

Develop Schedule Input

Source File: generated-documents\basic-docs\develop-schedule-input.md

Generated: 15/07/2025 at 11:35:16

Generated by: Requirements Gathering Agent - PDF Converter

DevelopScheduleInput

Generated by adpa-enterprise-framework-automation v3.2.0

Category: basic-docs

Generated: 2025-07-14T21:36:37.202Z

Description: Inputs and considerations for developing the project schedule.

Developscheduleinput

Project: ADPA - Advanced Document Processing & Automation Framework

Version: 3.2.0

Document Purpose: Detailed guidance for developing, managing, and tracking the schedule input processes and milestones for the ADPA (formerly Requirements Gathering Agent) project.

1. Introduction

The ADPA Enterprise Framework is a modular, AI-powered automation platform for generating standards-compliant documentation in enterprise environments. It integrates robust project management, business analysis, and data governance frameworks (BABOK v3, PMBOK 7th, DMBOK 2.0), and provides both a CLI and REST API. This Developscheduleinput document defines the process, tools, and checkpoints for capturing, validating, and managing schedule inputs throughout the product lifecycle.

2. Objectives

- **Standardize schedule input collection** across CLI, API, and admin interfaces.
 - Ensure **traceability** of schedule data from input to output artifacts.
 - Enable **multi-framework compliance** with PMBOK 7th and BABOK v3 scheduling standards.
 - Facilitate **automation and integration** with enterprise tools (Confluence, SharePoint, Jira, etc.).
 - Support **real-time collaboration** and version control on schedule inputs.
-

3. Schedule Input Process Overview

3.1 Schedule Input Sources

| Source | Description | Interface |
|----------|---|-------------------------------|
| CLI | Command-line entry of schedule parameters | <code>adpa generate</code> |
| REST API | Programmatic schedule data submission | <code>/api/v1/generate</code> |

| Source | Description | Interface |
|------------------|---|-------------------|
| Admin Web UI | Interactive scheduling forms & templates | Next.js portal |
| Integration APIs | Imports from Jira, Azure DevOps, Confluence, SharePoint | Integration Layer |

3.2 Schedule Input Types

- **Project Charter & High-Level Milestones**
- **Task Breakdown Structures (WBS)**
- **Resource Assignments**
- **Dependencies & Constraints**
- **Schedule Baselines & Updates**
- **Risk-Adjusted Schedules**

4. Schedule Input Data Model

All schedule inputs must adhere to a structured, standards-compliant JSON schema. Example (PMBOK/BABOK hybrid):

```
{
  "projectId": "string",
  "schedule": {
    "milestones": [
      {
        "id": "uuid",
        "name": "Milestone Name",
        "plannedStart": "YYYY-MM-DD",
        "plannedEnd": "YYYY-MM-DD",
        "dependencies": ["milestoneId"],
        "responsible": "userId"
      }
    ]
  }
}
```

```
{
  "tasks": [
    {
      "id": "uuid",
      "name": "Task Name",
      "start": "YYYY-MM-DD",
      "end": "YYYY-MM-DD",
      "assignedTo": ["userId"],
      "status": "Planned|In Progress|Completed",
      "predecessors": ["taskId"]
    }
  ],
  "baseline": {
    "approvedOn": "YYYY-MM-DD",
    "approvedBy": "userId"
  },
  "updates": [
    {
      "date": "YYYY-MM-DD",
      "changes": "string",
      "updatedBy": "userId"
    }
  ]
}
```

5. Schedule Input Methods

5.1 CLI Input

- **Command:**

```
adpa generate --key schedule-input --output ./docs/schedule.md
```

- **Features:**

- Interactive prompts for milestones, tasks, dependencies.
- Optionally load from CSV, Excel, or structured JSON.

5.2 API Submission

- **Endpoint:**
POST /api/v1/generate
- **Payload:**
Schedule data in JSON (see model above).
- **Validation:**
Automatic schema validation; errors returned in JSON format.

5.3 Admin Web UI

- **Module:**
Schedule Input Wizard (Next.js interface)
- **Features:**
 - Form-driven entry of all schedule elements.
 - Real-time validation and Gantt chart preview.
 - Version history and approval workflow.

5.4 Enterprise Integrations

- **Jira/Azure DevOps Import:**
Schedule inputs can be mapped from issues/epics/sprints.
 - **Confluence/SharePoint Sync:**
Export and synchronize schedule documentation.
-

6. Schedule Input Validation

- **Schema Validation:**
All inputs are validated using [zod](#) and/or Joi.
- **Business Rules:**
 - No circular dependencies.
 - Milestone dates must be sequential and within project bounds.
 - Assigned resources must exist in the project's user directory.
 - Baseline updates require change reason and approver.

- **Automated Testing:**

Use `npm test` and `npm run test:integration` to verify schedule input handling.

7. Schedule Input Storage & Versioning

- **Primary Store:**

JSON files in `/generated-documents/` and/or project DB (future: SQL/NoSQL).

- **Change History:**

- All modifications logged with timestamps and user IDs.
- Rollback and comparison supported via CLI/API/Admin UI.

- **Integration with Version Control:**

- `adpa vcs commit` to track schedule changes in Git.
-

8. Schedule Output and Reporting

- **Document Generation:**

- `adpa generate --key project-schedule` produces Markdown, PDF, or Confluence page.

- **API Output:**

- `/api/v1/documents` endpoints provide download links and metadata.

- **Integration Exports:**

- Publish to Confluence/SharePoint using dedicated commands.
-

9. Compliance and Standards Alignment

- **PMBOK 7th Edition:**

- Schedule Management Plan, Baseline, Updates, and Reporting.

- **BABOK v3:**

- Business Analysis Planning, Stakeholder Communication Schedules.

- **Audit Trail:**
 - All schedule changes are traceable for SOX/GDPR compliance.
-

10. Schedule Input Roles & Permissions

- **Role-Based Access:**
 - Only Project Managers and authorized Business Analysts can approve or baseline schedules.
 - Contributors can propose but not approve schedule changes.
 - **Audit Logging:**
 - All actions are logged for compliance.
-

11. Collaboration & Automation

- **Real-Time Collaboration:**
 - Multiple users can edit schedule inputs via Web UI (Q3 2025 roadmap).
 - **Automated Notifications:**
 - Email or system alerts for schedule changes, approvals, and deadlines.
-

12. Practical Guidance & Best Practices

- **Start with the CLI or Admin UI for initial schedule input.**
 - **Validate data early:** Use the built-in validation before baselining.
 - **Integrate with Jira/DevOps** for automated task/milestone population.
 - **Leverage version control** for traceability and rollback.
 - **Export and share** schedules via Confluence/SharePoint for stakeholder transparency.
 - **Use API for automation** in CI/CD or enterprise integration scenarios.
-

13. Roadmap & Continuous Improvement

- **Q2 2025:**
 - Enhanced DMBOK schedule input support.
 - Dockerized deployment of scheduling modules.
 - **Q3 2025:**
 - Real-time multi-user editing.
 - Mobile schedule input and approval.
 - **Feedback:**
 - Submit improvement ideas via [GitHub Issues](#).
-

14. References

- [PMBOK 7th Edition: Schedule Management](#)
 - [BABOK v3: Business Analysis Planning](#)
 - [ADPA Documentation](#)
 - [Contributing Guide](#)
-

Contact:

For support or customization, reach out via [GitHub Discussions](#) or [email](#).

This Developscheduleinput document ensures that ADPA users and developers can efficiently and compliantly manage all project schedule inputs across the full lifecycle, maximizing automation, traceability, and enterprise value.
