Growing a team

The team map

Point a forest at a team and it becomes something new: one picture of what the team can actually do. Three reads matter, in this order.

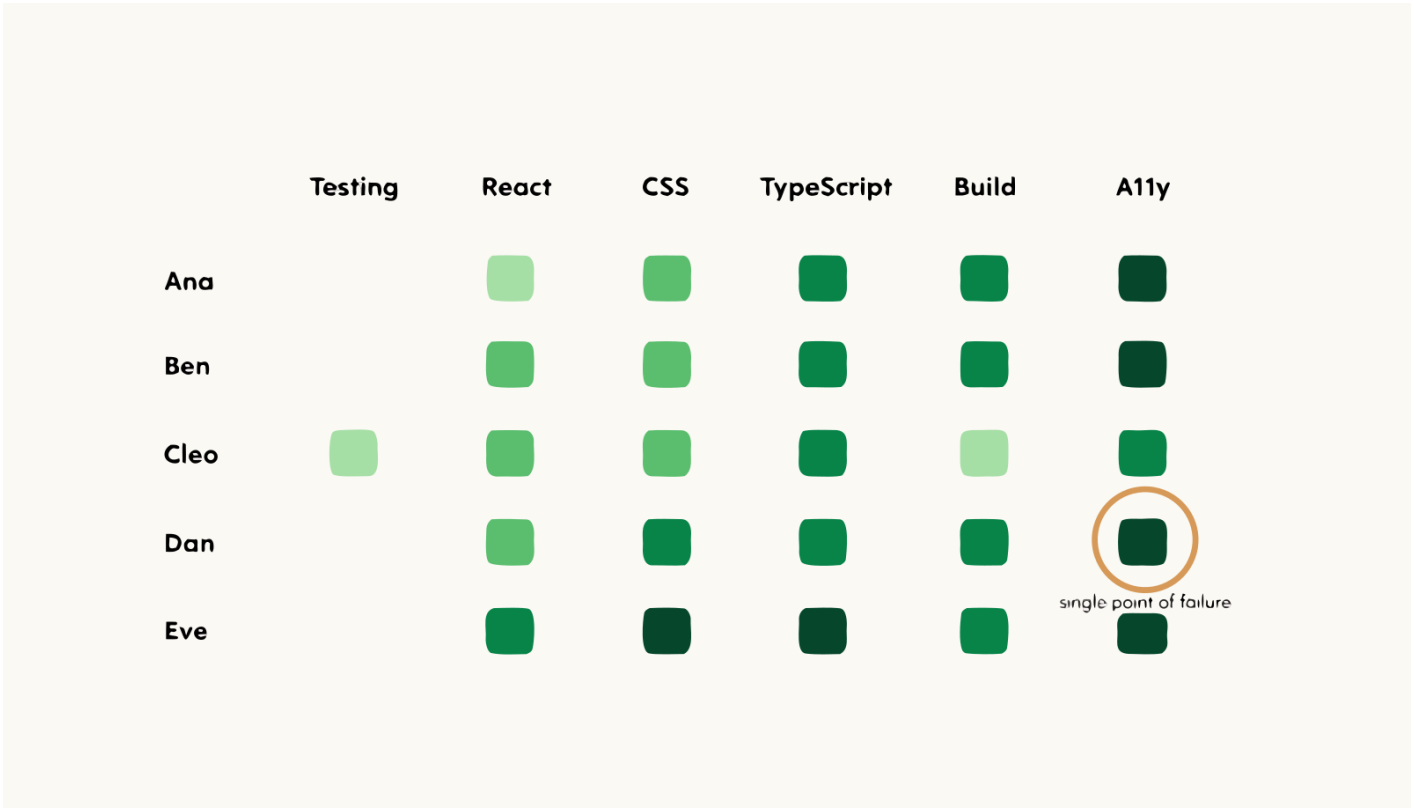
Coverage. For each critical skill: how many people hold it, and at what level? Read this first, because it's the first thing that breaks. A team's real capacity isn't its headcount — it's its coverage of the skills the work demands.

Gaps. Skills the team needs but nobody has — or holds only at beginner. A gap on a critical skill is a commitment you can't currently keep, discovered *before* you keep it. Gaps found on the map cost a plan; gaps found in production cost much more.

Single points of failure. Skills exactly one person holds at a usable level. This is the read leaders learn to fear productively: a skill with one confident owner is a risk the moment that person is on leave, busy, or gone. The map shows you every such skill *today*, while there's still time to grow a second owner.

The tool that makes all three reads instant is the **skills heat map**: team members down one axis, skills across the other, level in each cell. The whole forest becomes one glance — dark columns are strengths, thin columns are the work, and a column with a single dark cell is a single point of failure announcing itself.

Everything else in this chapter — planning, hiring, reviews, development — is a response to what these three reads show you.


Testing
React
CSS
TypeScript
Build
A11y
Ana
Ben
Cleo
Dan
Eve
single point of failure

Planning growth

A team's development plan shouldn't need inventing — the map, read against the future, mostly writes it. Three inputs:

Match the forest to upcoming work. Take the next projects on the horizon and read off the skills they'll demand. Compare against current coverage. The shortfall *is* the plan — and because prerequisites and roots dictate learning order, the map even sequences it. "We need two confident owners of X by spring, and X requires Y, so Y starts now" is planning at its least mysterious.

Grow people toward the next rank. Each person's target rank is a concrete set of skills — so "develop your people" stops being an abstract duty and becomes specific: these skills, for this person, in this order. The beauty is that personal ambition and team need mostly align; the rank was defined to describe what the team values.

Spread risk deliberately. Wherever the heat map shows a single point of failure, grow a second owner *before* you need one. This is the cheapest insurance a team can buy: the learner gains a valuable skill, the current owner gets the expert-making experience of teaching it (teaching-back — the confident→expert test), and the team sleeps better. One decision, three payoffs.

Braid the three inputs together per person — what the work needs, what their next rank asks, what risk demands — and keep each person's active set to the usual three-to-five skills. The result is a growth plan tied to real work, honest about order, and legible to everyone on the map it came from.

Hiring against the forest

Most job openings are written from memory and vibes — a list of technologies, some adjectives, "5+ years." A forest lets you write one from the map instead.

Define the opening by roots and target rank. The forest's roots are the derived, honest answer to "what must someone already hold to enter this role?" Add the rank you're hiring at, and the opening defines itself: the entry requirements, plus the rank's skill set at its levels. Sharper than buzzwords — for you *and* for candidates, who can see what the job actually is.

Separate must-have from grow-into. This is the hiring mistake the map prevents best. Roots and the entry rank are the bar; everything above is what the role *teaches*. Screening candidates out on skills the job exists to develop shrinks your pipeline for no reason — you're demanding people pre-learn the growth you're supposedly offering. The forest makes the line explicit: below this line, must arrive with; above it, will grow here.

Read candidates onto the map. Interviews assess the same way the team assesses — skills at levels, using the same behavioural tells (needs help along the way? delivers alone? could teach it?). Each candidate becomes a shape on the heat map: this one closes the testing gap, that one duplicates strengths you already have. "Best candidate" becomes "best candidate *for these gaps*" — a question the map answers almost by itself.

A quiet bonus: openings written this way promise honestly. Candidates arrive knowing what they must bring and what they'll grow — and the ones the map welcomes tend to be the ones the forest was missing.

Reviews and 1:1s on levels

Reviews are where the forest pays for itself most visibly — because it replaces the worst part of them: the adjectives.

Talk in levels and ranks. "Moved three skills to confident this half, holds the full mid-rank set, two skills from senior" is reviewable — comparable to last year, legible to both sides, checkable against the map. "Showed real growth" is none of those things. Once a team tastes level-based reviews, adjective-based ones feel like astrology.

Use the forest as the shared artifact. The review conversation happens *over the map*, literally. Both people look at the same forest, and disagreement — the thing that makes reviews dread-inducing — changes character entirely. Instead of "I think you're underrating my year" (a conflict about judgment), it's "I'd put myself at confident on X" (a conversation about one skill at one level, with behavioural tells to check and a champion to consult if needed). Big vague conflicts become small concrete ones, and small concrete ones resolve.

Next steps are already on the map. The traditional review scramble — inventing "development goals" in the meeting — disappears. The next rank's skills *are* the plan for the period ahead; the only real conversation is sequencing and which real work will carry each skill. The person walks out with the same map they walked in with, plus agreement on the path.

The 1:1 version is the same thing, smaller: the monthly check-in from the review cadence. What moved, what stalled (look down — usually a prerequisite), what's the focus next month. Ten minutes over the map, and the annual review loses its power to surprise anyone.

Developing the team

The map shows what to grow; these are the mechanisms that do the growing — and the guardrail that keeps it healthy.

Skill champions. For each critical skill, name the person who can answer questions, mentor, and validate level assessments. Champions turn every important skill into a place someone owns: learners know who to ask, assessments have a reference point, and the champion gets the recognition of being *the* person for something.

Cross-training. Wherever the heat map shows a single point of failure, pair a learner with the confident or expert owner. The learner grows a needed skill, the owner grows through teaching it, the column gains a second cell.

Learning partnerships. Match complementary gaps — she needs what he has, he needs what she has — and let them trade. Then *rotate the pairs*, so knowledge spreads through the team instead of pooling in corners.

And the guardrail: **keep it from becoming a contest.** A visible map of everyone's levels can curdle into a leaderboard if led carelessly — people hoarding expertise to stay special, inflating levels to keep up appearances. The countermeasures are cultural, and they're the leader's job:

- **Emphasise collective coverage over individual scores.** The team's win is a dark heat map, not a personal high score.
- **Recognise the people who teach.** If sharing knowledge is what gets celebrated, hoarding it stops paying.
- **Start with volunteers and show an early win** — enthusiasm converts skeptics better than mandates.
- **Keep check-ins brief and tied to real work** — the moment forest upkeep feels like ceremony, it starts dying.

A team run this way develops a distinctive feel: growth is normal, teaching is status, and nobody is irreplaceable in the brittle way — only valued in the durable one.