Perform Integration Change Control Process

> **Source File:** generated-documents\pmbok\perform-integration-change-control-process.md
> **Generated:** 30/07/2025 at 06:59:35
> **Generated by:** Requirements Gathering Agent - PDF Converter

PerformIntegrationChangeControlProcess

Generated by adpa-enterprise-framework-automation v3.2.0
Category: pmbok
Generated: 2025-07-14T21:34:30.798Z
Description: PMBOK process for managing and controlling project changes.

Performintegrationchangecontrolprocess

Project: ADPA - Advanced Document Processing & Automation Framework
Version: 3.2.0
PMBOK Alignment: 7th Edition – Perform Integrated Change Control Process
Date: [Insert Current Date]
Prepared by: [Project Manager / Change Control Board]

---

1. Purpose

This document defines the Perform Integrated Change Control process for the ADPA (Advanced Document Processing & Automation Framework) project. The objective is to ensure all changes to project artifacts, deliverables, documents, and baselines are systematically identified, documented, evaluated, approved or rejected, implemented, and communicated in accordance with PMBOK 7th Edition

best practices. The process ensures the ADPA framework remains modular, standards-compliant, and enterprise-ready as it evolves to support AI-powered document automation, multi-provider integrations, and regulatory requirements.

---

2. Scope

This process applies to all changes impacting:

- Project documentation (Charter, Scope, Plans, Requirements, etc.)
- Source code (Node.js, TypeScript, Express.js, Next.js, etc.)
- AI provider integrations (OpenAI, Google AI, GitHub Copilot, Ollama, Azure OpenAI)
- Framework templates for BABOK v3, PMBOK 7th Edition, DMBOK 2.0
- REST API and CLI interface specifications
- Enterprise integrations (Confluence, SharePoint, Adobe Document Services, VCS, SSO)
- Security, compliance, and regulatory controls
- Project baselines (scope, cost, schedule, quality)

---

3. Change Control Roles and Responsibilities

Role	Responsibilities
Project Manager	Chairs the Change Control Board (CCB), logs changes, coordinates analysis, and ensures process adherence.
Change Control Board	Evaluates, approves/rejects, and prioritizes change requests; ensures alignment with project objectives.
Technical Lead	Assesses technical feasibility, impacts on architecture, integrations, and security.
QA Lead	Evaluates testing impacts, regression coverage, and compliance risk.
Stakeholder(s)	Review major changes, especially those impacting compliance, integrations, or deliverables.

---

4. Change Control Process Flow

Step 1: Change Identification
Any team member, stakeholder, or automated monitoring tool can submit a change request (CR) via the designated channel (e.g., GitHub Issues, JIRA, Change Request Form).

Step 2: Change Documentation
All CRs must include:

- Description of the change
- Rationale/business case
- Impacted components (code, API, templates, compliance artifacts, etc.)
- Urgency and priority
- Risk assessment (security, compliance, schedule, etc.)
- Proposed implementation approach

Step 3: Change Logging
CRs are logged in the project's issue tracker and assigned a unique identifier. Relevant documentation and code references are attached.

Step 4: Initial Review
Project Manager screens for completeness and urgency. Minor changes may be fast-tracked; significant changes proceed to full CCB review.

Step 5: Impact Analysis

- Technical Lead: Assesses technical feasibility, security, AI model/provider compatibility, and integration impacts.
- QA Lead: Assesses testing, regression, and documentation update needs.
- Compliance Lead: Assesses regulatory (GDPR, SOX, PCI DSS, etc.) and standards (PMBOK/BABOK/DMBOK) implications.
- Schedule and resource impacts are evaluated.

Step 6: CCB Decision

- Approve: Change is authorized for implementation (scope/schedule/cost baselines updated as needed).
- Request More Info: CR returned to originator for clarification.
- Reject: Change is closed with rationale.

Step 7: Change Implementation

- Assigned to responsible team/resource.
- Code changes are developed in feature branches (with traceable commit messages per Conventional Commits).
- Documentation, templates, and configuration files are updated as required.
- Automated and manual tests are updated/executed.

Step 8: Verification and Validation

- QA Lead confirms change meets acceptance criteria.
- Regression and integration tests are executed (npm test, npm run test:integration, etc.).
- Security and compliance checks (lint, audit, static analysis).

Step 9: Change Communication

- Stakeholders are notified via project channels (e.g., GitHub Discussions, Slack, email).
- Changelog and release notes are updated.
- Documentation (GitHub Wiki, API docs, admin portal notes) is updated.

Step 10: Change Closure

- CR is marked complete and archived.
- Lessons learned are captured for future improvement.

---

5. Change Control Tools & Channels

- **Issue Tracking:** GitHub Issues, JIRA (for enterprise deployments)
- **Version Control:** GitHub, GitLab, Azure DevOps (feature branches, pull requests)
- **Documentation:** Markdown files in `/docs/`, `/api-specs/`, and GitHub Wiki
- **Testing:** Jest, TypeScript, automated test scripts (see `npm test`)
- **Communication:** GitHub Discussions, Slack, email distribution lists
- **Release Management:** Changelogs, semantic versioning, npm registry publishing
- **Templates:** Standardized CR form, impact analysis checklist

---

6. Special Considerations for ADPA

- **Multi-Provider AI Integration:** All changes must be validated across supported providers (OpenAI, Google AI, GitHub Copilot, Ollama, Azure OpenAI) to prevent regression/failover issues.
- **Enterprise Integration:** Changes to Confluence, SharePoint, and Adobe Document Services APIs require credential, authentication, and permission review.
- **Compliance & Security:** All changes must be screened for regulatory compliance (Basel III, MiFID II, GDPR, SOX, PCI DSS) and enterprise security best practices (e.g., OAuth2, Active Directory, SAML).
- **Template & API Versioning:** All updates to framework templates and API endpoints must maintain backward compatibility or include clear migration guidance.
- **Automated Pipelines:** Continuous integration and automated deployment pipelines must validate all approved changes prior to release.
- **Documentation Generation:** All changes affecting document output or structure must trigger regeneration of sample documents and API documentation.

---

7. Change Control Metrics & Reporting

- **Number of CRs submitted, approved, rejected, and implemented**
- **Average time from CR submission to closure**
- **Number of emergency/expedited changes**
- **Impact on key project baselines (scope, schedule, cost, quality)**
- **Post-implementation issue rates (regressions, incidents)**
- **Stakeholder satisfaction with change process**

---

8. Appendix: Change Request Template

Change Request ID:
Submitted By:
Date Submitted:
Description of Change:
Business Justification:
Affected Components:
Impact Analysis (Technical, Schedule, Cost, Quality):
Compliance/Security Considerations:
Urgency/Priority:

Proposed Implementation Plan:
CCB Decision & Rationale:
Implementation Owner:
Verification & Validation Results:
Closure Date:

---

9. References

- PMBOK 7th Edition – Perform Integrated Change Control
- ADPA Project Documentation
- [BABOK v3, DMBOK 2.0, and relevant compliance frameworks]
- [Project Roadmap and Release Plan]

---

10. Version Control

Version	Date	Author	Description
1.0	[Today]	[Your Name]	Initial process definition

End of Document