

Document Automation Requirements and Test Pack

Define one document family, preserve source evidence, design human control, test safe writes, and choose the minimum maintainable implementation.

Course: <https://flowgrammer.ca/courses/document-processing-automation-course>

Complete one section after each lesson. Cedar & Quay examples are fictional. Replace them with fictional, redacted, or approved documents. Label every unrun proof gate Not tested.

1. System boundary

Define one family, intake, reviewer, controlled destination, exclusions, and first decision.

Boundary pattern: We receive [family] through [channel], prepare [result] for [reviewer], and create [non-final destination] only after [decision].

Boundary sentence

Actor or system	Current responsibility	Stop reason	Owner

Supported	Excluded	Forbidden	Reason

Go, repair, narrow, keep manual, or stop decision

2. Schema and provenance

Own the normalized record and preserve original values, source pointers, versions, and unknowns.

Field and type	Required?	Source pointer	Original / normalized	Unknown rule

Identity or evidence	Formula or source	Used by	Retention owner

[] Source hash [] Business key [] Version [] Adapter version [] Correction history [] Destination receipt

3. Validation and exception controls

Separate file checks, extraction evidence, deterministic rules, confidence policy, and duplicates.

Rule or field	Signal / condition	Continue / review / stop	Owner	Evidence

Exception code	Status	Blocks write?	Allowed action	Authority

Leave confidence thresholds blank until calibrated on representative documents and field consequences.

4. Human review queue

Give named reviewers source context, authority, version locks, correction history, and escalation.

Queue field or view	Purpose	Required evidence	Owner

From	Decision or event	To	Actor	Version rule

Timeout, absence, unauthorized actor, stale version, and escalation policy

5. Destination and recovery contract

Define the allowed draft, idempotency, retries, receipt, reconciliation, and forbidden actions.

Destination item	Required contract	Evidence	Owner

Failure	Known state	Allowed retry	Reconciliation	Stop rule

Stable idempotency key and destination lookup

[] Draft only [] Payment blocked [] Bank change blocked [] Same retry key [] Receipt stored [] Attempts != writes

6. Representative and adversarial tests

Write expected fields, control states, review evidence, attempts, and writes before execution.

Case and input	Expected extract / exception	Review / authority	Attempts / writes	Pass / fail / not tested

☐ Clean ☐ Duplicate ☐ Missing ☐ Conflict ☐ Timeout ☐ Unknown type ☐ Stale ☐ Unauthorized ☐ Replay ☐ Adversarial

Held-back cases, gaps, repairs, reruns, and regression set

7. Platform and implementation plan

Map layers to evidence, gaps, owners, proof stages, monitoring, rollback, and change control.

Layer / requirement	Option	Evidence	Gap / disqualifier	Owner

Proof stage	Entry evidence	Pass condition	Rollback / stop

Proceed, test more, repair, narrow, keep manual, or stop decision

8. Capstone critical gates

Pass every safety gate before dimensional scoring.

Critical gate	Evidence	Pass / fail / not tested

Critical gates: bounded family; explicit unknowns; no invalid approval write; duplicate stop; version lock; non-final destination only; idempotent ambiguous retry; proportional evidence claims.

Any failed or untested critical gate means the capstone is not yet acceptable.

9. Capstone dimensional scoring

Grade completeness without averaging away a safety failure.

Dimension	0 missing	1 weak	2 partial	3 usable	4 decision-ready	Score

Acceptance: all critical gates pass and every dimension earns at least 3. This supports a next decision, not production deployment.

Final decision, approvers, conditions, unresolved gaps, and next review date