PIERRE BOCKNER · EMBEDDED SYSTEMS & EDGE AI

Yocto License Risk Checklist

A practical self-assessment for embedded Linux product teams · bockner.com

> Sixteen checks across four risk areas. Work through them against your *actual release image*, not your intentions. Any unchecked item in sections A or B is a potential release blocker; sections C and D expose maintainability and procurement risk. Version 2026-11.

A · License inventory & traceability

- ☐ **Recorded licenses.** Every package in the shipping image has a license recorded as an SPDX identifier — not free text, not “unknown”.
- ☐ **Image-based SBOM.** An SBOM (SPDX or CycloneDX) is generated from the actual release image, not guessed from build-time dependencies.
- ☐ **Verified recipe metadata.** Recipe LICENSE flags are verified against upstream sources; generic fallback licenses are not used as a shortcut.
- ☐ **Ship vs. build separation.** Build-only tools (native/nativesdk packages) are clearly separated from target packages that actually ship to customers.

B · Copyleft & GPLv3 exposure

- ☐ **Known GPLv3 footprint.** You know exactly which GPLv3/AGPL components land in the shipping image — e.g. via the license manifest of a release build.
- ☐ **Documented boundaries.** Copyleft boundaries are documented: which code is proprietary, which is derived from copyleft code, and how they are separated.
- ☐ **Compatibility checked.** License compatibility is checked for combined and linked works (e.g. GPL vs. Apache-2.0, advertising clauses, exception clauses).
- ☐ **Quarantine plan.** A replacement or quarantine plan exists for components that legal or procurement rejects — before design-in, not after.

C · Build reproducibility

- ☐ **Version pinning.** All layers and release-relevant recipes are version-pinned; no floating AUTOREV in release builds.
- ☐ **Rebuild verified.** A release image can be rebuilt months later with bit-for-bit equality or only documented, controlled differences.
- ☐ **Archived inputs.** Shared-state cache and source downloads are archived per release; dead upstream URLs are not a single point of failure.
- ☐ **Clean CI build.** CI builds the release from a clean checkout without manual steps or developer-machine state.

D · Maintenance & procurement

- ☐ **LTS strategy.** Kernel and BSP long-term maintenance covers the intended field lifetime (e.g. 10 years), including security backports.
- ☐ **Compliance evidence on demand.** SBOM, license notices and source offers can be delivered to customers within days — not weeks of research.

- [ ] **Procurement sign-off.** Procurement has approved the open-source components before design-in, with a record you can point to.
- [ ] **License monitoring owner.** A named owner watches upstream license changes and new versions for the components you ship.

> **How to read your result:** unchecked A/B items should block the next release until resolved. Unchecked C/D items are debt that compounds with every year in the field — schedule them, don't ignore them.

Want this checked on your actual build system? Start with a free, confidential 15–20 minute private technical review: bockner.com/review. Fixed-scope follow-up: Yocto Build & License Risk Scan — bockner.com/yocto-risk-scan.

Pierre Bockner · Senior Embedded Software Engineer · Nürnberg, Germany · bockner.com · info.bockner@posteo.de · This checklist is general guidance, not legal advice.