

Keeping the forest alive

- [The review cadence](#)
- [What a forest review changes](#)
- [Consistency without flattening](#)

The review cadence

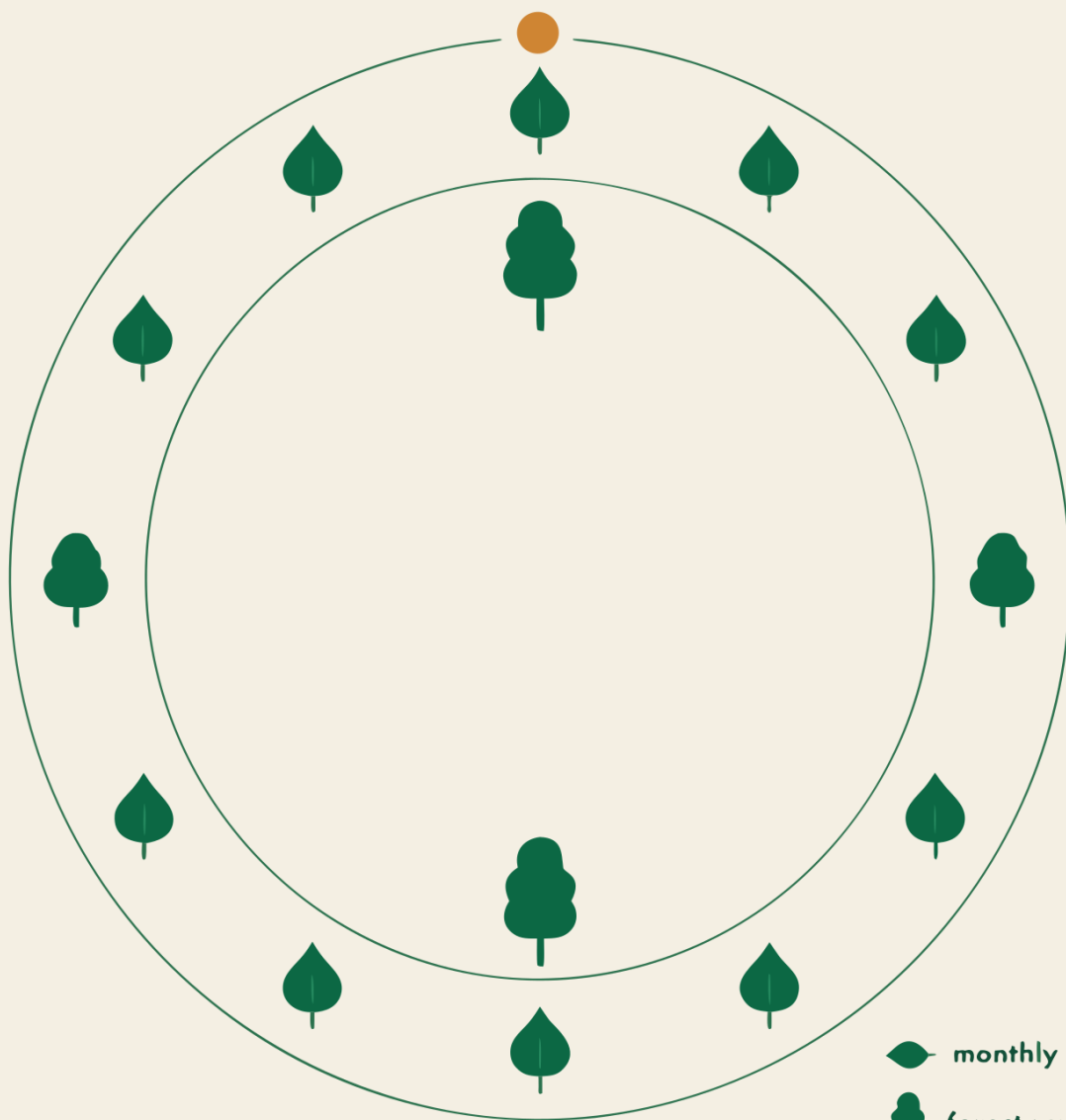
A forest is a living map, and living things need tending. The method runs on two nested rhythms — one for people, one for the forest itself — plus one hard deadline.

Individual progress: roughly monthly. A short catch-up per person keeps their levels accurate. Note the word *short* — this is a check-in, not a ceremony. And nobody is expected to advance every month; most months the honest update is "still working on the same three skills," which is exactly what well-cut skills look like mid-growth (remember: one level takes a month to three). The monthly rhythm fits fast-moving roles especially well — paired with the career progress and opportunities that justify asking more of people.

The forest itself: every six months. Fields move. Twice a year, review the forest — the next page covers what actually changes. With fast-moving fields the changes needn't be large, but they need to *happen*; a review that changes nothing is still a review that confirmed the map.

One year is the hard maximum. Past a year without a forest review, the map drifts from reality — skills that no longer matter, missing skills everyone now needs, ranks that describe last year's expectations. And people notice. The moment a team senses the forest doesn't match their real work, they stop trusting it, and a map nobody trusts is worse than no map. **Owning this cadence is the job, not an afterthought.**

The two rhythms feed each other: monthly check-ins surface the small signals ("three people said this skill feels outdated") that the six-month review acts on.



-  monthly check-in
-  forest review
-  hard maximum

What a forest review changes

A forest review is gardening, and it makes exactly three kinds of change. Walking in knowing this keeps the session focused — two to three hours with the team and the lead, not an offsite.

Skills move between ranks. Expectations shift. The skill that marked a senior three years ago — containerised deployments, say — is now table stakes for mid. The review moves it down. Occasionally the traffic runs the other way: a skill turns out rarer and more valuable than assumed, and moves up. Either way, ranks stay honest descriptions of *current* expectations rather than a fossil record of old ones.

Outdated skills retire. The framework nobody uses anymore, the process the org abandoned — cut them. Retiring skills isn't erasing anyone's history; it's keeping the *forward-looking* map clean. A forest cluttered with dead skills makes every read (coverage, gaps, rank progress) a little more false.

Emerged skills join. New tools, new practices, new demands the team now genuinely faces. They enter the forest with prerequisites attached and a rank assignment, so they arrive as part of the map, not a loose sticky note on the side of it.

Two habits make reviews cheap. **Collect as you go** — the monthly check-ins surface most of what needs changing ("this feels outdated," "we keep needing X and it's not on the map"); the review confirms and applies rather than discovers. And **review with the team, not for them** — the people doing the work know first when reality moves, and a forest changed over their heads is a forest they stop trusting.



Consistency without flattening

The moment more than one team runs forests, a tension appears. The organisation wants consistency — comparable levels, comparable ranks, reports that add up. Teams want fit — trees that match *their* work, not a committee's idea of it. Resolve this the wrong way and you either flatten every team onto one generic map, or fragment into private vocabularies nobody can compare.

The method's resolution fits in one line: **share definitions, not whole trees.**

What's shared, organisation-wide:

- **Skill names and definitions** — "React II" means the same thing everywhere it appears (skills are defined once, so this is already structural).
- **The level scale** — beginner / confident / expert, with the same behavioural tells.
- **Assessment criteria** — what counts as evidence for a level, including teaching-back.

What stays with each team:

- **The trees themselves** — which skills the team's work needs and how the area slices.
Identical trees forced onto different work destroy exactly the fit that makes forests useful.

In short: **teams own their trees; the organisation owns the vocabulary.** A light central registry of skill definitions and ranks keeps reporting meaningful while every team keeps a map that matches its actual work.

And govern lightly. A small cross-area group — not a department, a handful of people — maintains the shared vocabulary and resolves naming clashes ("your 'API design' and their 'API design' are different skills; one of them needs a new name"). Its job is to keep the language coherent, never to approve trees. The moment governance becomes a bottleneck teams route around, the shared vocabulary dies with it.

