

Skill management across the organisation

- [Roles and ranks as one system](#)
- [Ranks, evaluation, and pay](#)
- [Reporting and planning](#)
- [Rolling it out](#)

Roles and ranks as one system

Every role has a forest; every forest has ranks; every person holds skills at levels. Run across the whole organisation, this becomes a single system with answers HR and leadership usually have to improvise.

"Is this person ready for the next rank?" — a defensible answer. They hold the skills that define the rank at the required levels, or they don't. Promotion committees stop debating impressions and start checking a map both the person and their manager have been looking at all year. "Defensible" matters: the same evidence that convinces a committee protects the decision afterwards — to the person promoted, to those who weren't, to anyone auditing fairness across teams.

Roots as the promotion and hiring bar. What someone must already know to enter a forest is *derived* from the skills' own prerequisites, not guessed by whoever wrote the job description. The same derived bar serves both doors: hiring from outside and moving people across roles inside.

One organisation-wide view of capability. With shared skill definitions and a common level scale (chapter 6), forests aggregate. Leadership can see where the org is thin, where single points of failure span *teams* — the skill that three departments each assume some other department owns — and who across the whole company is near their next rank.

Sideways visibility. Where forests overlap, the map shows where people can move laterally and what they'd need to pick up — turning internal mobility from a lucky accident into something you can actually plan.

The compounding effect is the point: each team's honest map is useful; all the maps together, speaking one vocabulary, are an organisational capability the org never had before — the ability to *see itself*.



Ranks, evaluation, and pay

Connecting the forest to evaluation — and eventually money — is where the method carries the most weight and deserves the most care. Three principles.

Anchor reviews on levels and ranks, not adjectives. "Moved three skills to confident, holds the mid-rank set" is comparable year over year, across managers, across teams. Adjective-based reviews aren't just vague — they're *incomparable*, which makes every calibration meeting a negotiation between writing styles. Levels give the organisation one currency of progress.

Connect ranks to bands deliberately. The tempting shortcut — pay scales with skill count — fails quickly: a raw count rarely maps cleanly to value, and it incentivises collecting cheap skills over deepening critical ones. Instead, decide *which* skills and ranks matter for *which* band, and **write it down** so it's consistent across managers. The forest gives you honest inputs; the compensation design is still a human decision — the method just makes it explicit instead of folklore.

Keep assessment honest under pressure. The predictable failure: once levels touch pay, they inflate — self-ratings and manager ratings alike; not from dishonesty, but because incentives lean on the scale. The counterweights from chapter 5 stop being optional at this point: **peer validation** (a claimed level is one colleagues confirm) and **skill champions** (each critical skill has a reference person who validates assessments). An organisation that ties pay to levels without standing up validation first is scheduling its own credibility crisis.

Sequencing advice hides in that last line: let the forest prove itself in reviews and development *first*, then connect money. A map the org already trusts can bear the weight; a brand-new one can't.



Reporting and planning

With every team's forest speaking the same vocabulary, the organisation gains four reads that used to require expensive guesswork.

Coverage and gaps. Which critical skills are thin org-wide? Where are the *cross-team* single points of failure — the one person in the whole company who holds a skill four teams depend on? Team-level maps each show their own risks; only the aggregated view shows the risks that live between teams.

Talent pipeline. Who's close to their next rank — nearly ready to step up, or step across? This turns succession from a name-on-a-napkin exercise into a live list: for each key position or rank, the people within a few skills of it, and exactly which skills. Development budgets can be pointed at the shortest real distances.

Capability vs. strategy. Take the next two years' priorities and read off the skills they'll demand — then compare against current coverage. The shortfall *is* the workforce-planning agenda: what to grow internally (and how long the prerequisite chains say that takes), what to hire (and against which derived entry bars), what to reconsider. Strategy meetings acquire an unusual property: the capability question gets answered with data.

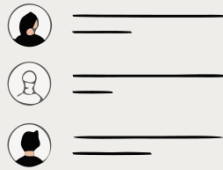
Project readiness. Before committing to an upcoming project, read off the skills it needs and check how many people hold them, at what levels, and where the gaps are. "Can we staff this?" becomes a lookup instead of an opinion — and when the answer is no, the map shows the cheapest path to yes.

A note on spirit: these reads are for *planning*, not surveillance. The moment reporting makes individuals feel watched rather than developed, honest assessment — the input everything rests on — starts to corrode. Report on the forest; develop the people.

Coverage



Pipeline



Strategy vs coverage

Product A	High
Product A	High
Product B	Low
Product B	Low

Project readiness



Rolling it out

The org-wide system of the previous pages is the destination. Nobody should start there. Skillforests spreads the way trust spreads — one team at a time, on demonstrated value.

Pilot with one willing team. Willing is the load-bearing word. A team curious about the method will forgive early clumsiness and help refine it; a team that had it imposed will comply minimally and prove it "doesn't work here." Build the first forest with them, run the monthly rhythm for a few cycles, hold one forest review — and *learn what works in your organisation* before any thought of scaling.

Make the value concrete for individuals first. This is the rollout's golden rule: people engage when the forest helps *their own growth*, not just the org's reporting. If the pilot team's members can each point at something the forest gave them personally — a visible path to the next rank, a review that finally felt fair, a skill they grew on purpose — the method sells itself internally. If the first visible artifact is a management dashboard, you've taught everyone the map is for watching them, and honest assessment dies in the crib.

Scale gradually. Let the pilot's story travel; expand to the next willing teams; stand up the shared vocabulary and light governance (chapter 6) as the *second* or third team joins — that's when consistency starts earning its cost. Connect ranks to evaluation, and eventually pay, only after the maps have earned trust — the sequencing argued in the pay page.

The pattern is the method applied to itself: clear the roots first (a willing team, individual value), don't skip prerequisites (trust before money), and let visible monthly progress do the persuading.

