6.9 Tool Baseline Checklist Instances

Go To Top

Return to Prerequisites and Tooling

Each phase, build, release candidate, test run, demonstration Baseline, or deployment Baseline should create a separate checklist instance from the static checklist. The instance records the actual tool selections, approved products, version ranges, observed versions, verification results, exceptions, and Evidence for that specific Baseline.

A checklist instance should live outside the stable handbook so that normal phase-to-phase tool changes do not require handbook edits. The Repository may store checklist instances in a controlled location such as:

```
docs/checklists/
    phase0-tool-baseline-checklist.md
    phase0-demo-tool-baseline-checklist.md
    release-candidate-001-tool-baseline-checklist.md
```

Each checklist instance should identify:

1. Baseline, build, test run, release candidate, demonstration, or deployment environment it applies to.
2. Date the checklist was completed.
3. Person or role completing the checklist.
4. Approved product, platform, or implementation for each category.
5. Product status for each entry.
6. Acceptable version range.
7. Observed version.
8. Applicable role or environment.
9. Verification command or method used.
10. Verification result.
11. Links or references to supporting Evidence.
12. Any approved exceptions or deviations.
13. Any open risks or unresolved tool decisions.
14. Any related implementation decision record.

The team should update a checklist instance when the corresponding Baseline changes. The team should update the handbook only when the checklist structure, required categories, role classifications, product-status definitions, version-range conventions, or completion rules change.

The team may represent checklist instances as Markdown tables, CSV files, spreadsheets, YAML files, or another controlled format. Markdown tables work well for reviewable documentation. CSV or spreadsheet formats work well when the team wants sorting, filtering, role-based views, or later automation. YAML or JSON formats may work well when the team wants scripts to generate setup commands, validation checks, or CI runner definitions.

The recommended approach is to keep the explanatory rules in Section 6, place the reusable checklist template in an appendix or annex, and store completed checklist instances under

`docs/checklists/` or another controlled Repository location. This separation keeps the handbook stable while allowing each phase or build to record its actual implementation environment.
