Monitor And Control Project Work

> **Source File:** generated-documents\pmbok\monitor-and-control-project-work.md
> **Generated:** 30/07/2025 at 06:59:32
> **Generated by:** Requirements Gathering Agent - PDF Converter

MonitorAndControlProjectWork

Generated by adpa-enterprise-framework-automation v3.2.0
Category: pmbok
Generated: 2025-07-14T21:32:27.672Z
Description:

Monitorandcontrolprojectwork

Project: ADPA – Advanced Document Processing & Automation Framework
Version: 3.2.0
Prepared by: Project Management Office
Date: July 2025

---

1. Purpose

This document defines the Monitor and Control Project Work process for the ADPA (Advanced Document Processing & Automation Framework) project. It describes the mechanisms, tools, metrics, and controls by which project deliverables, processes, and risks are tracked, reviewed, and corrected to ensure alignment with the project management plan, enterprise standards (PMBOK 7th Edition), and business objectives.

---

2. Scope of Monitoring and Control

The Monitor and Control phase applies to all workstreams within ADPA, including:

- Core platform development (Node.js/TypeScript)
- AI integration (OpenAI, Google AI, GitHub Copilot, Ollama, Azure OpenAI)
- Standards-compliant document generation (BABOK v3, PMBOK 7th Edition, DMBOK 2.0)
- API and CLI development
- Enterprise integrations (Confluence, SharePoint, Adobe, SSO)
- Regulatory compliance and security
- Roadmap features and releases

---

3. Monitoring Methods

3.1 Performance Metrics

Area	Key Metrics	Method/Frequency
Schedule	Milestone completion, sprint burndown, roadmap adherence	Weekly review; Jira board

Area	Key Metrics	Method/Frequency
Scope	Requirements traceability, scope change log	Bi-weekly; PM review
Quality	Test pass rates, code coverage, code review findings	CI/CD pipeline; per commit
Cost (if applicable)	Resource hours, tooling costs, cloud/API usage	Monthly; PM report
Security & Compliance	Audit logs, dependency scans, compliance checklists	Automated; monthly review
Integration/Deployment	API uptime, API test pass rates, deployment success/failures	Per deployment; dashboards
User Feedback	Issue tracker, community discussions, GitHub stars/forks	Continuous; monthly summary

3.2 Tools and Systems

- **Jira/Azure DevOps:** Issue, sprint, and release tracking
- **GitHub:** Source control, pull request reviews, code standards enforcement
- **CI/CD (GitHub Actions, npm scripts):** Automated testing, builds, deployments
- **Swagger UI/TypeSpec:** API documentation verification
- **Express.js Health Endpoints:** Real-time monitoring (`/api/v1/health`)
- **Test Suites:** Jest unit/integration/performance test runs
- **Security:** Helmet, CORS, rate limiting, dependency audit tools
- **Reporting:** Automated dashboards for test coverage, deployment status, and API health

---

4. Control Processes

4.1 Change Control

- All changes to requirements, features, or major architecture must be submitted as a GitHub Issue or Jira ticket.
- Changes are reviewed by the Change Control Board (lead developer, architect, product owner).
- Approved changes are reflected in the project plan and communicated to all stakeholders.

4.2 Quality Assurance

- **Automated Testing:** All code must pass unit, integration, and performance tests before merging.
- **Manual Verification:** Key features (e.g., Confluence/SharePoint/Adobe integration) are manually tested pre-release.
- **Peer Review:** Pull requests require at least one peer approval and compliance with lint/formatting rules.

4.3 Risk Monitoring

- **Risk Register:** Maintained in project management system; reviewed bi-weekly.
- **Key Risks:** AI API rate limits, provider outages, compliance gaps, security vulnerabilities, integration drift.
- **Mitigation Actions:** Multi-provider AI failover, automated dependency updates, regular compliance checks.

4.4 Issue Tracking and Response

- **Bugs/Incidents:** Logged via GitHub Issues or Jira.
- **SLAs:** High-severity issues responded to within 24 hours; fixes prioritized in next sprint.
- **Postmortems:** Conducted for all critical incidents.

4.5 Compliance and Security

- **Dependency Scanning:** Automated checks using npm audit, Snyk, and GitHub Dependabot.
- **API Security:** Monitored via security middleware and endpoint health checks.
- **Regulatory Review:** Quarterly audits against GDPR, SOX, PCI DSS, and other required standards.

---

5. Reporting and Communication

5.1 Status Reporting

- **Weekly Project Status Report:** Distributed to stakeholders, highlighting progress, issues, risks, and upcoming milestones.
- **Sprint Reviews:** Demos of new/updated features at the end of each sprint.
- **Release Notes:** Published with each major/minor version update.

5.2 Stakeholder Communication

- **Channels:** Email, Slack/MS Teams, GitHub Discussions, scheduled review meetings.
- **Documentation:** All deliverables and process documentation in GitHub Wiki and `/docs` directory.
- **Enterprise Support:** Dedicated contact for high-priority support (see README for details).

---

6. Special Considerations for ADPA

6.1 Multi-Provider AI Monitoring

- **Redundancy:** Automated provider failover tested regularly; fallback scenarios documented.
- **Logging:** Detailed logs for AI provider performance and errors.
- **Provider Selection:** Interactive menu and configuration validation monitored via usage analytics.

6.2 Standards Compliance

- **Templates:** All generated documents validated against BABOK v3 and PMBOK 7th Edition checklists.
- **Deviation Analysis:** Automated compliance reporting for project artifacts; deviations logged and reviewed.

6.3 Integration Monitoring

- **Confluence/SharePoint/Adobe:** API usage, authentication status, and publishing logs monitored per integration.
- **Version Control:** All document and template changes tracked via Git for full auditability.

6.4 Security and Access

- **Authentication:** All API and admin interfaces require JWT/OAuth2; access logs reviewed monthly.
- **Permissions:** Role-based access controls enforced on all sensitive operations.

6.5 Scalability and Performance

- **Load Testing:** Performed prior to major releases; results reviewed for infrastructure scaling.
- **Health Checks:** Automated probes in place for microservice uptime and response times.

---

7. Continuous Improvement

- **Lessons Learned:** Collected at the end of each phase and after major incidents; incorporated into process updates.
- **Feedback Loops:** User and contributor feedback reviewed monthly for actionable improvements.
- **Roadmap Adjustments:** Roadmap updated quarterly based on monitoring insights and stakeholder input.

---

8. Appendix

8.1 Monitoring Checklist

- ☑ All sprints tracked in Jira/DevOps
- ☑ Automated test coverage > 85% on all core modules
- ☑ Weekly health check logs reviewed
- ☑ Security scans run for every dependency update
- ☑ Compliance reports generated for all major releases
- ☑ All integration endpoints tested and validated
- ☑ Stakeholder status reports distributed on schedule

8.2 Key Documents

- Project README
- Architecture Documentation
- API Testing Summary
- [Roadmap](See README or `/docs` directory)
- Support & Issue Tracking

This document will be reviewed and updated at the end of each project phase or when major changes are introduced.