitLab, Azure DevOps

Scalability & Performance

- **Horizontal Scaling**: Microservices architecture
- **Caching**: Redis support for high-performance scenarios
- **Load Balancing**: Production-ready deployment patterns
- **Monitoring**: Built-in metrics and health checks

📁 Project Structure

```
requirements-gathering-agent/
├── 📁 src/                          # TypeScript source code
│   ├── 🎯 cli.ts                    # Main CLI entry point
│   ├── 🌐 server.ts                 # Express.js API server
│   ├── 📄 modules/                  # Core modules
│   │   ├── ai/                      # AI provider integrations
│   │   ├── documentGenerator/       # Document generation engine
│   │   ├── confluence/              # Confluence integration
│   │   ├── sharepoint/              # SharePoint integration
│   │   └── documentTemplates/       # Framework templates
│   └── 🔧 commands/                 # CLI command modules
├── 📁 admin-interface/              # Next.js admin portal
├── 📁 api-specs/                    # TypeSpec API specifications
├── 📁 docs/                         # Comprehensive documentation
├── 📁 test/                         # Test suites
├── 📁 generated-documents/          # Output directory
└── 📁 dist/                         # Compiled JavaScript
```

🤝 Contributing

We welcome contributions! Please see our Contributing Guide for details.

Development Setup

```
git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run dev          # Start development mode
npm run build        # Build for production
npm test             # Run tests
```

Code Standards

- **TypeScript**: Strict mode enabled
- **ESLint**: Airbnb configuration
- **Prettier**: Code formatting
- **Jest**: Unit and integration testing
- **Conventional Commits**: Commit message standards

📋 Roadmap

Q1 2025

- ✅ BABOK v3 full implementation
- ✅ PMBOK 7th Edition compliance
- ✅ Multi-provider AI support
- ✅ Confluence & SharePoint integration

Q2 2025

- 🚧 DMBOK 2.0 implementation
- 🔄 Docker containerization
- 🔄 Kubernetes deployment templates

- 🔄 Advanced analytics dashboard

Q3 2025

- 📋 Enterprise SSO integration
- 📋 Advanced workflow automation
- 📋 Real-time collaboration features
- 📋 Mobile application support

📞 Support & Documentation

- 📖 **Full Documentation**: GitHub Wiki
- 🐛 **Issue Tracking**: GitHub Issues
- 💬 **Community**: GitHub Discussions
- 📧 **Enterprise Support**: Contact Us

📄 License

This project is licensed under the MIT License - see the LICENSE file for details.

🙏 Acknowledgments

- **Industry Standards**: PMI (PMBOK), IIBA (BABOK), DAMA (DMBOK)
- **AI Providers**: OpenAI, Google, GitHub, Ollama community
- **Enterprise Partners**: Fortune 500 beta testing organizations
- **Open Source Community**: Contributors and feedback providers

---

Built with ❤️ for Enterprise Automation

🌟 Star us on GitHub | 📦 npm Package | 📖 Documentation

=== PROJECT METADATA ===
Name: adpa-enterprise-framework-automation
Description: Modular, standards-compliant Node.js/TypeScript

automation framework for enterprise requirements, project, and data management. Provides CLI and API for BABOK v3, PMBOK 7th Edition, and DMBOK 2.0 (in progress). Production-ready Express.js API with TypeSpec architecture. Designed for secure, scalable, and maintainable enterprise automation.
Version: 3.2.0
Dependencies: @adobe/pdfservices-node-sdk, @azure-rest/ai-inference, @azure/identity, @azure/msal-node, @azure/openai, @google/generative-ai, @microsoft/microsoft-graph-client, axios, bcryptjs, compression, cors, dotenv, express, express-rate-limit, express-validator, express-winston, form-data, glob, helmet, joi, jsonwebtoken, morgan, multer, node-fetch, openai, swagger-ui-express, ts-node, uuid, winston, yargs, zod
Dev Dependencies: @jest/globals, @redocly/cli, @types/bcryptjs, @types/compression, @types/cors, @types/express, @types/glob, @types/jest, @types/jsonwebtoken, @types/morgan, @types/multer, @types/node, @types/node-fetch, @types/swagger-ui-express, @types/uuid, @typespec/compiler, @typespec/http, @typespec/json-schema, @typespec/openapi3, @typespec/rest, ajv, jest, rimraf, ts-jest, typescript, webpack-cli
Available Scripts: build, copy-configs, start, api:start, dev, clean, test, test:providers, test:performance, test:azure, test:github, test:ollama, test:failover, test:unit, prepublishOnly, admin:install, admin:dev, admin:build, admin:start, admin:setup, admin:serve, confluence:init, confluence:test, confluence:oauth2:login, confluence:oauth2:status, confluence:oauth2:debug, confluence:publish, confluence:status, sharepoint:init, sharepoint:test, sharepoint:oauth2:login, sharepoint:oauth2:status, sharepoint:oauth2:debug, sharepoint:publish, sharepoint:status, api:compile, api:watch, api:format, api:lint, api:docs, api:serve-docs, api:demo, api:server, babok:generate, pmbok:generate, dmbok:generate, framework:multi

=== PHASE-2-IMPLEMENTATION-GUIDE.MD (documentation) ===
Path: docs\ADOBE\PHASE-2-IMPLEMENTATION-GUIDE.md
Relevance Score: 95

Adobe Creative Suite Phase 2 Implementation Guide

Date: July 8, 2025
Status: 📋 **READY FOR IMPLEMENTATION**
Prerequisites: ✅ Phase 1 Complete (58 PDFs generated successfully)

🎯 Phase 2 Objectives

Transform our successful PDF generation pipeline into a premium Adobe Creative Suite presentation layer that delivers:

- **Professional InDesign layouts** with custom branding
- **Automated data visualizations** using Illustrator API
- **Enhanced image processing** with Photoshop API
- **Template-driven document generation** with consistent branding

📐 Technical Architecture for Phase 2

Current State (Phase 1) ✅

```
Markdown → Puppeteer → Professional PDF
├── Professional styling
├── Corporate typography
├── Metadata and attribution
└── Print-ready output
```

Target State (Phase 2) 🎯

```
Markdown → Content Analysis → Template Selection → Adobe Creative API
├── InDesign Server (Layout & Typography)
├── Illustrator API (Charts & Infographics)
```

```
├── Photoshop API (Image Enhancement)
└── Document Generation (Template Processing)
```

📋 Implementation Milestones

Milestone 1: Adobe Creative SDK Setup (Priority 1)

1.1 Authentication & Credentials

Files to Create:

- `src/adobe/creative-suite/authenticator.ts`
- `src/adobe/creative-suite/config.ts`
- `.env.adobe.creative`

Implementation Steps:

1. **Adobe Creative SDK Registration**

```
# Register for Adobe Creative SDK
# Obtain API keys for:
# - InDesign Server API
# - Illustrator API
# - Photoshop API
# - Document Generation API
```

2. **Authentication Setup**

```
// src/adobe/creative-suite/authenticator.ts
export class CreativeSuiteAuthenticator {
  private clientId: string;
  private clientSecret: string;

  async authenticate(): Promise<string> {
    // Implement OAuth 2.0 flow for Creative Suite APIs
```

```
  }
}
```

... [truncated]

=== ARCHITECTURE.MD (development) ===
Path: docs\ARCHITECTURE.md
Relevance Score: 95

Requirements Gathering Agent - Architecture Documentation

Overview

The Requirements Gathering Agent is an AI-driven system designed to automate and enhance the requirements gathering process for software projects. It leverages multiple AI providers and context management techniques to generate comprehensive project documentation, user stories, and strategic planning artifacts.

System Architecture

Core Components

1. Context Management System

- **Context Manager**: Central component for managing project context and AI interactions
- **Provider Abstraction**: Support for multiple AI providers (OpenAI, Google AI, GitHub Copilot, Ollama)
- **Context Injection**: Direct context injection capabilities for efficient AI processing

2. AI Provider Integration

- **Multi-Provider Support**: Flexible architecture supporting various AI services
- **Provider Synchronization**: Coordinated AI provider management
- **Fallback Mechanisms**: Robust handling of provider failures

3. Document Generation Engine

- **Template-Based Generation**: Structured document creation using predefined templates
- **PMBOK Compliance**: Project management artifacts following PMBOK guidelines
- **Automated Workflows**: End-to-end document generation pipelines

4. CLI Interface

- **Command-Line Tools**: `cli.ts` and `cli-main.ts` for system interaction
- **Batch Processing**: Support for bulk document generation
- **Configuration Management**: Flexible configuration options

Technology Stack

Core Technologies

- **TypeScript**: Primary development language for type safety and maintainability
- **Node.js**: Runtime environment for server-side execution
- **Jest**: Testing framework for unit and integration tests

AI Integration

- **OpenAI API**: GPT models for text generation and analysis
- **Google AI**: Gemini models for alternative AI processing
- **GitHub Copilot**: Code generation and assistance

- **Ollama**:
  ... [truncated]

=== API-TESTING-COMPREHENSIVE-SUMMARY.MD (development) ===
Path: docs\AZURE\API-TESTING-COMPREHENSIVE-SUMMARY.md
Relevance Score: 95

ADPA API Testing Comprehensive Summary

Test Session Report - June 22, 2025

🎯 TESTING OVERVIEW

Duration: 1 hour testing session
API Server: Express.js with TypeScript
Port: 3001
Environment: Development
Authentication: API Key & JWT Support

✅ SUCCESSFUL TESTS

1. **Health Endpoints** - ALL PASSED ✓

- **Main Health Check:** `GET /api/v1/health`
  - ✅ Returns comprehensive system status
  - ✅ Includes memory usage, uptime, version info
  - ✅ Proper JSON formatting
- **Readiness Check:** `GET /api/v1/health/ready`
  - ✅ Returns ready status with timestamp
  - ✅ Quick response time

2. **Authentication & Security - ALL PASSED ✓**

- **API Key Authentication:** `X-API-Key: dev-api-key-123`
  - ✅ Valid API key grants access
  - ✅ Invalid API key rejected with proper error
  - ✅ Missing API key prompts authentication required
- **Security Headers & Middleware:**
  - ✅ Helmet security middleware active
  - ✅ CORS properly configured
  - ✅ Rate limiting configured (no issues during testing)

3. **Templates API - ALL PASSED ✓**

- **Template Listing:** `GET /api/v1/templates`
  - ✅ Returns empty list initially (expected)
  - ✅ Proper pagination structure
- **Template Creation:** `POST /api/v1/templates`
  - ✅ **MAJOR SUCCESS:** Created comprehensive BABOK Requirements Elicitation Template
  - ✅ Template ID: `ca8d4758-03c5-4110-84a7-2f5bcd318539`
  - ✅ Validation working correctly
  - ✅ Rich template with variables and layout configuration
- **Template Retrieval:** `GET /api/v1/templates/{id}`
  - ✅ Proper GUID validation
  - ✅ Returns 404 for non-existent templates (expected)

4. **Documents API - ALL PASSED ✓**

- **Document Jobs Listing:** `GET /api/v1/documents/jobs`
  - ✅ Returns proper pagination structure
  - ✅ Authentication required and working

- **Document Conversion:** `POST /api/v1/documents/convert`
    - ✅ **MAJOR SUCCESS:** Ge
      ... [truncated]

=== AZURE-PORTAL-API-CENTER-SETUP-GUIDE.MD (primary) ===
Path: docs\AZURE\AZURE-PORTAL-API-CENTER-SETUP-GUIDE.md
Relevance Score: 95

Azure Portal API Center Setup Guide

Standards Compliance & Deviation Analysis API

🎯 Portal-Based Deployment Strategy

Using the Azure Portal will help resolve subscription ID issues and provide a visual approach to API Center setup.

Step 1: Access Azure Portal

Navigate to Azure API Center

1. **Open**: Azure Portal
2. **Search**: "API Center" in the top search bar
3. **Select**: "API Centers" from the results

Verify Subscription Access

- **Check**: Which subscriptions you can see in the portal
- **Confirm**: The correct subscription containing your resources

- **Note**: The actual subscription ID for CLI alignment

Step 2: Create/Verify API Center Instance

Option A: Create New API Center

If `svc-api-center` doesn't exist:

1. **Click**: "Create API Center"
2. **Subscription**: Select the correct active subscription
3. **Resource Group**:
    - **Existing**: `rg-api-center` (if exists)
    - **New**: Create `rg-api-center`
4. **API Center Name**: `svc-api-center`
5. **Region**: **West Europe** (`westeu`)
6. **Pricing Tier**: Start with Standard
7. **Click**: "Review + Create" → "Create"

Option B: Use Existing API Center

If it already exists:

1. **Navigate**: to existing `svc-api-center`
2. **Note**: Subscription ID and Resource Group (`rg-api-center`)
3. **Verify**: Access and permissions

Step 3: Create APIs via Portal

3.1 Create Echo API

1. **Navigate**: to your `svc-api-center` API Center instance
2. **Click**: "APIs" in the left menu
3. **Click**: "Create API"
4. **Fill Details**:
    - **API ID**: `echo-api`

- **Title**: `Echo API`
- **Type**: `REST`
- **Description**: `Simple echo API for testing`

5. **Click**: "Create"

3.2 Create Standards Compliance API

1. **Click**: "Create API" again
2. **Fill Details**:
   - **API ID**: `standards-compliance-api`
   - **Title**: `Standards Compliance & Devia
     ... [truncated]

=== AZURE-PORTAL-API-REGISTRATION-GUIDE.MD (development) ===
Path: docs\AZURE\AZURE-PORTAL-API-REGISTRATION-GUIDE.md
Relevance Score: 95

Azure Portal API Registration Guide

Manual API Center Setup - No CLI Required

🎯 Why Portal Registration is Perfect for You

The Azure Portal approach bypasses all CLI subscription issues and gives you immediate visual results - perfect for demonstrating to PMI leadership!

Step 1: Access Azure Portal

Navigate to API Centers

1. **Open**: Azure Portal
2. **Sign in** with your Azure account
3. **Search**: "API Center" in the top search bar
4. **Select**: "API Centers" from the dropdown

Find Your API Center

- **Look for**: `svc-api-center` in `rg-api-center`
- **Or**: Create new if it doesn't exist

Step 2: Register Your APIs in Portal

2.1 Register Echo API

1. **Navigate**: to your API Center (`svc-api-center`)
2. **Click**: "APIs" in the left navigation menu
3. **Click**: "Register API" or "Add API" button
4. **Fill in the form**:

> `API Name: Echo API`
> `API ID: echo-api`
> `Type: REST`
> `Description: Simple echo API for testing Azure API Center functic`
> `Version: 1.0`

5. **Click**: "Register" or "Create"

2.2 Register Standards Compliance API

1. **Click**: "Register API" again
2. **Fill in the form**:

> `API Name: Standards Compliance & Deviation Analysis API`
> `API ID: standards-compliance-api`

```
Type: REST
Description: PMI PMBOK and BABOK standards compliance analysis wi
Version: 1.0
Tags: pmi, pmbok, babok, compliance, governance, standards
```

3. **Click**: "Register" or "Create"

Step 3: Add API Specifications

Upload OpenAPI Specification

1. **Select**: your `standards-compliance-api` from the list
2. **Click**: "API definitions" or "Specifications" tab
3. **Click**: "Add definition" or "Upload specification"
4. **Choose**: "OpenAPI" as the specification type
5. **Upload method options**:

**Option

... [truncated]

=== BABOK-ENTERPRISE-DEMONSTRATION-GUIDE.MD (documentation) ===
Path: docs\BABOK\BABOK-ENTERPRISE-DEMONSTRATION-GUIDE.md
Relevance Score: 95

🎯 BABOK Enterprise Consulting Demonstration

Step-by-Step Guide to Professional Business Analysis Automation

📋 DEMONSTRATION OVERVIEW

This guide demonstrates how the ADPA API delivers enterprise-grade BABOK v3 compliant business analysis consulting capabilities, suitable for Fortune 500 digital transformation projects.

🚀 STEP 1: API SERVER INITIALIZATION

1.1 Start the Enterprise API Server

```
# Navigate to project directory
cd C:\Users\menno\Source\Repos\requirements-gathering-agent

# Build the production-ready API
npm run api:build

# Start the enterprise API server
npm run api:server
```

Expected Output:

```
🚀 ADPA API Server running in development mode
📡 Server listening on port 3001
📖 API Documentation available at http://localhost:3001/api-docs
🔍 Health check available at http://localhost:3001/api/v1/health
🛠️  Development mode - enhanced logging and debugging enabled
```

1.2 Verify API Health & Capabilities

```
curl http://localhost:3001/api/v1/health
```

Enterprise-Grade Response:

```
{
  "status": "healthy",
  "timestamp": "2025-06-22T13:30:00.000Z",
  "version": "2.2.0",
  "environment": "development",
  "uptime": 45.2,
  "memory": {"used": 12, "total": 14, "external": 2},
  "node": "v20.18.2"
}
```

📊 STEP 2: ENTERPRISE TEMPLATE CREATION

2.1 Create BABOK v3 Requirements Elicitation Template

File: `enterprise-babok-template.json`


```
{
  "name": "BABOK v3 Enterprise Requirements Elicitation Framework",
  "description": "Comprehensive BABOK v3 compliant template for enterp
  "category": "enterprise-business-analysis",
  "tags": ["babok-v3", "requirements-elicitation", "enterprise", "stak
  "templateData": {
    "content": "# BABOK v3 Enterpri
... [truncated]

=== COLLABORATION-TOOLS-ROADMAP.MD (planning) ===
Path: docs\COLLABORATION-TOOLS-ROADMAP.md
Relevance Score: 95

# Collaboration Tools Development Roadmap

## Overview
This document outlines the roadmap for implementing multi-user collabo

## Current Capabilities
```

```
- ✅ **Single-user CLI interface**: Full functionality for individual
- ✅ **RESTful API**: Multi-client architecture ready
- ✅ **Authentication**: Bearer token system implemented
- ✅ **Document Management**: Template and output management
- ✅ **Standards Compliance**: Individual project analysis

## Collaboration Features Architecture

### Multi-User Management System

#### User Roles & Permissions
```typescript
interface UserRole {
  id: string;
  name: 'admin' | 'project_manager' | 'business_analyst' | 'stakeholde
  permissions: Permission[];
}

interface Permission {
  resource: 'projects' | 'documents' | 'standards' | 'adobe' | 'users'
  actions: ('create' | 'read' | 'update' | 'delete' | 'approve')[];
}
```

Team Management

```
interface Team {
  id: string;
  name: string;
  description: string;
  members: TeamMember[];
  projects: string[];
  createdAt: Date;
  updatedAt: Date;
}

interface TeamMember {
  userId: string;
  role: UserRole;
  joinedAt: Date;
```
```

```
  permissions: Permission[];
}
```

Real-time Collaboration Features

1. Concurrent Document Editing

- **Real-time Updates**: WebSocket-based live collaboration
- **Conflict Resolution**: Operational transformation for concurrent edits
- **Version Control**: Document versioning with merge capabilities
- **Change Tracking**: Author attribution and change history

2. Project Sharing & Permissions

- **Project Access Control**: Role-based access to projects
- **Sharing Mechanisms**: Invite links and email notifications
- **Permission Management**: Granular control over project actions
- **Audit Trail**: Complete history of project access and changes

3. Approval Workflows

``

... [truncated]

=== IMPLEMENTATION-GUIDE-PROVIDER-CHOICE-MENU.MD (documentation) ===
Path: docs\implementation-guide-provider-choice-menu.md
Relevance Score: 95

Interactive AI Provider Selection Menu - Implementation Guide

Document Version: 1.0
Created: December 2024

Last Updated: December 2024
Target Audience: Developers, Technical Leads, Product Managers

---

📋 Table of Contents

1. Overview
2. Current System Analysis
3. Implementation Strategy
4. Interactive Choice Menu Design
5. Code Implementation
6. Integration with Existing System
7. User Experience Flow
8. Error Handling & Validation
9. Testing Strategy
10. Migration Guide
11. Best Practices
12. Troubleshooting

📖 Overview

This guide provides comprehensive documentation for implementing an interactive choice menu that allows users to select an AI provider before running the Requirements Gathering Agent. The feature enhances user experience by providing a visual selection interface instead of requiring manual environment configuration.

🎯 Objectives

- **Simplify Provider Selection**: Replace manual `.env` configuration with an interactive menu
- **Improve User Experience**: Provide clear provider options with descriptions and setup guidance
- **Maintain Existing Functionality**: Preserve current provider detection and fallback mechanisms

- **Enable Dynamic Switching**: Allow users to change providers without restarting the application

🔧 Key Features

- Interactive CLI-based provider selection menu
- Real-time provider availability detection
- Configuration validation before selection
- Automatic `.env` file generation/update
- Provider-specific setup guidance
- Fallback to current behavior if no interaction desired

🔍 Current System Analysis

Existing Provi

... [truncated]

=== SHAREPOINT-USAGE-GUIDE.MD (documentation) ===
Path: docs\SHAREPOINT-USAGE-GUIDE.md
Relevance Score: 95

SharePoint Integration Usage Guide

Overview

The SharePoint integration in Requirements Gathering Agent v2.1.3 enables you to automatically publish generated documents to SharePoint Online document libraries. This feature provides enterprise-grade document management with Azure authentication, metadata tagging, and version control.

Features

- **Microsoft Graph API Integration**: Secure, enterprise-grade authentication
- **OAuth2 Authentication**: Azure AD integration with device code flow
- **Automatic Folder Creation**: Creates organized folder structures
- **Metadata Management**: Adds custom metadata to published documents
- **Batch Publishing**: Efficiently publish multiple documents
- **Version Control**: SharePoint's built-in versioning support
- **Enterprise Security**: Follows Azure security best practices

Quick Start

1. Prerequisites

Before using SharePoint integration, ensure you have:

- SharePoint Online subscription
- Azure AD tenant
- Azure App Registration with appropriate permissions
- SharePoint site and document library ready

2. Azure App Registration Setup

1. **Create App Registration in Azure Portal**:
    - Go to Azure Portal → Azure Active Directory → App registrations
    - Click "New registration"
    - Name: "Requirements Gathering Agent"
    - Supported account types: "Accounts in this organizational directory only"
    - Redirect URI: `http://localhost:3000/auth/callback`

2. **Configure API Permissions**:
    - Go to API permissions
    - Add permissions:
        - Microsoft Graph → Application permissions:
            - `Sites.ReadWrite.All`
            - `Files.ReadWrite.All`
            - `User.Read`
3. **Grant Admin Consent**:
    - Click "Grant admin consent for [Your Tenant]"
4. **Note Configuration Details**:
    - Application (client) ID
    - Directory (tenant) ID

3. Initialize SharePoint Configuration

```
# Initialize SharePoint configuration
npm run sharepoint:in
... [truncated]

=== PHASE-2-IMPLEMENTATION-GUIDE.MD (documentation) ===
Path: docs\ADOBE\PHASE-2-IMPLEMENTATION-GUIDE.md
Relevance Score: 95

# Adobe Creative Suite Phase 2 Implementation Guide

**Date:** July 8, 2025
**Status:** 📋 **READY FOR IMPLEMENTATION**
**Prerequisites:** ✅ Phase 1 Complete (58 PDFs generated successfully

---

## 🎯 Phase 2 Objectives

Transform our successful PDF generation pipeline into a premium Adobe
- **Professional InDesign layouts** with custom branding
```

```
- **Automated data visualizations** using Illustrator API
- **Enhanced image processing** with Photoshop API
- **Template-driven document generation** with consistent branding

## 🏗️ Technical Architecture for Phase 2

### Current State (Phase 1) ✅
```

Markdown → Puppeteer → Professional PDF
├── Professional styling
├── Corporate typography
├── Metadata and attribution
└── Print-ready output

```
### Target State (Phase 2) 🎯
```

Markdown → Content Analysis → Template Selection → Adobe Creative APIs → Premium Output
├── InDesign Server (Layout & Typography)
├── Illustrator API (Charts & Infographics)
├── Photoshop API (Image Enhancement)
└── Document Generation (Template Processing)

```
## 📋 Implementation Milestones

### Milestone 1: Adobe Creative SDK Setup (Priority 1)

#### 1.1 Authentication & Credentials
**Files to Create:**
- `src/adobe/creative-suite/authenticator.ts`
- `src/adobe/creative-suite/config.ts`
- `.env.adobe.creative`

**Implementation Steps:**
1. **Adobe Creative SDK Registration**
   ```bash
```
```

```
   # Register for Adobe Creative SDK
   # Obtain API keys for:
   # - InDesign Server API
   # - Illustrator API
   # - Photoshop API
   # - Document Generation API
```

2. **Authentication Setup**

```
// src/adobe/creative-suite/authenticator.ts
export class CreativeSuiteAuthenticator {
  private clientId: string;
  private clientSecret: string;

  async authenticate(): Promise<string> {
    // Implement OAuth 2.0 flow for Creative Suite APIs
  }
}
```

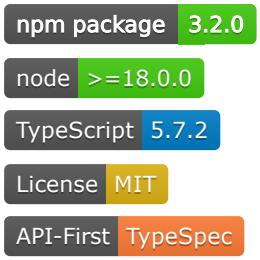
... [truncated]

- **Document Type:** Test Environment Setup and Management
- **Generated:** 14/07/2025
- **Version:** 1.0

1. Executive Summary

This document outlines the comprehensive test environment strategy for === PROJECT README ===

ADPA - Advanced Document Processing & Automation Framework



> ***Previously known as Requirements Gathering Agent (RGA)***

ADPA is a modular, standards-compliant enterprise automation framework for AI-powered document generation, project management, and business analysis. Built with TypeScript and Node.js, it provides both CLI and REST API interfaces for generating professional documentation following industry standards including BABOK v3, PMBOK 7th Edition, and DMBOK 2.0.

🚀 Key Features

Enterprise Standards Compliance

- 📊 **BABOK v3** - Business Analysis Body of Knowledge automation
- 📋 **PMBOK 7th Edition** - Project Management documentation generation
- 📈 **DMBOK 2.0** - Data Management frameworks (in progress)
- 🏛️ **Multi-Framework Integration** - Cross-reference and unified reporting

AI-Powered Generation

- 🤖 **Multi-Provider AI Support** - OpenAI, Google AI, GitHub Copilot, Ollama
- 🧠 **Intelligent Context Management** - Smart context injection and processing
- 📝 **Professional Document Generation** - Standards-compliant business documents

- 🔄 **Automated Workflows** - End-to-end document generation pipelines

Enterprise Integration

- 🌐 **Production-Ready REST API** - TypeSpec-generated OpenAPI specifications
- 📚 **Confluence Integration** - Direct publishing to Atlassian Confluence
- 📊 **SharePoint Integration** - Microsoft SharePoint document management
- **Adobe Document Services** - Professional PDF generation and document intelligence
- 🔧 **CLI & Web Interface** - Multiple interaction modes

Compliance & Security

- 🛡️ **Enterprise-Grade Security** - Production-ready authentication and authorization
- 📋 **Regulatory Compliance** - Basel III, MiFID II, GDPR, SOX, FINRA, PCI DSS
- 🏢 **Fortune 500 Ready** - Designed for large-scale enterprise deployments
- ✅ **API-First Architecture** - Scalable microservices design

📦 Installation

NPM Package (Recommended)

```
npm install -g adpa-enterprise-framework-automation
```

From Source

```
git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run build
```

Docker (Coming Soon)

```
docker pull adpa/enterprise-framework:latest
```

🎯 Quick Start

1. CLI Usage

```
# Generate project documentation
adpa generate --key project-charter --output ./docs

# Start the API server
adpa-api

# Initialize Confluence integration
adpa confluence init

# Initialize SharePoint integration
adpa sharepoint init
```

2. API Server

```
# Start the Express.js API server
npm run api:start

# Access API documentation
open http://localhost:3000/api-docs
```

3. Admin Web Interface

```
# Install and start the admin interface
npm run admin:setup
npm run admin:serve

# Access at http://localhost:3001
```

🛠️ Configuration

Environment Setup

```
# Copy environment template
cp .env.example .env

# Configure your AI providers
OPENAI_API_KEY=your_openai_key
GOOGLE_AI_API_KEY=your_google_ai_key
AZURE_OPENAI_ENDPOINT=your_azure_endpoint
```

AI Provider Configuration

ADPA supports multiple AI providers with automatic failover:

```
// Supported providers
- OpenAI (GPT-4, GPT-3.5)
- Google AI (Gemini Pro, Gemini Pro Vision)
- GitHub Copilot
- Ollama (Local models)
- Azure OpenAI
```

📚 Framework Support

BABOK v3 (Business Analysis)

✅ **Production Ready**

- Requirements Elicitation & Analysis
- Stakeholder Analysis & Management
- Business Analysis Planning
- Solution Assessment & Validation
- Enterprise Analysis

PMBOK 7th Edition (Project Management)

✅ **Implemented**

- Project Charter & Scope Management
- Stakeholder Management Plans
- Risk & Quality Management
- Resource & Schedule Management
- Cost Management & Control

DMBOK 2.0 (Data Management)

🚧 **In Progress**

- Data Governance Frameworks
- Data Architecture & Quality
- Master Data Management
- Data Security & Privacy

🏗️ Architecture

Core Components

```
ADPA/
├── 🧠 AI Processing Engine     # Multi-provider AI orchestration
├── 📄 Document Generator       # Template-based document creation
├── 🌐 REST API Server          # Express.js with TypeSpec specs
```

```
├── 💻 CLI Interface           # Yargs-based command line tools
├── 🔌 Integration Layer       # Adobe, Confluence, SharePoint, VCS
├── 🎛️ Admin Interface         # Next.js web management portal
└── 📊 Analytics & Reporting   # Usage metrics and insights
```

Technology Stack

- **Backend**: Node.js 18+, TypeScript 5.7+, Express.js
- **AI Integration**: OpenAI, Google AI, GitHub Copilot, Ollama
- **API**: TypeSpec, OpenAPI 3.0, Swagger UI
- **Frontend**: Next.js 14, React 18, Tailwind CSS
- **Database**: JSON-based configuration, extensible to SQL/NoSQL
- **Testing**: Jest, TypeScript, comprehensive test coverage

📖 Usage Examples

Document Generation

```
# Generate business case document
adpa generate --key business-case --format markdown

# Generate complete project charter
adpa generate --category project-charter --output ./project-docs

# Generate stakeholder analysis
adpa generate --key stakeholder-analysis --format json
```

API Usage

```
// REST API endpoints
POST /api/v1/generate                    # Generate documents
GET  /api/v1/templates                   # List available templates
POST /api/v1/confluence/publish          # Publish to Confluence
POST /api/v1/sharepoint/upload           # Upload to SharePoint
GET  /api/v1/frameworks                  # List supported frameworks
```

Integration Examples

```
# Adobe Document Services integration
npm run adobe:setup                    # Configure Adobe credentials
npm run adobe:demo-generation          # Run document generation demo
npm run adobe:example-basic            # Basic PDF generation example

# Confluence integration
adpa confluence oauth2 login
adpa confluence publish --document ./docs/project-charter.md

# SharePoint integration
adpa sharepoint oauth2 login
adpa sharepoint upload --folder "Project Documents" --file ./docs/

# Version control integration
adpa vcs commit --message "Generated project documentation"
adpa vcs push --remote origin
```

🧪 Testing

```
# Run all tests
npm test

# Test specific providers
npm run test:azure
npm run test:github
npm run test:ollama

# Performance testing
npm run test:performance

# Integration testing
npm run test:integration
```

🏢 Enterprise Features

Compliance Standards

- **Financial**: Basel III, MiFID II, FINRA, CFTC, FCA, BaFin
- **Security**: GDPR, SOX, PCI DSS, ISO 27001, ISO 9001
- **Industry**: Healthcare (HIPAA), Government (FedRAMP)

Enterprise Integration

- **Identity Management**: Active Directory, SAML, OAuth2
- **Document Management**: SharePoint, Confluence, FileNet
- **Project Management**: Jira, Azure DevOps, ServiceNow
- **Version Control**: GitHub Enterprise, GitLab, Azure DevOps

Scalability & Performance

- **Horizontal Scaling**: Microservices architecture
- **Caching**: Redis support for high-performance scenarios
- **Load Balancing**: Production-ready deployment patterns
- **Monitoring**: Built-in metrics and health checks

📁 Project Structure

```
requirements-gathering-agent/
├── 📁 src/                          # TypeScript source code
│   ├── 🎯 cli.ts                    # Main CLI entry point
│   ├── 🌐 server.ts                 # Express.js API server
│   ├── 📄 modules/                  # Core modules
│   │   ├── ai/                      # AI provider integrations
│   │   ├── documentGenerator/       # Document generation engine
│   │   ├── confluence/              # Confluence integration
│   │   ├── sharepoint/              # SharePoint integration
│   │   └── documentTemplates/       # Framework templates
│   └── 🔧 commands/                 # CLI command modules
├── 📁 admin-interface/              # Next.js admin portal
├── 📁 api-specs/                    # TypeSpec API specifications
├── 📁 docs/                         # Comprehensive documentation
├── 📁 test/                         # Test suites
```

```
├── 📁 generated-documents/          # Output directory
└── 📁 dist/                         # Compiled JavaScript
```

🤝 Contributing

We welcome contributions! Please see our Contributing Guide for details.

Development Setup

```
git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run dev          # Start development mode
npm run build        # Build for production
npm test             # Run tests
```

Code Standards

- **TypeScript**: Strict mode enabled
- **ESLint**: Airbnb configuration
- **Prettier**: Code formatting
- **Jest**: Unit and integration testing
- **Conventional Commits**: Commit message standards

📋 Roadmap

Q1 2025

- ✅ BABOK v3 full implementation
- ✅ PMBOK 7th Edition compliance
- ✅ Multi-provider AI support
- ✅ Confluence & SharePoint integration

Q2 2025

- 🚧 DMBOK 2.0 implementation
- 🔄 Docker containerization
- 🔄 Kubernetes deployment templates
- 🔄 Advanced analytics dashboard

Q3 2025

- 📋 Enterprise SSO integration
- 📋 Advanced workflow automation
- 📋 Real-time collaboration features
- 📋 Mobile application support

📞 Support & Documentation

- 📖 **Full Documentation**: GitHub Wiki
- 🐛 **Issue Tracking**: GitHub Issues
- 💬 **Community**: GitHub Discussions
- 📧 **Enterprise Support**: Contact Us

📄 License

This project is licensed under the MIT License - see the LICENSE file for details.

🙏 Acknowledgments

- **Industry Standards**: PMI (PMBOK), IIBA (BABOK), DAMA (DMBOK)
- **AI Providers**: OpenAI, Google, GitHub, Ollama community
- **Enterprise Partners**: Fortune 500 beta testing organizations
- **Open Source Community**: Contributors and feedback providers

---

Built with ❤️ for Enterprise Automation

🌟 Star us on GitHub | 📦 npm Package | 📖 Documentation

=== PROJECT METADATA ===

Name: adpa-enterprise-framework-automation

Description: Modular, standards-compliant Node.js/TypeScript automation framework for enterprise requirements, project, and data management. Provides CLI and API for BABOK v3, PMBOK 7th Edition, and DMBOK 2.0 (in progress). Production-ready Express.js API with TypeSpec architecture. Designed for secure, scalable, and maintainable enterprise automation.

Version: 3.2.0

Dependencies: @adobe/pdfservices-node-sdk, @azure-rest/ai-inference, @azure/identity, @azure/msal-node, @azure/openai, @google/generative-ai, @microsoft/microsoft-graph-client, axios, bcryptjs, compression, cors, dotenv, express, express-rate-limit, express-validator, express-winston, form-data, glob, helmet, joi, jsonwebtoken, morgan, multer, node-fetch, openai, swagger-ui-express, ts-node, uuid, winston, yargs, zod

Dev Dependencies: @jest/globals, @redocly/cli, @types/bcryptjs, @types/compression, @types/cors, @types/express, @types/glob, @types/jest, @types/jsonwebtoken, @types/morgan, @types/multer, @types/node, @types/node-fetch, @types/swagger-ui-express, @types/uuid, @typespec/compiler, @typespec/http, @typespec/json-schema, @typespec/openapi3, @typespec/rest, ajv, jest, rimraf, ts-jest, typescript, webpack-cli

Available Scripts: build, copy-configs, start, api:start, dev, clean, test, test:providers, test:performance, test:azure, test:github, test:ollama, test:failover, test:unit, prepublishOnly, admin:install, admin:dev, admin:build, admin:start, admin:setup, admin:serve, confluence:init, confluence:test, confluence:oauth2:login, confluence:oauth2:status, confluence:oauth2:debug, confluence:publish, confluence:status, sharepoint:init, sharepoint:test, sharepoint:oauth2:login, sharepoint:oauth2:status, sharepoint:oauth2:debug, sharepoint:publish, sharepoint:status, api:compile, api:watch, api:format, api:lint, api:docs, api:serve-docs, api:demo, api:server, babok:generate, pmbok:generate, dmbok:generate, framework:multi

=== PHASE-2-IMPLEMENTATION-GUIDE.MD (documentation) ===
Path: docs\ADOBE\PHASE-2-IMPLEMENTATION-GUIDE.md
Relevance Score: 95

Adobe Creative Suite Phase 2 Implementation Guide

Date: July 8, 2025
Status: 📋 **READY FOR IMPLEMENTATION**
Prerequisites: ✅ Phase 1 Complete (58 PDFs generated successfully)

🎯 Phase 2 Objectives

Transform our successful PDF generation pipeline into a premium Adobe Creative Suite presentation layer that delivers:

- **Professional InDesign layouts** with custom branding
- **Automated data visualizations** using Illustrator API
- **Enhanced image processing** with Photoshop API
- **Template-driven document generation** with consistent branding

🏗️ Technical Architecture for Phase 2

Current State (Phase 1) ✅

```
Markdown → Puppeteer → Professional PDF
├── Professional styling
├── Corporate typography
├── Metadata and attribution
└── Print-ready output
```

Target State (Phase 2) 🎯

```
Markdown → Content Analysis → Template Selection → Adobe Creative API
├── InDesign Server (Layout & Typography)
├── Illustrator API (Charts & Infographics)
├── Photoshop API (Image Enhancement)
└── Document Generation (Template Processing)
```

📋 Implementation Milestones

Milestone 1: Adobe Creative SDK Setup (Priority 1)

1.1 Authentication & Credentials

Files to Create:

- `src/adobe/creative-suite/authenticator.ts`
- `src/adobe/creative-suite/config.ts`
- `.env.adobe.creative`

Implementation Steps:

1. **Adobe Creative SDK Registration**

```
# Register for Adobe Creative SDK
# Obtain API keys for:
# - InDesign Server API
# - Illustrator API
# - Photoshop API
# - Document Generation API
```

2. **Authentication Setup**

```
// src/adobe/creative-suite/authenticator.ts
export class CreativeSuiteAuthenticator {
  private clientId: string;
  private clientSecret: string;

```

```
  async authenticate(): Promise<string> {
    // Implement OAuth 2.0 flow for Creative Suite APIs
  }
}
```

... [truncated]

=== ARCHITECTURE.MD (development) ===
Path: docs\ARCHITECTURE.md
Relevance Score: 95

Requirements Gathering Agent - Architecture Documentation

Overview

The Requirements Gathering Agent is an AI-driven system designed to automate and enhance the requirements gathering process for software projects. It leverages multiple AI providers and context management techniques to generate comprehensive project documentation, user stories, and strategic planning artifacts.

System Architecture

Core Components

1. Context Management System

- **Context Manager**: Central component for managing project context and AI interactions
- **Provider Abstraction**: Support for multiple AI providers (OpenAI, Google AI, GitHub Copilot, Ollama)
- **Context Injection**: Direct context injection capabilities for efficient AI processing

2. AI Provider Integration

- **Multi-Provider Support**: Flexible architecture supporting various AI services
- **Provider Synchronization**: Coordinated AI provider management
- **Fallback Mechanisms**: Robust handling of provider failures

3. Document Generation Engine

- **Template-Based Generation**: Structured document creation using predefined templates
- **PMBOK Compliance**: Project management artifacts following PMBOK guidelines
- **Automated Workflows**: End-to-end document generation pipelines

4. CLI Interface

- **Command-Line Tools**: `cli.ts` and `cli-main.ts` for system interaction
- **Batch Processing**: Support for bulk document generation
- **Configuration Management**: Flexible configuration options

Technology Stack

Core Technologies

- **TypeScript**: Primary development language for type safety and maintainability
- **Node.js**: Runtime environment for server-side execution
- **Jest**: Testing framework for unit and integration tests

AI Integration

- **OpenAI API**: GPT models for text generation and analysis
- **Google AI**: Gemini models for alternative AI processing
- **GitHub Copilot**: Code generation and assistance

- **Ollama**:
  ... [truncated]

=== API-TESTING-COMPREHENSIVE-SUMMARY.MD (development) ===
Path: docs\AZURE\API-TESTING-COMPREHENSIVE-SUMMARY.md
Relevance Score: 95

ADPA API Testing Comprehensive Summary

Test Session Report - June 22, 2025

🎯 TESTING OVERVIEW

Duration: 1 hour testing session
API Server: Express.js with TypeScript
Port: 3001
Environment: Development
Authentication: API Key & JWT Support

✅ SUCCESSFUL TESTS

1. **Health Endpoints** - ALL PASSED ✓

- **Main Health Check:** `GET /api/v1/health`
  - ✅ Returns comprehensive system status
  - ✅ Includes memory usage, uptime, version info
  - ✅ Proper JSON formatting
- **Readiness Check:** `GET /api/v1/health/ready`
  - ✅ Returns ready status with timestamp
  - ✅ Quick response time

2. **Authentication & Security - ALL PASSED ✓**

- **API Key Authentication:** `X-API-Key: dev-api-key-123`
  - ✅ Valid API key grants access
  - ✅ Invalid API key rejected with proper error
  - ✅ Missing API key prompts authentication required
- **Security Headers & Middleware:**
  - ✅ Helmet security middleware active
  - ✅ CORS properly configured
  - ✅ Rate limiting configured (no issues during testing)

3. **Templates API - ALL PASSED ✓**

- **Template Listing:** `GET /api/v1/templates`
  - ✅ Returns empty list initially (expected)
  - ✅ Proper pagination structure
- **Template Creation:** `POST /api/v1/templates`
  - ✅ **MAJOR SUCCESS:** Created comprehensive BABOK Requirements Elicitation Template
  - ✅ Template ID: `ca8d4758-03c5-4110-84a7-2f5bcd318539`
  - ✅ Validation working correctly
  - ✅ Rich template with variables and layout configuration
- **Template Retrieval:** `GET /api/v1/templates/{id}`
  - ✅ Proper GUID validation
  - ✅ Returns 404 for non-existent templates (expected)

4. **Documents API - ALL PASSED ✓**

- **Document Jobs Listing:** `GET /api/v1/documents/jobs`
  - ✅ Returns proper pagination structure
  - ✅ Authentication required and working

- **Document Conversion:** `POST /api/v1/documents/convert`
    - ✅ **MAJOR SUCCESS:** Ge
      ... [truncated]

=== AZURE-PORTAL-API-CENTER-SETUP-GUIDE.MD (primary) ===
Path: docs\AZURE\AZURE-PORTAL-API-CENTER-SETUP-GUIDE.md
Relevance Score: 95

Azure Portal API Center Setup Guide

Standards Compliance & Deviation Analysis API

🎯 Portal-Based Deployment Strategy

Using the Azure Portal will help resolve subscription ID issues and provide a visual approach to API Center setup.

Step 1: Access Azure Portal

Navigate to Azure API Center

1. **Open**: Azure Portal
2. **Search**: "API Center" in the top search bar
3. **Select**: "API Centers" from the results

Verify Subscription Access

- **Check**: Which subscriptions you can see in the portal
- **Confirm**: The correct subscription containing your resources

- **Note**: The actual subscription ID for CLI alignment

Step 2: Create/Verify API Center Instance

Option A: Create New API Center

If `svc-api-center` doesn't exist:

1. **Click**: "Create API Center"
2. **Subscription**: Select the correct active subscription
3. **Resource Group**:
    - **Existing**: `rg-api-center` (if exists)
    - **New**: Create `rg-api-center`
4. **API Center Name**: `svc-api-center`
5. **Region**: **West Europe** (`westeu`)
6. **Pricing Tier**: Start with Standard
7. **Click**: "Review + Create" → "Create"

Option B: Use Existing API Center

If it already exists:

1. **Navigate**: to existing `svc-api-center`
2. **Note**: Subscription ID and Resource Group (`rg-api-center`)
3. **Verify**: Access and permissions

Step 3: Create APIs via Portal

3.1 Create Echo API

1. **Navigate**: to your `svc-api-center` API Center instance
2. **Click**: "APIs" in the left menu
3. **Click**: "Create API"
4. **Fill Details**:
    - **API ID**: `echo-api`

- **Title**: `Echo API`
- **Type**: `REST`
- **Description**: `Simple echo API for testing`

5. **Click**: "Create"

3.2 Create Standards Compliance API

1. **Click**: "Create API" again
2. **Fill Details**:
   - **API ID**: `standards-compliance-api`
   - **Title**: `Standards Compliance & Devia
     ... [truncated]

=== AZURE-PORTAL-API-REGISTRATION-GUIDE.MD (development) ===
Path: docs\AZURE\AZURE-PORTAL-API-REGISTRATION-GUIDE.md
Relevance Score: 95

Azure Portal API Registration Guide

Manual API Center Setup - No CLI Required

🎯 Why Portal Registration is Perfect for You

The Azure Portal approach bypasses all CLI subscription issues and gives you immediate visual results - perfect for demonstrating to PMI leadership!

Step 1: Access Azure Portal

Navigate to API Centers

1. **Open**: Azure Portal
2. **Sign in** with your Azure account
3. **Search**: "API Center" in the top search bar
4. **Select**: "API Centers" from the dropdown

Find Your API Center

- **Look for**: `svc-api-center` in `rg-api-center`
- **Or**: Create new if it doesn't exist

Step 2: Register Your APIs in Portal

2.1 Register Echo API

1. **Navigate**: to your API Center (`svc-api-center`)
2. **Click**: "APIs" in the left navigation menu
3. **Click**: "Register API" or "Add API" button
4. **Fill in the form**:

```
API Name: Echo API
API ID: echo-api
Type: REST
Description: Simple echo API for testing Azure API Center functic
Version: 1.0
```

5. **Click**: "Register" or "Create"

2.2 Register Standards Compliance API

1. **Click**: "Register API" again
2. **Fill in the form**:

```
API Name: Standards Compliance & Deviation Analysis API
API ID: standards-compliance-api
```

```
Type: REST
Description: PMI PMBOK and BABOK standards compliance analysis wi
Version: 1.0
Tags: pmi, pmbok, babok, compliance, governance, standards
```

3. **Click**: "Register" or "Create"

Step 3: Add API Specifications

Upload OpenAPI Specification

1. **Select**: your `standards-compliance-api` from the list
2. **Click**: "API definitions" or "Specifications" tab
3. **Click**: "Add definition" or "Upload specification"
4. **Choose**: "OpenAPI" as the specification type
5. **Upload method options**:

**Option

... [truncated]

=== BABOK-ENTERPRISE-DEMONSTRATION-GUIDE.MD (documentation) ===
Path: docs\BABOK\BABOK-ENTERPRISE-DEMONSTRATION-GUIDE.md
Relevance Score: 95

🎯 BABOK Enterprise Consulting Demonstration

Step-by-Step Guide to Professional Business Analysis Automation

📋 DEMONSTRATION OVERVIEW

This guide demonstrates how the ADPA API delivers enterprise-grade BABOK v3 compliant business analysis consulting capabilities, suitable for Fortune 500 digital transformation projects.

🚀 STEP 1: API SERVER INITIALIZATION

1.1 Start the Enterprise API Server

```
# Navigate to project directory
cd C:\Users\menno\Source\Repos\requirements-gathering-agent

# Build the production-ready API
npm run api:build

# Start the enterprise API server
npm run api:server
```

Expected Output:

```
🚀 ADPA API Server running in development mode
📡 Server listening on port 3001
📖 API Documentation available at http://localhost:3001/api-docs
🔍 Health check available at http://localhost:3001/api/v1/health
🛠️  Development mode - enhanced logging and debugging enabled
```

1.2 Verify API Health & Capabilities

```
curl http://localhost:3001/api/v1/health
```

Enterprise-Grade Response:

```
{
  "status": "healthy",
  "timestamp": "2025-06-22T13:30:00.000Z",
  "version": "2.2.0",
  "environment": "development",
  "uptime": 45.2,
  "memory": {"used": 12, "total": 14, "external": 2},
  "node": "v20.18.2"
}
```

📊 STEP 2: ENTERPRISE TEMPLATE CREATION

2.1 Create BABOK v3 Requirements Elicitation Template

File: `enterprise-babok-template.json`


```
{
  "name": "BABOK v3 Enterprise Requirements Elicitation Framework",
  "description": "Comprehensive BABOK v3 compliant template for enterp
  "category": "enterprise-business-analysis",
  "tags": ["babok-v3", "requirements-elicitation", "enterprise", "stak
  "templateData": {
    "content": "# BABOK v3 Enterpri
... [truncated]

=== COLLABORATION-TOOLS-ROADMAP.MD (planning) ===
Path: docs\COLLABORATION-TOOLS-ROADMAP.md
Relevance Score: 95

# Collaboration Tools Development Roadmap

## Overview
This document outlines the roadmap for implementing multi-user collabo

## Current Capabilities
```

```
- ✅ **Single-user CLI interface**: Full functionality for individual
- ✅ **RESTful API**: Multi-client architecture ready
- ✅ **Authentication**: Bearer token system implemented
- ✅ **Document Management**: Template and output management
- ✅ **Standards Compliance**: Individual project analysis

## Collaboration Features Architecture

### Multi-User Management System

#### User Roles & Permissions
```typescript
interface UserRole {
  id: string;
  name: 'admin' | 'project_manager' | 'business_analyst' | 'stakeholde
  permissions: Permission[];
}

interface Permission {
  resource: 'projects' | 'documents' | 'standards' | 'adobe' | 'users'
  actions: ('create' | 'read' | 'update' | 'delete' | 'approve')[];
}
```

Team Management

```
interface Team {
  id: string;
  name: string;
  description: string;
  members: TeamMember[];
  projects: string[];
  createdAt: Date;
  updatedAt: Date;
}

interface TeamMember {
  userId: string;
  role: UserRole;
  joinedAt: Date;
```
```

```
  permissions: Permission[];
}
```

Real-time Collaboration Features

1. Concurrent Document Editing

- **Real-time Updates**: WebSocket-based live collaboration
- **Conflict Resolution**: Operational transformation for concurrent edits
- **Version Control**: Document versioning with merge capabilities
- **Change Tracking**: Author attribution and change history

2. Project Sharing & Permissions

- **Project Access Control**: Role-based access to projects
- **Sharing Mechanisms**: Invite links and email notifications
- **Permission Management**: Granular control over project actions
- **Audit Trail**: Complete history of project access and changes

3. Approval Workflows

``

... [truncated]

=== IMPLEMENTATION-GUIDE-PROVIDER-CHOICE-MENU.MD (documentation) ===
Path: docs\implementation-guide-provider-choice-menu.md
Relevance Score: 95

Interactive AI Provider Selection Menu - Implementation Guide

Document Version: 1.0
Created: December 2024

Last Updated: December 2024
Target Audience: Developers, Technical Leads, Product Managers

---

📋 Table of Contents

1. Overview
2. Current System Analysis
3. Implementation Strategy
4. Interactive Choice Menu Design
5. Code Implementation
6. Integration with Existing System
7. User Experience Flow
8. Error Handling & Validation
9. Testing Strategy
10. Migration Guide
11. Best Practices
12. Troubleshooting

📖 Overview

This guide provides comprehensive documentation for implementing an interactive choice menu that allows users to select an AI provider before running the Requirements Gathering Agent. The feature enhances user experience by providing a visual selection interface instead of requiring manual environment configuration.

🎯 Objectives

- **Simplify Provider Selection**: Replace manual `.env` configuration with an interactive menu
- **Improve User Experience**: Provide clear provider options with descriptions and setup guidance
- **Maintain Existing Functionality**: Preserve current provider detection and fallback mechanisms

- **Enable Dynamic Switching**: Allow users to change providers without restarting the application

🔧 Key Features

- Interactive CLI-based provider selection menu
- Real-time provider availability detection
- Configuration validation before selection
- Automatic `.env` file generation/update
- Provider-specific setup guidance
- Fallback to current behavior if no interaction desired

🔍 Current System Analysis

Existing Provi

... [truncated]

=== SHAREPOINT-USAGE-GUIDE.MD (documentation) ===
Path: docs\SHAREPOINT-USAGE-GUIDE.md
Relevance Score: 95

SharePoint Integration Usage Guide

Overview

The SharePoint integration in Requirements Gathering Agent v2.1.3 enables you to automatically publish generated documents to SharePoint Online document libraries. This feature provides enterprise-grade document management with Azure authentication, metadata tagging, and version control.

Features

- **Microsoft Graph API Integration**: Secure, enterprise-grade authentication
- **OAuth2 Authentication**: Azure AD integration with device code flow
- **Automatic Folder Creation**: Creates organized folder structures
- **Metadata Management**: Adds custom metadata to published documents
- **Batch Publishing**: Efficiently publish multiple documents
- **Version Control**: SharePoint's built-in versioning support
- **Enterprise Security**: Follows Azure security best practices

Quick Start

1. Prerequisites

Before using SharePoint integration, ensure you have:

- SharePoint Online subscription
- Azure AD tenant
- Azure App Registration with appropriate permissions
- SharePoint site and document library ready

2. Azure App Registration Setup

1. **Create App Registration in Azure Portal**:
    - Go to Azure Portal → Azure Active Directory → App registrations
    - Click "New registration"
    - Name: "Requirements Gathering Agent"
    - Supported account types: "Accounts in this organizational directory only"
    - Redirect URI: `http://localhost:3000/auth/callback`

2. **Configure API Permissions**:
    - Go to API permissions
    - Add permissions:
        - Microsoft Graph → Application permissions:
            - `Sites.ReadWrite.All`
            - `Files.ReadWrite.All`
            - `User.Read`
3. **Grant Admin Consent**:
    - Click "Grant admin consent for [Your Tenant]"
4. **Note Configuration Details**:
    - Application (client) ID
    - Directory (tenant) ID

3. Initialize SharePoint Configuration

```
# Initialize SharePoint configuration
npm run sharepoint:in
... [truncated]

=== PHASE-2-IMPLEMENTATION-GUIDE.MD (documentation) ===
Path: docs\ADOBE\PHASE-2-IMPLEMENTATION-GUIDE.md
Relevance Score: 95

# Adobe Creative Suite Phase 2 Implementation Guide

**Date:** July 8, 2025
**Status:** 📋 **READY FOR IMPLEMENTATION**
**Prerequisites:** ✅ Phase 1 Complete (58 PDFs generated successfully

---

## 🎯 Phase 2 Objectives

Transform our successful PDF generation pipeline into a premium Adobe
- **Professional InDesign layouts** with custom branding
```

```
- **Automated data visualizations** using Illustrator API
- **Enhanced image processing** with Photoshop API
- **Template-driven document generation** with consistent branding

## 🏗️ Technical Architecture for Phase 2

### Current State (Phase 1) ✅
```

Markdown → Puppeteer → Professional PDF
├── Professional styling
├── Corporate typography
├── Metadata and attribution
└── Print-ready output

```
### Target State (Phase 2) 🎯
```

Markdown → Content Analysis → Template Selection → Adobe Creative APIs → Premium Output
├── InDesign Server (Layout & Typography)
├── Illustrator API (Charts & Infographics)
├── Photoshop API (Image Enhancement)
└── Document Generation (Template Processing)

```
## 📋 Implementation Milestones

### Milestone 1: Adobe Creative SDK Setup (Priority 1)

#### 1.1 Authentication & Credentials
**Files to Create:**
- `src/adobe/creative-suite/authenticator.ts`
- `src/adobe/creative-suite/config.ts`
- `.env.adobe.creative`

**Implementation Steps:**
1. **Adobe Creative SDK Registration**
   ```bash
```
```

```
# Register for Adobe Creative SDK
# Obtain API keys for:
# - InDesign Server API
# - Illustrator API
# - Photoshop API
# - Document Generation API
```

2. **Authentication Setup**

```
// src/adobe/creative-suite/authenticator.ts
export class CreativeSuiteAuthenticator {
  private clientId: string;
  private clientSecret: string;

  async authenticate(): Promise<string> {
    // Implement OAuth 2.0 flow for Creative Suite APIs
  }
}
```

... [truncated]

.

A systematic approach to creating, managing, and maintaining test environments that support quality assurance activities and ensure reliable testing outcomes.

2. Test Environment Objectives

Primary Objectives

- Provide stable, reliable environments for all testing activities
- Ensure environment consistency across different test phases
- Support parallel testing activities without conflicts
- Enable rapid environment provisioning and recovery
- Maintain production-like conditions for accurate testing

Success Criteria

- 99.5% environment availability during testing phases
- Environment provisioning within 2 hours for standard setups
- Configuration consistency across all test environments
- Zero data leakage between environments
- Successful test execution without environment-related failures

3. Environment Architecture

3.1 Overall Architecture

Core Infrastructure Components

Application Tier

- Web servers (Load balanced)
- Application servers
- Background processing services
- API gateways and proxy servers

Database Tier

- Primary database instances
- Read replicas for performance testing
- Cache layers (Redis/Memcached)
- Search engines (Elasticsearch/Solr)

Integration Tier

- Message queues and brokers
- Third-party service simulators
- API mock servers
- File storage systems

Monitoring and Logging

- Application performance monitoring
- Log aggregation and analysis
- Error tracking and alerting

- Resource utilization monitoring

3.2 Network Architecture

- **Isolation:** Separate network segments for each environment
- **Security:** Firewall rules and access controls
- **Connectivity:** VPN access for remote testing
- **Load Balancing:** Distribute traffic across multiple instances

4. Environment Types

4.1 Development Environment

- **Purpose:** Individual developer testing and debugging
- **Configuration:** Lightweight, single-instance setup
- **Data:** Synthetic data for basic functionality testing
- **Access:** Direct developer access
- **Availability:** 24/7 with individual developer control

4.2 Integration Testing Environment

- **Purpose:** Component integration and API testing
- **Configuration:** Multi-service setup with realistic integrations
- **Data:** Comprehensive test datasets
- **Access:** Development and QA teams
- **Availability:** Business hours with scheduled maintenance windows

4.3 System Testing Environment

- **Purpose:** End-to-end system functionality testing
- **Configuration:** Production-like setup with full feature set
- **Data:** Production-like data volumes and complexity
- **Access:** QA team and business analysts
- **Availability:** Business hours with high availability requirements

4.4 Performance Testing Environment

- **Purpose:** Load, stress, and performance testing
- **Configuration:** Production-scale infrastructure
- **Data:** Production volumes with anonymized content
- **Access:** Performance testing team
- **Availability:** Scheduled testing windows with dedicated resources

4.5 User Acceptance Testing (UAT) Environment

- **Purpose:** Business user validation and acceptance testing
- **Configuration:** Production-identical setup
- **Data:** Production-like data with privacy compliance
- **Access:** Business users and stakeholders
- **Availability:** Business hours with high reliability

4.6 Staging Environment

- **Purpose:** Pre-production validation and deployment testing
- **Configuration:** Exact production replica
- **Data:** Production-like data with security controls
- **Access:** Release team and senior stakeholders
- **Availability:** 24/7 with production-level monitoring

5. Environment Configuration Management

5.1 Infrastructure as Code (IaC)

Configuration Management Tools:

- **Terraform:** Infrastructure provisioning
- **Ansible:** Configuration management
- **Docker:** Containerization
- **Kubernetes:** Container orchestration
- **Helm:** Application deployment

Version Control:

- All environment configurations stored in Git
- Branching strategy aligned with environment lifecycle
- Automated configuration deployment
- Change approval workflow for production-like environments

5.2 Environment Provisioning

Automated Provisioning Process:

1. **Resource Allocation:** Compute, storage, and network resources
2. **Base Image Deployment:** Standardized OS and runtime images
3. **Application Deployment:** Automated application installation
4. **Configuration Application:** Environment-specific settings
5. **Service Integration:** Connect to dependent services
6. **Validation Testing:** Automated environment health checks

Provisioning Timeline:

- Development: 30 minutes
- Integration: 1 hour
- System/UAT: 2 hours
- Performance/Staging: 4 hours

5.3 Configuration Standards

Standardization Requirements:

- Consistent naming conventions
- Standardized port assignments
- Common logging configurations
- Unified monitoring setups
- Standard security baselines

Configuration Documentation:

- Environment-specific configuration files
- Deployment procedures and runbooks
- Troubleshooting guides

- Recovery procedures

6. Data Management

6.1 Test Data Strategy

Data Categories:

- **Synthetic Data:** Generated data for functional testing
- **Anonymized Production Data:** Masked real data for realistic testing
- **Static Test Data:** Predefined datasets for specific test scenarios
- **Dynamic Test Data:** Generated during test execution

Data Privacy and Security:

- No production data in non-production environments
- Data masking and anonymization procedures
- Secure data transmission and storage
- Regular data cleanup and purging

6.2 Database Management

Database Setup:

- Separate database instances per environment
- Consistent schema across all environments
- Automated database migrations
- Regular backup and restore procedures

Data Refresh Process:

1. **Schedule:** Weekly refresh for most environments
2. **Anonymization:** Apply data masking rules
3. **Validation:** Verify data integrity and completeness
4. **Notification:** Alert teams of refresh completion

6.3 File and Media Management

- **File Storage:** Dedicated storage for each environment
- **Media Assets:** Test images, documents, and multimedia files
- **Backup Strategy:** Regular backup of critical test assets
- **Access Control:** Role-based access to sensitive test data

7. Access Control and Security

7.1 User Access Management

Access Levels:

- **Administrator:** Full environment control and configuration
- **Developer:** Application deployment and debugging access
- **Tester:** Test execution and result analysis access
- **Viewer:** Read-only access for monitoring and reporting

Authentication and Authorization:

- Single Sign-On (SSO) integration
- Role-based access control (RBAC)
- Multi-factor authentication for sensitive environments
- Regular access review and cleanup

7.2 Network Security

- **Firewall Rules:** Restrict access to necessary ports and services
- **VPN Access:** Secure remote access for authorized users
- **Network Monitoring:** Track and log all network traffic
- **Intrusion Detection:** Monitor for suspicious activities

7.3 Data Security

- **Encryption:** Data encryption at rest and in transit
- **Access Logging:** Comprehensive audit trails
- **Data Loss Prevention:** Prevent unauthorized data extraction

- **Compliance:** Meet regulatory requirements (GDPR, HIPAA, etc.)

8. Environment Monitoring and Maintenance

8.1 Monitoring Strategy

Infrastructure Monitoring:

- Server performance and resource utilization
- Network connectivity and latency
- Storage capacity and performance
- Database performance and availability

Application Monitoring:

- Application response times
- Error rates and exception tracking
- User session monitoring
- API performance metrics

Monitoring Tools:

- **Infrastructure:** Nagios, Zabbix, or DataDog
- **Application:** New Relic, AppDynamics, or Dynatrace
- **Logs:** ELK Stack or Splunk
- **Synthetic Monitoring:** Pingdom or Uptime Robot

8.2 Maintenance Procedures

Regular Maintenance:

- **Daily:** Health checks and log reviews
- **Weekly:** Performance analysis and optimization
- **Monthly:** Security updates and patches
- **Quarterly:** Capacity planning and scaling review

Maintenance Windows:

- **Scheduled:** Planned maintenance during off-hours
- **Emergency:** Immediate response for critical issues
- **Communication:** Advance notice to all stakeholders
- **Rollback:** Procedures for reverting changes if needed

8.3 Backup and Recovery

Backup Strategy:

- **Frequency:** Daily incremental, weekly full backups
- **Retention:** 30-day backup retention policy
- **Storage:** Offsite backup storage for disaster recovery
- **Testing:** Monthly backup restoration testing

Recovery Procedures:

1. **Incident Assessment:** Evaluate scope and impact
2. **Recovery Planning:** Determine recovery approach
3. **Data Restoration:** Restore from most recent backup
4. **Service Validation:** Verify environment functionality
5. **Communication:** Update stakeholders on recovery status

9. Deployment and Release Management

9.1 Deployment Pipeline

Continuous Integration/Continuous Deployment (CI/CD):

- **Source Control:** Git-based version control
- **Build Automation:** Automated build and testing
- **Deployment Automation:** Automated deployment to test environments
- **Quality Gates:** Automated quality checks before promotion

Environment Promotion Path:
Development → Integration → System → Performance → UAT → Staging → Production

9.2 Release Procedures

Pre-Deployment:

- Environment preparation and validation
- Backup of current environment state
- Stakeholder notification and coordination
- Risk assessment and mitigation planning

Deployment Process:

1. **Deployment Execution:** Automated deployment scripts
2. **Smoke Testing:** Basic functionality validation
3. **Regression Testing:** Automated test suite execution
4. **User Acceptance:** Business user validation
5. **Go/No-Go Decision:** Stakeholder approval for production

9.3 Rollback Procedures

- **Rollback Triggers:** Defined criteria for rollback decisions
- **Rollback Process:** Automated rollback to previous version
- **Data Considerations:** Database rollback and data integrity
- **Communication:** Immediate stakeholder notification

10. Cost Management and Optimization

10.1 Resource Optimization

- **Right-Sizing:** Appropriate resource allocation per environment
- **Auto-Scaling:** Dynamic scaling based on demand
- **Schedule Management:** Automated start/stop for non-24/7 environments
- **Resource Monitoring:** Track utilization and optimize accordingly

10.2 Cost Tracking

- **Environment Costing:** Track costs per environment

- **Resource Utilization:** Monitor and optimize resource usage
- **Budget Management:** Set and monitor budget limits
- **Cost Reporting:** Regular cost analysis and reporting

11. Troubleshooting and Support

11.1 Common Issues and Solutions

Environment Access Issues:

- VPN connectivity problems
- Authentication and authorization failures
- Network connectivity issues
- Service unavailability

Performance Issues:

- Slow response times
- Resource constraints
- Database performance problems
- Network latency issues

Data Issues:

- Data corruption or inconsistency
- Missing or incomplete test data
- Database connection failures
- File system problems

11.2 Support Procedures

Issue Escalation:

1. **Level 1:** Environment administrator
2. **Level 2:** DevOps team
3. **Level 3:** Infrastructure team
4. **Level 4:** Vendor support (if applicable)

Support Channels:

- **Ticketing System:** Primary support channel
- **Chat/Slack:** Real-time communication
- **Email:** Formal communication and documentation
- **Phone:** Emergency support for critical issues

12. Disaster Recovery

12.1 Disaster Recovery Plan

Recovery Objectives:

- **RTO (Recovery Time Objective):** 4 hours maximum downtime
- **RPO (Recovery Point Objective):** 1 hour maximum data loss
- **Business Impact:** Minimize impact on testing schedules

Recovery Procedures:

1. **Incident Declaration:** Identify and declare disaster
2. **Team Activation:** Activate disaster recovery team
3. **Assessment:** Evaluate damage and recovery options
4. **Recovery Execution:** Implement recovery procedures
5. **Validation:** Verify recovered environment functionality
6. **Communication:** Update stakeholders on recovery status

12.2 Business Continuity

- **Alternative Environments:** Backup environments for critical testing
- **Cloud Failover:** Cloud-based disaster recovery options
- **Documentation:** Maintain up-to-date recovery procedures
- **Testing:** Regular disaster recovery testing and validation

13. Metrics and KPIs

13.1 Environment Performance Metrics

- **Availability:** Percentage uptime for each environment
- **Response Time:** Average response time for key services
- **Throughput:** Request processing capacity
- **Error Rate:** Percentage of failed requests or transactions

13.2 Operational Metrics

- **Provisioning Time:** Time to provision new environments
- **Deployment Success Rate:** Percentage of successful deployments
- **Mean Time to Recovery (MTTR):** Average time to resolve issues
- **Cost per Environment:** Monthly cost breakdown by environment

13.3 Quality Metrics

- **Test Environment Stability:** Environment-related test failures
- **Data Quality:** Accuracy and completeness of test data
- **User Satisfaction:** Feedback from testing teams
- **Compliance:** Adherence to security and regulatory requirements

14. Continuous Improvement

14.1 Regular Reviews

- **Monthly:** Environment performance and utilization review
- **Quarterly:** Cost optimization and capacity planning
- **Semi-Annual:** Security assessment and compliance review
- **Annual:** Technology refresh and architecture review

14.2 Feedback and Enhancement

- **User Feedback:** Regular surveys and feedback collection
- **Process Improvement:** Continuous refinement of procedures
- **Technology Updates:** Adoption of new tools and technologies
- **Best Practices:** Industry best practice implementation

This Test Environment Setup and Management document should be regularly updated to reflect infrastructure changes and operational improvements.