



Retirement Gate – checklist

Letting dead knowledge go is a governance act, not housekeeping. Stale articles are the raw material of confident wrong answers — and deletion (as opposed to archival) destroys the audit trail and breaks every citation ever made. Target: ≤ 3 minutes.

1 – Is it actually dead?

- ☐ A real retirement trigger applies:
- **Fixed**: known-error's fix shipped (`fixed_in` set) *and* adoption has crossed your threshold (e.g. 80% of active installs), or
- **Superseded**: a newer article covers the current truth (id at hand), or
- **EOL**: the product/version/plan it applies to is no longer supported, or
- **Demand died**: sustained near-zero usage *and* no open gap signals for it.
- ☐ Not merely *stale* — an article that's wrong-but-needed goes to the **update gate**, not here. Retirement is for knowledge whose question no longer exists.

2 – Before archiving

- ☐ Successor linked (`links: KO-xxxxx # superseded-by`) if one exists.
- ☐ Inbound links checked — articles pointing here get repointed to the successor or annotated.
- ☐ The article's `owner` signs off. (No owner? That's an orphan — fix ownership per Module 5 *first*, then retire properly.)

3 – Archive – never delete

- ☐ `status: archived` — the id, content, and history remain intact forever.
- ☐ Confirm the delivery layer excludes archived content from answers *and* that citations to the id still resolve (to the archived page or its successor).
- ☐ Retirement is itself a versioned change: bump `version`, decision log row (date, actor, id, trigger, successor).

4 – Close the loop

- ☐ If retirement leaves a coverage hole (EOL'd article but the question lives on in a newer version) — file a capture request now, while the context is in front of you.