

Oracle Fusion Quarterly Update Configuration Checklist

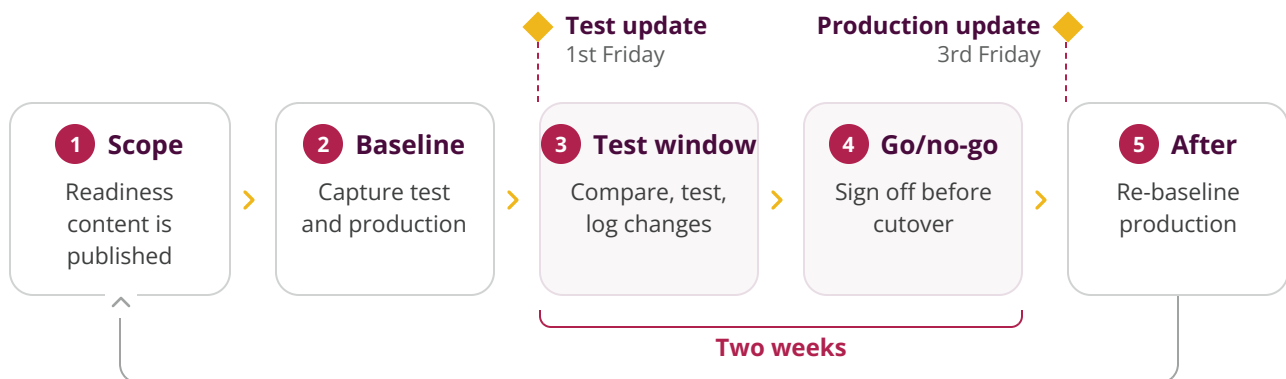
How to use this checklist

Run this checklist for every Oracle Fusion quarterly update, starting the day Oracle publishes the release readiness content. It covers the configuration side of the update: what your environments contain, how they differ, and what changed. Functional testing runs alongside it.

 Applies to	Every quarterly update (A, B, C and D), all three cohorts.
 Timing	Test environments update on the first Friday of the update month, and production follows two weeks later on the third Friday. Your cohort sets the month, so confirm it in the Oracle Cloud Console every quarter.
 Release-specific details	Feature counts, deprecations and new privileges change every quarter. Pull them from Oracle's Feature Listing for the current update.
 Output	A baseline, a change log and a go/no-go record that become your SOX ITGC evidence and the starting point for the next update.

At a glance

Every quarter runs five phases around two update Fridays



Between updates: monthly drift checks, then the next quarter

Phases 1 and 2 happen before the test update, phases 3 and 4 fill the two weeks before production, and phase 5 sets the baseline for next quarter.



PHASE 1

Scope the update

Start when Oracle publishes the readiness content, at least 30 days before Cohort A test environments update.

FROM FEATURE LISTING TO SCOPE



Feature Listing



Your modules



Flag what matters



Enable or defer

FLAG

• Large-scale impact

• Deprecations

• Security Adjustment

• Opt-in windows ending

- ☐ Confirm your cohort and both update dates in the Oracle Cloud Console
- ☐ Map the two-week window against your own calendar: financial close, change freezes, holidays, key staff absences
- ☐ Pull the Feature Listing for the update and filter it to the modules you run
- ☐ Flag features marked large-scale impact to existing processes
- ☐ Flag deprecations and replacements, including AI agents, BI view objects and integrations
- ☐ Flag features whose Action to Enable is Security Adjustment, or that add new roles or privileges
- ☐ List opt-in features from earlier updates whose opt-in window ends with this update
- ☐ Decide which new features you will enable this quarter, and which you will defer

Phase 1 checked ☐ of 8

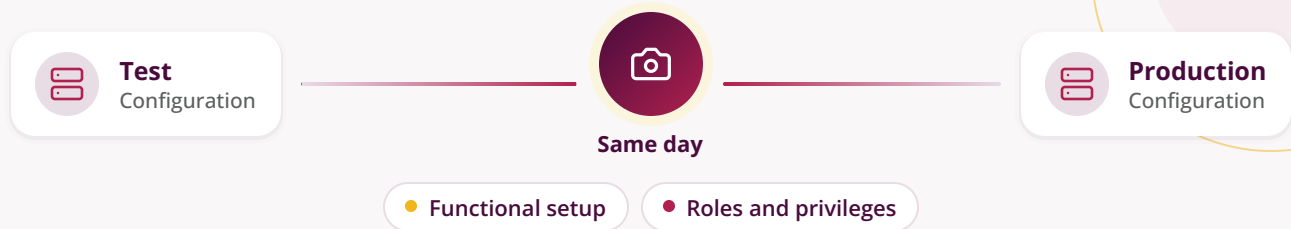


PHASE 2

Baseline and inventory

Finish before your test environment updates. Without a baseline, you cannot separate what the update changed from what was already different.

ONE SNAPSHOT, BOTH ENVIRONMENTS



- ☐ Capture test and production configuration on the same day, so you compare like with like
- ☐ Include functional setup for every module in scope, including profile options and offerings, plus Redwood page and Page Composer personalizations
- ☐ Include security: job, duty, abstract and data roles, privileges, and security profiles
- ☐ Agree who owns AI agent changes and how they are reviewed before production
- ☐ Compare test and production now and resolve or log existing drift, so it is not mistaken for an update change later
- ☐ Inventory extensions, integrations and extracts that depend on anything being deprecated
- ☐ Build a feature enablement map: each feature you will enable, its setup type, and any new privileges it needs
- ☐ Name an owner for each area

Phase 2 checked ☐ of 8



PHASE 3

The two-week testing window

Run configuration checks alongside functional testing. Most teams run functional testing well; the configuration side is where gaps appear.

THE TWO-WEEK WINDOW

Test update · 1st Friday

Production update · 3rd Friday



● Compare to baseline

● Configure and record

● Test with real roles

● Log every difference

- ☐ On the day test updates, compare test against your pre-update baseline
- ☐ Compare test against production for every area in your baseline
- ☐ Log every difference: what changed, why, who approved it, and whether it moves to production
- ☐ Configure the features on your enablement map in test, and record each setup change
- ☐ Test privilege-gated features with real role assignments, not administrator accounts
- ☐ Validate AI agent behavior in test before production, using Oracle's own testing tools
- ☐ Confirm extracts and integrations that use deprecated objects still run, or are updated
- ☐ Review Redwood personalizations, extensions and integrations on any pages that changed
- ☐ Keep comparison reports and test evidence with the change record for SOX ITGC

Phase 3 checked of 9



PHASE 4

Go/no-go gate

Hold this review before production cutover. Any unchecked item is a no-go unless its owner signs off on a documented exception.

THE GATE BEFORE CUTOVER



- ☐ Test matches the pre-update baseline, except for documented update changes
- ☐ Every difference between test and production is logged and approved
- ☐ Setup changes for enabled features are documented and ready to promote
- ☐ Roles and privileges for enabled features are tested with real role assignments
- ☐ AI agent changes have an agreed owner and have been reviewed
- ☐ Dependencies on deprecated objects are updated or have a migration plan
- ☐ Change record and comparison reports are filed as SOX ITGC evidence
- ☐ A fallback plan is agreed for production cutover
- ☐ Cutover timing is checked against close and change freezes
- ☐ Functional, security and DevOps leads have signed off

Phase 4 checked of 10

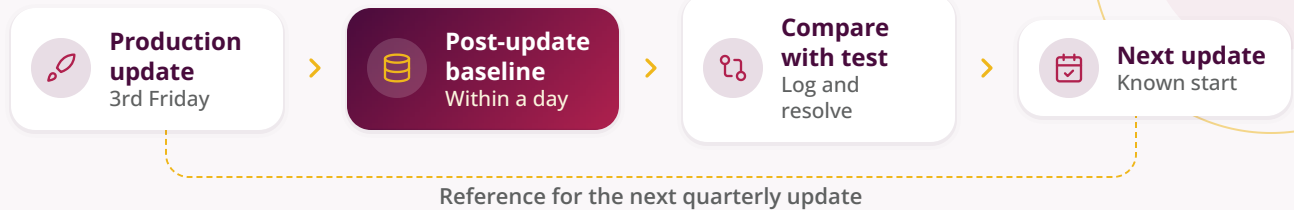


PHASE 5

After production

Confirm that what you validated is what landed, then reset your baseline for the next update.

THE BASELINE CARRIES FORWARD



- ☐ Capture the post-update production baseline within a day of the update
- ☐ Compare production against test, and log and resolve any difference
- ☐ Promote any remaining approved setup changes and record them
- ☐ File the final comparison with the change record as SOX ITGC evidence
- ☐ Close out the feature enablement map and carry open items into next quarter's plan
- ☐ Note what slowed you down this cycle and fix it before the next one
- ☐ Keep the post-update baseline as the reference for the next quarterly update

Phase 5 checked ☐ of 7

ONGOING



Between updates: keep drift from building up

Drift builds up between updates too. A short monthly routine keeps each quarter starting from a known state.



- ☐ Compare test and production monthly, not only before an update
- ☐ Route every setup change through the same approval and promotion process as code
- ☐ Restrict who can change setup directly in production, and review that access quarterly
- ☐ Track changes to roles, privileges and security profiles as closely as functional setup
- ☐ Refresh lower environments from production on a set schedule, and re-baseline after each refresh
- ☐ Review the change log monthly for undocumented or emergency changes

Monthly routine checked ☐ of 6



See it in action

Flexagon Configuration Management and Flexagon DevOps handle the configuration work in this checklist for Oracle Fusion Cloud and EBS: baselines, environment comparison, change tracking, promotion and audit evidence.

Request a demo to see how it works for your next quarterly update.

REQUEST A DEMO