

Why R&D Knowledge Walks Out the Door When People Leave

09/06/2026 1:06 pm IST

Every R&D-active company has experienced it. A senior engineer leaves. A key developer moves on. And with them goes three years of accumulated technical knowledge. The company loses knowledge of what was tried, what failed, what the constraints were, why the current approach was chosen over the alternatives that were tested and abandoned.

The new person starts. They spend months getting up to speed. Sometimes they repeat work that was already done. Sometimes they make decisions that the departing engineer would have immediately recognised as a dead end. The cost is in time, in rework, in the erosion of technical momentum.

This is usually treated as a people problem. It is not. It is a structure problem.

Where the Knowledge Actually Lives

In most engineering and software companies, R&D knowledge is distributed across three places: the heads of the people who did the work, the project management tools those people used (Jira, SharePoint, ERP, email), and, if the company is lucky, some form of technical documentation that was written under deadline pressure and is already out of date.

None of these is a knowledge system. They are artefacts of activity, not records of learning. They capture what was done, not what was understood. What was decided, not why, and not what was tried first.

When someone leaves, the artefacts remain. The knowledge does not.

The structural fix is not primarily a documentation exercise. It is an architecture question: how is R&D activity organised, and what does it connect to? In companies where R&D is managed purely at the project level, technical knowledge is siloed by project. When the project ends and the team disperses, the knowledge disperses with it.

What a Technology-Structured Approach Looks Like

Companies that retain R&D knowledge effectively share a common structural practice. They organise their R&D not just around projects, but around the core technologies those projects are advancing.

A core technology is a named area of applied science or engineering that the company is systematically developing that persists beyond any individual project. In precision engineering, it might be a specific manufacturing process. In industrial automation, it might be the system architecture that underpins how the company integrates new capability into existing production environments. In software, it might be the proprietary algorithm or security model at the heart of the product. In agri-food, it might be the plant physiology knowledge that drives cultivar development.

When R&D activity is connected to these named technologies, not just to the project it happened within, something changes. Investigations accumulate against the technology over time. A new engineer joining the company can see not just what the current approach is, but how it was arrived at, what alternatives were explored, and what the state of the art was when key decisions were made. The knowledge lives in the technology record, not in the person.

This is what ReaDI Watch calls a Core Technology History File. Its a running compendium of everything the company has learned in a specific technology area, built up across projects, teams, and years.

The Co-ordination Problem in Disguise

For many R&D system owners and CTOs, the knowledge retention problem shows up first as a coordination problem. You are chasing engineers for updates. You have no single view of what R&D is happening across teams. When a grant report or a tax credit claim is due, you are reconstructing history from emails and Jira tickets.

These are symptoms of the same underlying issue: R&D activity is not connected to a structure that persists beyond the immediate project. The coordination problem is hard because there is nothing stable to coordinate around. The reporting problem is hard because there is no contemporaneous record to report from.

In manufacturing companies, in MedTech, in software, in industrial automation, the pattern is the same. The companies that have solved the coordination problem have almost always done so by building a technology-level structure first, and letting the project and work-package level activity connect to it.

What this Means in Practice

The question worth asking is not 'how do we document better?' It is 'what are the technologies we are advancing, and are our projects and work packages connected to them in a way that builds a lasting record?'

For most companies, the answer to the first part of that question comes quickly. Engineers know instinctively what the core technical areas are. The harder part is building the structure that connects ongoing activity to those technologies in real time, rather than reconstructing it retrospectively when someone leaves or an audit arrives.

ReaDI-Watch's Core Technologies framework is designed to make that connection practical. Not as an additional documentation burden, but as the organising structure that makes R&D manageable at scale.

Your coordination problem is probably a structure problem. Worth understanding the difference before the next person leaves.
