12. Change and release policy

> **Template.** Part of the open-source AI App Policy Kit, version 1.0. Replace everything in [square brackets], delete what does not apply, and have the person who runs the business approve it. A starting point, not legal advice.

Business	[Business name]
Document owner	[Name and role]
Approved by, and date	[Name], [date]
Next review	[Date, at most 12 months on]

Why this matters

Most outages are caused by a change. With AI tools a change takes minutes to make, so more of them reach customers, faster, with less thought. A few fixed steps between "it works on my screen" and "customers are using it" catch most of the damage.

Who it covers

Every change to [Business name]'s live apps: code, database structure, cloud settings, AI prompts and the choice of AI model.

The rules

1. **Everything is in version control.** Code, database changes, cloud set-up, and the prompts and settings that steer an AI model. If it can change what customers see, it has a history.
2. **Nobody edits the live system by hand.** Changes reach production through the release process below, not by typing into the live database or the hosting console. An emergency fix made by hand is written back into version control the same day.
3. **A test copy exists.** Changes run first in a test environment that is separate from production and holds no real customer data.
4. **Every change is reviewed** by a second person (document 05) and passes the automated checks.
5. **Releases are automatic and repeatable.** A pipeline builds and deploys. The same steps run every time, and it uses its own narrow credentials.
6. **You can go back.** Before each release you know how to return to the previous version, and you have done it at least once. Database changes are written so the old version of the app still works.
7. **Prompts and models are changes too.** A new system prompt, a new model or a new model version is tested against a fixed set of example questions before release, and the results are kept.
8. **Keep software up to date.** Security updates for the app's dependencies and platforms are applied within [14] days, sooner if actively exploited. An automated tool proposes them.
9. **Choose the moment.** No releases when nobody will be around to watch. Check the app and its alerts for [30] minutes after each one.

10. **Record each release:** what, who, when, and what happened.

Release steps

1. The change is written on a branch and proposed for review.
2. Automated checks pass: tests, linter, dependency scan, secret scan.
3. A second person reviews and approves.
4. It is deployed to the test environment and checked there.
5. [Name or role] approves the release to production.
6. The pipeline deploys. The releaser watches errors, alerts and one real journey through the app.
7. If something is wrong, roll back first and investigate after.

Release log

Your code host or pipeline may keep this for you. If so, say where: [link].

Date	What changed	Released by	Approved by	Result	Rolled back?
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Checklist

- ☐ The main branch is protected: no direct changes, review required, checks required.
- ☐ Nobody deploys from their own laptop.
- ☐ A rollback was practised in the last 6 months.
- ☐ Prompts and model versions are in version control, with a test set.
- ☐ No dependency has a known critical vulnerability older than [14] days.

References

- NCSC, *Secure development and deployment guidance*.
- Cyber Essentials, *Security update management*.

AI App Policy Kit, version 1.0. © 2026 PugliseWeb LTD, licensed under CC BY 4.0: use it, change it and share it, with credit. The whole kit, and a free Software Health Check of your app against it: https://www.puglieseweb.com/ai-policy-kit