

Policy Document Review – End-to-End Solution (SharePoint + Power Apps + Power Automate)

Purpose

Design and implementation guide for a policy document lifecycle where each document is reviewed exactly 3 years after creation, routed to multiple departments for structured review using a form, and—when changes are requested—allows uploading a new version to replace the existing document while preserving audit history.

Functional Requirements

- Store policy documents in a SharePoint document library with versioning enabled.
- Automatically detect when a document reaches its 3-year anniversary (based on Created date).
- Initiate a multi-department review at the 3-year mark.
- Capture structured review inputs via a form: decision (Approve/Reject/Needs Revision), comments, reviewer, department, and date.
- If revision is required, allow uploading a replacement document version within the same library item.
- Track per-department status and compute final outcome when all required reviews are complete.
- Provide a dashboard for reviewers/admins to track review progress and outcomes.
- Notify stakeholders (email/Teams) at key milestones (review start, reminders, completion).

Non-Functional Requirements

- Auditable: Maintain history of all reviews, decisions, comments, and versions.
- Security: Restrict who can upload new versions; use AD groups for department reviewers.
- Scalable: Support hundreds to thousands of documents with minimal daily processing cost.
- Maintainable: Parameterize site/library/config via environment variables where possible.

Solution Architecture (High-Level)

- SharePoint: One Policy Library for documents; one Review List to capture structured review records; optional Settings List for configuration (review departments, escalations, SLA days).
- Power Automate: Scheduled flow to detect 3-year anniversaries and spawn review tasks; item-updated flow to aggregate review outcomes; file-modified flow to finalize once a new version is uploaded.
- Power Apps (Canvas): Review portal for departments to submit decisions/comments and (if needed) upload a revised version; admin dashboard to monitor and manage assignments.

SharePoint Data Model

1) Policy Library (Document Library)

Column	Type	Purpose
Title	Single line of text	Human-friendly policy name.
Created	Built-in	Used to compute 3-year anniversary.

ReviewStatus	Choice (Not Due, Due, In Review, Complete)	In Review if <code>ReviewStatus = 'In Review'</code>
FinalStatus	Choice (Pending, Approved, Rejected, Declined)	Final outcome.
Notified3Years	Yes/No (default No)	Gate to prevent duplicate triggers.
Dept1Reviewer	Person/Group	Assigned reviewer for Dept 1 (or use groups).
Dept2Reviewer	Person/Group	Assigned reviewer for Dept 2.
Dept3Reviewer	Person/Group	Assigned reviewer for Dept 3.
Dept1Status	Choice (Pending, Approved, Rejected, Needs Revision)	Needs Revision
Dept2Status	Choice (Pending, Approved, Rejected, Needs Revision)	Needs Revision
Dept3Status	Choice (Pending, Approved, Rejected, Needs Revision)	Needs Revision
Dept1DueDate	Date/Time	Optional SLA due date.
Dept2DueDate	Date/Time	Optional SLA due date.
Dept3DueDate	Date/Time	Optional SLA due date.

Enable major versioning in Library Settings to preserve file history when a new version is uploaded.

2) Review List (Custom List)

Column	Type	Purpose
DocumentID	Number / Lookup to Policy	Relates review to a policy item.
DocumentLink	Hyperlink	Direct link to the file in Policy Library.
Department	Choice (e.g., HR, Legal, Compliance)	Department(s) that must review.
Reviewer	Person/Group	Assigned reviewer (can inherit from Policy Library).
Decision	Choice (Approve, Reject, Needs Revision, Not Applicable)	Needs Revision
Comments	Multiple lines of text	Reviewer notes.
ReviewDate	Date/Time	Submission timestamp.
State	Choice (Pending, Submitted, Completed)	Status of this department's review.

Optional: Settings List to define required departments and SLAs; this enables adding/removing departments without redeploying flows.

Power Automate – Flows (Detailed)

Flow A: Daily Review Trigger

- 1 Trigger: Recurrence (Every 1 day, e.g., 00:05).
- 2 Action: SharePoint – Get files (properties only) from Policy Library (enable pagination).
- 3 Filter: Keep items where `formatDateTime(addYears(Created, 3), 'yyyy-MM-dd')` equals `formatDateTime(utcNow(), 'yyyy-MM-dd')` AND `Notified3Years == false`.
- 4 Update Policy item: `ReviewStatus = 'In Review'`, `Notified3Years = true`.
- 5 For each required department (from columns or Settings List), create a Review List item with Department, Reviewer, DocumentID, DocumentLink, State='Pending'.
- 6 Notify each department via email/Teams with a link to the Power Apps form to submit review.

Expressions (Flow A)

- `formatDateTime(addYears(item()?'Created'], 3), 'yyyy-MM-dd')`
- `formatDateTime(utcNow(), 'yyyy-MM-dd')`

Flow B: Review Submission Aggregator

- 1 Trigger: When an item is created or modified (SharePoint – Review List).
- 2 Get related Policy Library item by DocumentID.
- 3 Compute department tallies: if all required departments have State='Submitted', then evaluate decisions.
- 4 If all Approved → set Policy.FinalStatus='Approved'; set ReviewStatus='Completed'; notify stakeholders.
- 5 If any Needs Revision or Rejected → set Policy.FinalStatus='Needs Revision'; notify the responsible owner to upload a new version.
- 6 Optionally set due date reminders/escalations if reviews remain Pending beyond SLA.

Flow C: New Version Uploaded (Finalize)

- 1 Trigger: When a file is created or modified (properties only) in Policy Library.
- 2 Condition: if ReviewStatus='In Review' AND FinalStatus='Needs Revision', then proceed.
- 3 Action: Mark ReviewStatus='Completed'; log 'Revised on ' to Review List (new history entry).
- 4 Notify stakeholders with version link; optionally reset Notified3Years=false and set a new BaselineDate for next review cycle if your policy re-starts the 3-year clock on revision.

Power Apps – Review Portal (Canvas App)

- Data Sources: Review List (primary), Policy Library (lookup), Settings List (optional).
- Home Screen: Filters for My Pending Reviews, In Review, Overdue; search by Title/Department.
- Review Screen: Displays Policy metadata, link to document, form fields (Decision, Comments), Submit button updates Review List item (State='Submitted', ReviewDate=Now()).
- Revision Screen (conditional when Needs Revision): Attach/Upload control bound to the corresponding Policy item to upload a new file version (SharePoint handles versioning).
- Admin Screen: Grid of all policies in review; per-department status; reassign reviewer; manual override (if permitted).

Key Form Logic (Power Apps)

- Submit review: Patch(ReviewList, ThisItem, { Decision: ddlDecision.Selected.Value, Comments: txtComments.Text, State: "Submitted", ReviewDate: Now() })
- Upload revised version: Use the built-in Attachment/File control connected to the Policy Library item form; OnSuccess triggers a flow or sets a flag if needed.

Permissions & Governance

- Use AD/Entra groups per department (e.g., Policy-Reviewers-HR) to assign Reviewer fields and secure the Review List.
- Restrict upload of new versions to policy owners/compliance team; reviewers can submit forms but not replace files unless allowed.

- Enable versioning and (optionally) content approval in Policy Library.
- Use retention labels/policies as required by compliance.

Deployment, Configuration & Environments

- Replicate SharePoint schema (columns, lists) across Dev/Test/Prod. Consider a provisioning script (PnP PowerShell).
- Parameterize site URLs and list/library names via environment variables for Power Automate and Power Apps where applicable (standalone apps).
- For list-integrated forms, recreate integration in each environment and copy app screens/logic.

Testing Strategy

- Unit test flow expressions (date calculations, filters).
- Integration test: seed documents with Created dates > 3 years to trigger immediately.
- User acceptance test: reviewers submit via app; upload revised version; verify version history and status updates.
- Performance test: validate daily run with realistic document counts and pagination.

Appendix – Sample Emails/Templates

Review Start

Subject: Policy review due – \${Title}
 Body: \${Title} has reached 3 years and requires review from \${Departments}. Please submit your decision and comments here: \${PowerAppsLink}

Revision Requested

Subject: Revision required – \${Title}
 Body: One or more departments requested changes. Upload a new version here: \${ItemLink}

Review Completed

Subject: Review completed – \${Title}
 Body: Final status: \${FinalStatus}. Version: \${VersionLabel}. View details: \${ItemLink}

Appendix – Key Power Automate Expressions

- Exact 3-year anniversary: `formatDateTime(addYears(item()?'Created', 3), 'yyyy-MM-dd') = formatDateTime(utcNow(), 'yyyy-MM-dd')`
- 3+ years or older (use with a Notified gate): `lessOrEquals(addYears(item()?'Created', 3), utcNow())`
- Prevent duplicates: boolean column `Notified3Years` flipped to true after first trigger

Appendix – Suggested Columns JSON (PnP/Provisioning placeholder)

Use your provisioning method of choice (PnP PowerShell/CLI for Microsoft 365) to provision lists and columns consistently across environments.