COMPLIANCE ISN'T A MILESTONE

A practical guide to managing code compliance throughout design.

E ESTABLISH → V VALIDATE → R REVALIDATE ↻ REPEATS → D DOCUMENT

EVERY COMPLIANCE AREA, TRACED THROUGH ALL SIX PHASES

Compliance checklists are often organized by phase: at schematic design, confirm occupancy; at design development, check accessibility. That structure is convenient, and it is also misleading. It suggests that a topic is **handled once and is then complete**.

In practice, requirements are established early, tested as the design takes shape, and tested again every time something they depend on changes. This guide organizes compliance around that cycle. Each compliance area is traced across six phases, with the kind of work that typically dominates at each point, a worked example you can check, and a self-assessment to run with your team.

READ THIS FIRST

- Timing and applicability vary by project type, jurisdiction, delivery method, and how the design develops. Fast-track, design-build, and phased projects compress or overlap these phases.
- Code references use the ICC model codes (I-Codes) as a common baseline and indicate where requirements usually live. Your project answers to the editions and local amendments your jurisdiction has adopted, and some jurisdictions and building types are governed by other codes and standards entirely.
- This is not legal advice or code advice. Compliance determinations belong to the licensed professional of record and, ultimately, to the authority having jurisdiction.

CONTENTS

How to use this guide 02

Worked example: a room that changes use 03

The compliance lifecycle 04

The matrix: eight areas across six phases 05

Compliance changes as the design changes 06

Where the gap opens, and how to close it 07

Self-assessment 08

The exercise 09

Where Kestrel fits 10

[02 · WORKED EXAMPLE]

A ROOM THAT CHANGES USE

The cost of a code problem is measured in the work built on top of it, not in days. One room, two phases, and math an architect can check.

Occupant load factors and thresholds reflect recent IBC editions. Confirm against the adopted code and amendments.

[01] SCHEMATIC DESIGN

A 900 sq ft training room in an office building is planned as part of the business occupancy. Calculated at the classroom factor of 20 sq ft per occupant, its occupant load is 45, below 50. Assuming the applicable common-path limits and other egress conditions are satisfied, a single exit access doorway can serve the room, and the door may swing into the room.

```
900 sq ft ÷ 20 sq ft per occupant = 45
```

[02] DESIGN DEVELOPMENT

The owner asks for the room to host events. At the factor for tables and chairs, 15 sq ft per occupant, its occupant load is now 60. At 900 square feet and an occupant load of 50 or more, the room no longer qualifies for the IBC's small accessory assembly exceptions and would generally be classified as Group A.

```
900 sq ft ÷ 15 sq ft per occupant = 60
```

[03] WHAT NOW DEPENDS ON IT

- A second exit or exit access doorway from the room, with the required separation where applicable.
- Egress doors serving the room swinging in the direction of egress travel, with panic or fire exit hardware where required.
- Corridor egress capacity and plumbing fixture counts recalculated with the new load.
- A decision on how the Group A space relates to the rest of the building: an accessory occupancy if the area limits allow it, separated occupancies, or a nonseparated analysis applying the more restrictive requirements.

THE POINT

None of this is hard to resolve the week the owner asks. It is harder once the adjacent rooms, the corridor, the door schedule, the plumbing layout, and the consultants' drawings have developed around a single-exit room. The code question did not change. **The amount of work depending on the answer did.**

ESTABLISH, VALIDATE, REVALIDATE, DOCUMENT

Every compliance area moves through the same four kinds of work. The order is not strictly linear: revalidation recurs for as long as the design is changing, and documentation is most useful when it records reasoning as it happens.

E [01]

ESTABLISH

Identify which requirements apply and set the working assumptions: classification, code editions, compliance paths, the numbers the design will be built around.

V [02]

VALIDATE

Test those assumptions against the design as it actually exists: the plan, the massing, the assemblies, the model.

R [03]

REVALIDATE

Check again whenever the design, the program, the team, or the governing requirements change. This is the stage that repeats.

↻ REPEATS WHENEVER THE DESIGN CHANGES

D [04]

DOCUMENT

Record the requirement, the decision, and the reasoning, so the project team, the reviewer, and the next project can follow it.

A REQUIREMENT IS NEVER SIMPLY “DONE.” IT IS CURRENT, OR IT HAS NOT BEEN CHECKED SINCE THE LAST RELEVANT CHANGE.

EIGHT COMPLIANCE AREAS ACROSS SIX PHASES

Each mark identifies the kind of work that often dominates at that point, not a required sequence. This page shows the marks and a short note for each cell. The full matrix, with the code basis for each area and what each stage looks like, is in the online edition.

LEGEND [E] ESTABLISH [V] VALIDATE [R] REVALIDATE [D] DOCUMENT

FULL MATRIX: KESTRELLABS.COM/MILESTONE

COMPLIANCE AREA	[01] CONCEPT / PRE-DESIGN	[02] SCHEMATIC DESIGN	[03] DESIGN DEVELOPMENT	[04] CONSTRUCTION DOCUMENTS	[05] PERMIT / REVIEW	[06] CONSTRUCTION ADMINISTRATION
[01] **OCCUPANCY AND USE**	E Classify uses; flag mixed occupancy	V Test classification against the plan	R Recheck when the program moves	D Code plans and occupant loads agree	R Resolve reviewer questions on use	R Tenant and program changes after permit
[02] **CONSTRUCTION TYPE AND BUILDING SIZE**	E Test allowable height, stories, area	V Confirm massing against allowable area	R Recheck area, stories, framing, assemblies	D Area calcs and ratings in the analysis	R Resolve comments on area and increases	R Substitutions checked against type
[03] **EGRESS AND LIFE SAFETY**	E Estimate loads; set exit and core assumptions	V Test exit count, separation, travel distance	R Doors, corridors, dead ends, capacity	D Life-safety plans and exit capacity	R Trace each comment to every sheet	R Field changes to partitions, doors, hardware
[04] **ACCESSIBILITY**	E Identify the standards that apply	V Routes, entrances, units, spaces	V Clearances as rooms are detailed	D Enlarged plans, heights, details	R Comments carried into plans and details	V Verify field dimensions and substitutions
[05] **FIRE RESISTANCE AND SEPARATIONS**	E Separations implied by occupancy and type	V Locate walls, barriers, shafts, openings	V Select assemblies; check continuity	D Rated plans, openings, penetration details	R Resolve rating and continuity comments	V Door, damper, glazing, firestop submittals
[06] **ENERGY**	E Code, climate zone, compliance path	V Test envelope and glazing assumptions	R Assemblies, glazing, systems rechecked	D Envelope values and compliance forms	R Resolve across arch, mech, elec	V Substitutions and commissioning
[07] **JURISDICTION-SPECIFIC REQUIREMENTS AND AMENDMENTS**	E Confirm editions, amendments, state codes	V Test unusual conditions with the AHJ	R Monitor adoption and effective dates	D Cite editions; record interpretations	R Record reviewer interpretations	D Inspector interpretations and alternates
[08] **DOCUMENTATION AND COORDINATION**	E Decide where assumptions live and who owns them	D Record assumptions with their reasoning	V Coordinate consultants to the assumptions	D Code sheets, plans, specs agree	R Trace every comment to the sheets	D Record RFIs and field changes with rationale

COMPLIANCE CHANGES AS THE DESIGN CHANGES

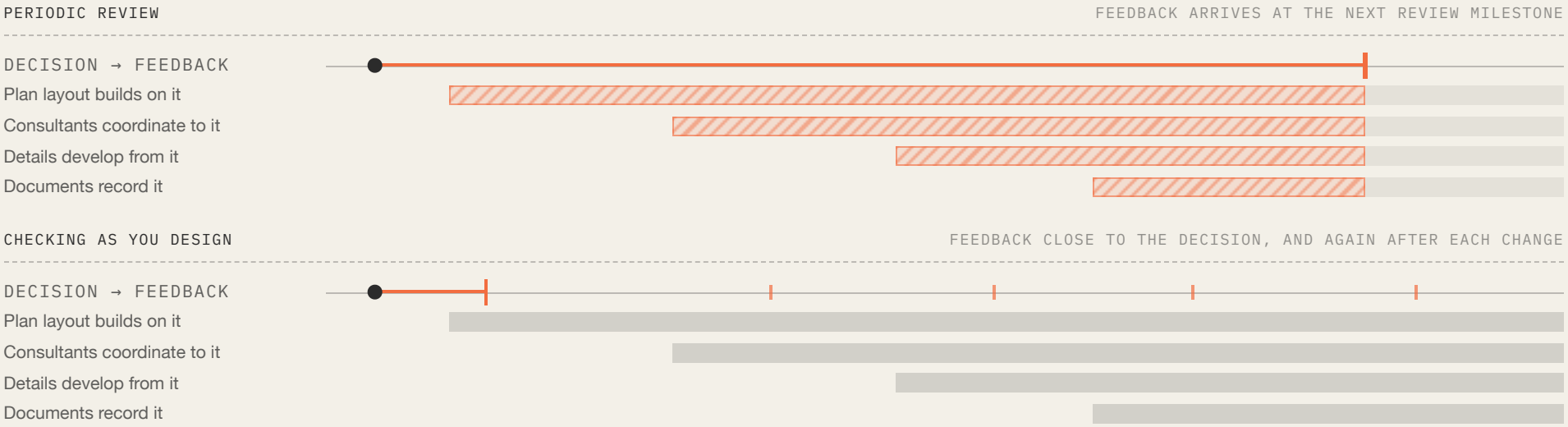
Compliance is not simply something that happens at the end of a phase. Early assumptions about occupancy, egress, accessibility, fire separation, and other requirements influence downstream design decisions.

When compliance feedback arrives significantly after the decision that created the issue, other work may already depend on that decision. The plan has been laid out around it, consultants have coordinated to it, details have been developed from it, and the documents describe it.

The opportunity is therefore not only to make review faster. It is to reduce the distance between a design decision and the compliance feedback on it. That distance is **measured in dependent work, not only in days**, which is why the same issue can be a quick correction in one week and a rework problem a month later.

REDUCE THE DISTANCE

DESIGN DECISION → COMPLIANCE FEEDBACK



DESIGN DECISION · COMPLIANCE FEEDBACK · WORK BUILT ON AN UNCHECKED DECISION · WORK BUILT ON A CHECKED DECISION · ILLUSTRATIVE, NOT TO SCALE

FIVE PLACES FEEDBACK FALLS BEHIND THE DESIGN

The distance between decision and feedback rarely comes from carelessness. It comes from how compliance work is structured: **who holds the assumptions, what gets checked, and when**. Each one can be closed without new software.

[01]

ASSUMPTIONS WITHOUT AN OWNER

An early classification or occupant load is set in a meeting and never written down with its reasoning, so no one knows to recheck it when the program moves.

HOW TO CLOSE IT

Open the code analysis as a working document on day one. Every assumption gets a reason, a date, and a name: the person who rechecks it when the program moves.

[02]

CHECKING AN OUTPUT INSTEAD OF THE DESIGN

Review happens on an exported set while the model keeps moving. By the time comments come back, they describe a version of the building that no longer exists.

HOW TO CLOSE IT

Check the model, not the export. When a review has to run on a printed set, date it, and recheck the affected areas against the model before the comments are closed.

[03]

CHANGES THAT ARRIVE FROM OUTSIDE

Consultant coordination, value engineering, and owner requests change the design without anyone asking which code assumptions they touch.

HOW TO CLOSE IT

Add one question to every coordination meeting, VE round, and owner request: which code assumptions does this touch? Route the answer to the person who owns them.

[04]

KNOWLEDGE HELD BY A FEW PEOPLE

The firm knows how a jurisdiction reads a provision, but only because one senior architect remembers. Everyone else finds out at review.

HOW TO CLOSE IT

Keep a firm-level record of how each jurisdiction has read a provision, updated from every review response, where the next project team can find it.

[05]

FINDINGS WITHOUT A TRAIL

A comment is resolved on one sheet and never traced to the other sheets, schedules, and model elements built on the same decision.

HOW TO CLOSE IT

Before a comment is closed, list the sheets, schedules, and model elements built on the same decision, and check each one.

HOW FAR DOES FEEDBACK TRAIL THE DECISION IN YOUR PRACTICE?

Ten prompts for a project team or a practice leadership meeting. There is no score. The statements you can't confidently check are where the **distance between decision and feedback** is likely longest.

PROJECT _____ TEAM _____ DATE _____

- ☐ 01 We identify applicable requirements early in design.
- ☐ 02 Compliance is revalidated as the model changes.
- ☐ 03 Architects receive feedback close to the point of the design decision.
- ☐ 04 Findings are connected to the underlying requirement.
- ☐ 05 Previous compliance decisions and their rationale are accessible to project teams.
- ☐ 06 Jurisdiction-specific interpretations are available across the firm.
- ☐ 07 Code knowledge is not dependent on knowing which senior architect to ask.
- ☐ 08 Junior architects can understand why an issue was identified, not simply that something needs to change.
- ☐ 09 Teams can apply compliance requirements consistently across projects.
- ☐ 10 We can identify compliance issues before substantial downstream work depends on the affected decision.

STATEMENTS WE COULD NOT CHECK, AND WHERE THE DISTANCE IS LONGEST FOR US

[08 · TRY IT ON ONE ACTIVE PROJECT]

THE EXERCISE

Pick a decision made last month. Then ask two questions.

THE DECISION

DATE MADE

01

HOW LONG DID IT TAKE TO LEARN WHETHER IT COMPLIED?

02

HOW MUCH DEPENDENT WORK WAS COMPLETED BEFORE YOU KNEW?

FROM PERIODIC REVIEW TO CHECKING AS YOU DESIGN

Traditional code workflows depend heavily on periodic review. Kestrel brings that feedback loop into the building model, so compliance can be checked throughout design rather than at a milestone.

Kestrel runs inside Revit and evaluates structured code requirements against the model. A check runs when an architect runs it, on the elements in the model at that moment, and an individual check takes about thirty seconds. That makes it practical to recheck a requirement every time the design changes. Each finding identifies the model element and cites the governing code provision.

The code content behind those citations is licensed from the International Code Council, the publisher of the I-Codes. A finding points to the section as the ICC publishes it, in the edition that applies, not to a paraphrase or a summary. That is what makes a result something a project architect can verify against the book, and something a plan reviewer can follow.

The goal is not to automate architectural judgment. It is to give architects better information when they exercise that judgment.

KESTREL CHECKS THE CODE. ARCHITECTS KEEP THE JUDGMENT.

SEE KESTREL CHECK YOUR MODEL

Run it on an active Revit project.

SEE KESTREL ON YOUR PROJECT →

KESTRELLABS.COM/SCHEDULE-A-DEMO

SCAN TO SCHEDULE

© 2026 Kestrel Labs. This guide is a planning resource, not code or legal advice. Code references use the ICC model codes (I-Codes) as a common baseline; confirm requirements against the editions and amendments your jurisdiction has adopted. Online edition: kestrellabs.com/milestone