

You cannot scale a product nobody has written down

The line doubles, a second source is qualified, the build moves to a new site — and the engineering definition that six people carried in their heads does not come with it. Half a day and one engineer will tell you whether it is there.

The order lands, or the volume triples, or the build moves to a second site — and the shop floor starts producing questions nobody had to ask before. Which revision is on that unit. Whether the part from the new supplier is the same part. What exactly shipped last March. The plant scaled. The engineering definition underneath it did not, because it was never written down in the first place. It was carried by the same six people who have always built the thing, and it does not come with the capital.

That is why the usual evidence is so misleading. A forty-person engineering department, an ISO 9001 certificate on the wall, a product that has shipped for eleven years — none of it establishes that the product is defined. ISO 9001 certifies that a company follows its own documented procedure. It says nothing at all about whether that procedure defines the product well enough for somebody else to build it. A large engineering department tells you how many people are employed, not what they have written down. And a product that has shipped for eleven years is often the strongest case of all for the foundation being absent, because it was built by the same six people who have always built it, and the definition lives in them. Ask those six to draw it and you find out what you actually own.

It usually surfaces as a tool question: which PLM should we buy. But before the vendor demonstrations, before the requirements workshop, before anybody draws a boundary between PDM, PLM and ERP, run a presence test on the engineering skeleton. Not a maturity assessment — a presence test. Each item either exists or it does not. That binary is what keeps the check fast and completely independent of which system you eventually buy.

What has to be there

Part numbers that exist, are unique, and where one number means one thing. A drawing standard that exists and that somebody actually applies. Parts lists that drive the bill of materials rather than sitting beside it as a second, quietly divergent record. Tolerances specified on the dimensions that matter rather than defaulted or left blank. Fasteners and torques defined rather than assumed by whoever is holding the wrench. A change that is a document with a number rather than a conversation — if changes travel by email and verbal authorisation, there is no change process, whatever the quality manual says. And a change control board that exists, has a named chair, and can block.

Then there is a second tier that is absent more often than it is present, and expensive in proportion to how quietly it fails.

THE SEVEN THAT COST THE MOST WHEN THEY ARE MISSING

- **The revision rule.** When does a change take a new revision, and when does it take a new part number? This is the form-fit-function trigger. Without it the same part number describes two different parts in the field — the most expensive single ambiguity in an installed base.

- **Released versus in-work, with a named approver.** A drawing with no approval signature is a suggestion. It costs a title block and a signature.
- **A general tolerance note, and datums.** The general note covers everything not explicitly dimensioned, which is most of the drawing. And a tolerance without a datum is a wish: two inspectors measure from different faces and both pass.
- **Material, finish and process callouts.** Left blank, the supplier chooses. It will be whatever passes visual inspection most cheaply, and it will change without notice.
- **Firmware carried as a part number on the bill of materials.** In equipment and power electronics the shipped build is routinely untracked. If firmware is not on the BOM, the as-built record is incomplete and field diagnosis is guesswork.
- **Effectivity.** The serial number or date from which a change applies. Without it you cannot answer what is actually installed at that site, which means you cannot scope a retrofit, a recall or a warranty reserve.
- **Key characteristics flagged.** Which dimensions get measured at receiving inspection. Unflagged, the answer is everything or nothing, and it is nearly always nothing.

How to run it in half a day

One engineer, no system access beyond read. Take one unit that shipped. Pull its bill of materials. Walk down two levels, pick a child part at random, and ask for the drawing. Then ask which revision of that drawing shipped on that serial number.

That single trace exercises part numbering, BOM integrity, drawing standards, revision control, effectivity and the as-built record in one pass, and it does not depend on which system the answers come out of — a shared drive and a filing cabinet will answer it if the definition is there. Where the trace stops is the finding. Run it three times on three different units and you will have a map of the foundation that no vendor demonstration would ever have given you.

Where the trace stops is the finding.

Why this has to come before the tool, not with it

Everything in that check is tool-agnostic. It is equally required in PLM, in ERP, on a shared drive, or on paper. That is precisely what makes it a precondition to selection rather than a part of it. The requirements workshop assumes there is something to have requirements about.

A tool does not supply what was never written down. It reproduces the gap faster, at considerably more expense, and — this is the damaging part — with a system of record that now looks authoritative. Before the implementation, everyone in the building knew the part numbering was a mess and worked around it. Afterwards, the mess has a login screen and an audit trail, and people believe it. The organisation has bought certainty it has not earned, and the first time that certainty is tested is usually a field failure, a recall scope, or a second-source qualification that cannot be completed.

The same argument runs one level up, which is where most programmes go wrong. Teams spend weeks drawing boundaries between PDM, PLM and ERP — what data lives where, which system is master — and the boundary exercise is sound. But you cannot draw a boundary around data that was never defined. So the question to settle first is not which tool. It is whether there is anything to put in one — and that costs half a day to answer, against rather more for the tool.

A fancy PDM on top of undocumented engineering standards is a mess with a fancy UI.

Where else it shows up

Once you start looking for the skeleton you find it underneath a surprising number of other problems, which is the argument for checking it first rather than diagnosing each symptom separately.

An interface control document is unbuildable without part numbers, drawing standards, datums and tolerances underneath it — form and fit are tolerance-and-datum statements, function is a key-characteristic statement, and none of those mean anything if the underlying definition is absent. In a supplier audit the two-level trace runs on site in twenty minutes, and an absent skeleton is a reliable leading indicator of the change-order churn, revision confusion and traceability gaps that surface months later as yield and test-data findings. Before asking whether a supplier can build at scale, ask the cheaper prior question: is the product defined well enough for anyone other than the incumbent team to build it at all? And in diligence, an undefined product is a transferability problem — it bounds second sourcing, offshoring, footprint moves, and any synergy case that assumes another plant can build the same thing.

No assumptions

The discipline here is a refusal, not a technique. Do not assume the engineering foundation is there. It is assumed constantly, by capable people, on evidence that does not support the assumption — a licence, a certificate, a headcount, a shipping history. The check exists because the assumption is wrong more often than it is right, and because finding out costs half a day before the programme and a great deal more after it.

That is the work I do.

WHERE THIS GOES NEXT

Knowing exactly what you ship →

The product definition and change control method — part numbering, BOM architecture, and a change process your managers operate without you. Or start a conversation: rtackconsulting.com/contact