

The release readiness checklist

31 checks to run before you ship. Tick what you already do. The gaps are where your risk sits.

1. Know what's in the release

- ☐ Every change in the release is listed and linked to its ticket or pull request.
- ☐ Nothing has slipped in that wasn't planned, reviewed and tested.
- ☐ Database migrations, configuration changes and new feature flags are listed alongside the code changes.

2. Understand the risk

- ☐ The riskiest changes are named: money, data, security, the busiest user journeys.
- ☐ The riskiest changes got the deepest testing, not the same depth as everything else.
- ☐ Anything not tested, and the reason, is written down.

3. Testing you actually believe

- ☐ Automated checks pass, and none covering the risky areas are muted, skipped or quarantined.
- ☐ Someone has explored the risky changes by hand, beyond the scripted cases.
- ☐ The critical user journeys have been checked end to end.

4. Defects reviewed

- ☐ Every open defect touching this release has been looked at by someone who can decide.
- ☐ No serious defect ships unless someone has accepted it by name, with a plan.
- ☐ Known issues that customers might notice are ready for support and release notes.

5. Environments and data

- ☐ Testing happened in an environment close enough to production to trust the results.
- ☐ Migrations have been rehearsed on realistic data, including how long they take.
- ☐ Production configuration and feature flag settings have been checked for this release.

6. The things that aren't features

- ☐ Performance has been checked for anything that changes load, queries or payload sizes.
- ☐ Security-relevant changes, such as authentication, permissions and new inputs, have had a second look.
- ☐ Interface changes have had a basic accessibility check: keyboard, contrast, screen-reader labels.

7. Release mechanics

- ☐ The deployment steps are written down and someone other than the author could follow them.
- ☐ Rollback or roll-forward has been tried, not just assumed.
- ☐ Risky changes sit behind a flag or switch that can turn them off without a redeploy.
- ☐ The timing avoids peak trading and the end of the working week.

8. Know what broken looks like

- ☐ Dashboards and alerts cover the changed areas, not only the system as a whole.
- ☐ The team knows which signals would mean the release has gone wrong.
- ☐ Someone is named to watch those signals for the first hours after release.

9. People and communication

- ☐ Whoever is on call knows what is going out and what to look for.
- ☐ Support has the release notes and the known issues before customers do.
- ☐ Any customer-facing change has its message ready.

10. Make the decision, and write it down

- ☐ Three lines recorded: what was tested and found, what wasn't tested and why, and who accepted the remaining risk.
- ☐ The person accepting the risk has the authority to do so.
- ☐ The decision is go, no-go, or go with named conditions, never "probably fine".