

The Skillforests method

- [What Skillforests is](#)
- [Where it comes from](#)
- [The problems it solves](#)
- [How to read this book](#)

What Skillforests is

Skillforests is a way of managing skills that borrows its best idea from video games: the skill tree. Instead of job descriptions full of adjectives and reviews full of vibes, every role gets a **forest** — a living map of all the skills that role needs, how they connect, and the order in which they're learned.

A forest is made of **trees** (areas of knowledge, like "Testing" or "Styling"), trees are made of **skills**, and every skill is something concrete you either hold or don't — at a level you can name. Look at someone's forest and you can see where they stand, what they can take on next, and exactly what the next promotion requires.

Three things make it different from the competency frameworks you may have met before:

- **It's team-specific.** The trees are built *with* the team doing the work, not downloaded from a generic framework. A map of the work you actually do beats a map of work in general.
- **It's collaborative.** Building and maintaining a forest is a shared activity, and that's part of the point — the conversations are as valuable as the map.
- **It's honest by design.** Levels are defined so concretely that placing yourself doesn't require guessing, and the whole method leans on keeping that placement truthful.

And you don't need software to start. The method has been run for years on paper, sticky notes, and wall charts. The map matters more than the medium.



Where it comes from

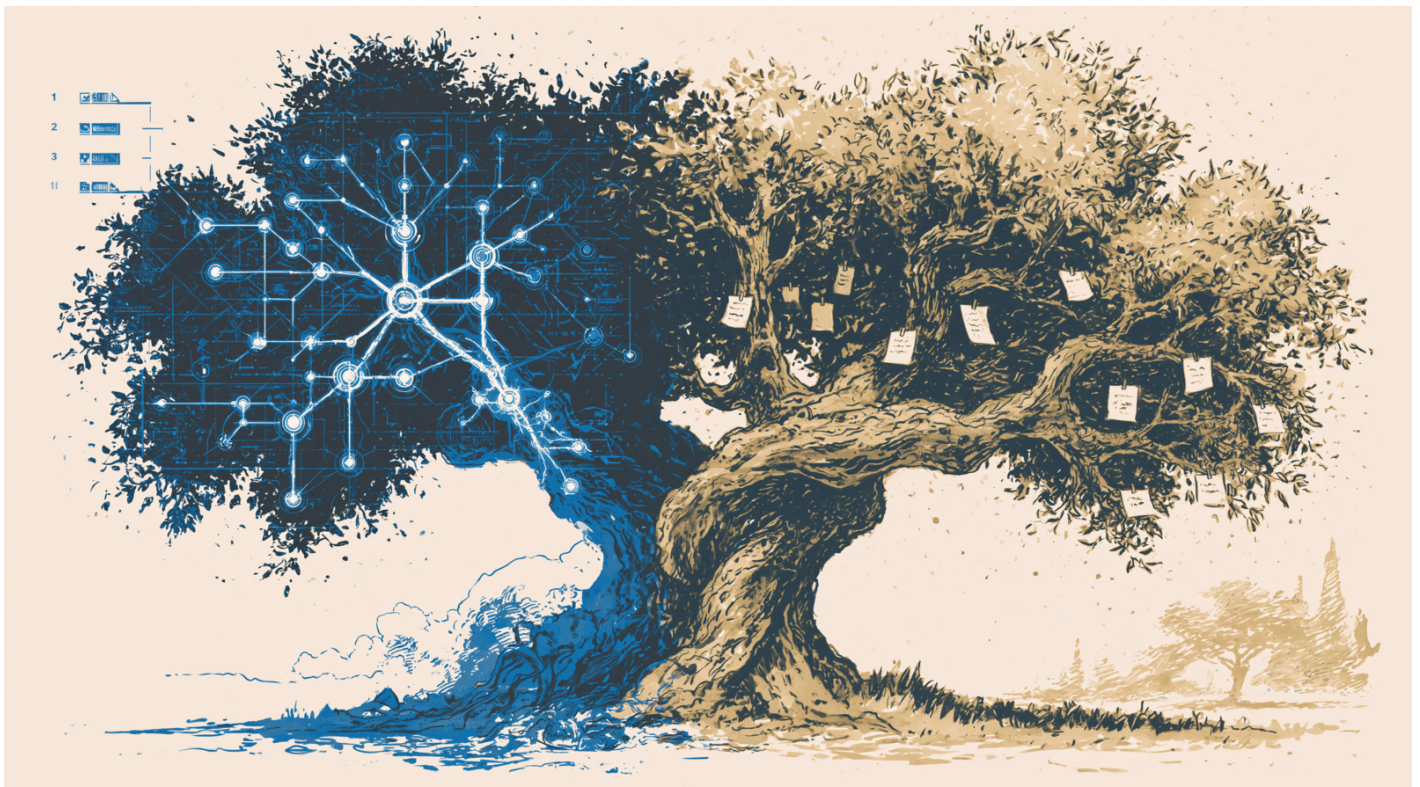
In early 2020, the question was simple: professional competencies relate to each other in complex ways — could those relations be drawn as a graph that's actually easy to understand?

Games had already solved this. Path of Exile, the Diablo series, Skyrim — all of them present sprawling systems of skills and experience to players through some version of a **skill tree**, and players navigate them without a manual. You see where you are, what's unlocked, what's next, and what it costs. Nobody writes a self-assessment essay to level up.

The move Skillforests makes is taking that presentation layer and pointing it at real professional competencies. Two convictions shaped how:

- **Trees should be team-specific.** A generic "front-end developer framework" fits nobody exactly. A tree built around the work one team actually does fits like a glove — so the method builds trees per team, not per industry.
- **Trees should be built collaboratively.** With a background in Design Thinking, the natural way to build them was together — workshops where the team maps its own work into skills and trees. People trust a map they helped draw.

The method ran for years on paper before it was ever software — growing real teams from a handful of people to a department. The forest metaphor stuck because it earned its keep: skills grow, trees have roots, and a healthy forest needs tending.



The problems it solves

Most organisations manage skills through a fog. The information exists — in people's heads, scattered across CVs and review documents — but nobody can see it. The forest replaces the fog with a map, and a few chronic problems dissolve on contact.

Reviews made of adjectives. "Shows real growth." "Strong performer." Nobody can compare that year over year, and both sides of the table know it. With a forest, a review reads "moved three skills to confident, holds the mid-rank set" — concrete, comparable, and visible to both sides before the meeting starts.

Promotions without a bar. "Is she ready for senior?" is a debate when senior is a feeling. When a rank is a defined set of skills at defined levels, readiness is something you check, not argue about — and the person can see the exact shape of their next step all year long.

Invisible gaps and single points of failure. The skill only one person holds is invisible right up until that person is on leave, busy, or gone. A forest shows coverage at a glance: which skills are thin, which have one owner, where the next project will hit a wall.

Hiring by buzzword. Job ads list technologies; interviews probe vibes. A forest defines an opening by what a hire must actually walk in knowing (the roots and entry rank) versus what the role will teach them — a sharper filter for everyone, including the candidate.

Learning without direction. "I should learn more" is a mood, not a plan. A forest turns it into an ordered path: these skills, in this order, a few at a time.



How to read this book

The book has two parts, and the split is deliberate.

Part I (chapters 1-6) is the shared foundation. It defines the building blocks — skills, levels, trees, roots, forests, ranks — and the two disciplines that keep a forest trustworthy: honest assessment and regular upkeep. Everyone should read Part I, whatever their role. It's short, and every later chapter leans on it instead of re-explaining terms.

Part II (chapters 7-10) is one chapter per way of using the method. Read the one that matches your seat; skim the others to understand what the people around you are doing with the same map.

If you are...	Read	You'll use the forest to...
Growing your own career	Chapter 7	See where you stand and pick your next skills, in order
Leading a team	Chapter 8	Read coverage, gaps, and risks; plan growth and hiring
Coaching or training people	Chapter 9	Baseline each person, find the gap, guide the path
Running HR or leadership	Chapter 10	Tie ranks to reviews and pay; report and plan org-wide

A **glossary** at the back holds every term in one place, for the moments you need a quick reminder rather than a chapter.

One reading tip: the method is simpler than a book-length treatment makes it look. Skills connect to skills; trees group them; forests group trees; ranks slice forests by seniority. Everything else is practice and discipline. If you ever feel lost, come back to that sentence.