Generic software makes you fit it. It should fit you.

The case for industry-led software — why business tools should arrive already shaped to your sector, not the other way round

Version 1.0 · ixlcore.com

Executive summary

Most business software is built to fit everyone, which is another way of saying it fits no one in particular. It ships as a blank shape and asks the business to fill it in — to translate its own trade into the software's generic vocabulary, to rebuild its workflows inside someone else's assumptions, and to do it all before a single day of real work gets easier. The software never quite reflects how the business actually operates, so onboarding drags, adoption stalls, and the tool that was meant to run the business becomes one more thing the business has to run.

This paper makes the case for the opposite approach: **industry-led software** — software that arrives already shaped to a specific sector, with that sector's language, workflows and rules in place on day one. We explain why the usual answers (generic horizontal suites, a pile of niche vertical tools, custom builds) each fall short, and how one shared platform foundation combined with **industry rule-packs** delivers software that fits the business from the start — without fragmenting into a hundred incompatible products.

The problem: one shape for every business

A law firm, a construction contractor and a wellness spa run on completely different logic. The firm tracks matters and billable hours. The contractor tracks jobs, costs against a budget, and retentions held until practice completion. The spa tracks appointments, therapists, packages and commissions. Three trades, three vocabularies, three ways of counting money and time.

Generic software forces all three into the same shape. It offers a "project", a "record", a "transaction" — abstract boxes that mean nothing to anyone until someone spends weeks deciding what each box will stand for in *their* world. That translation is the hidden cost of generic software, and it lands as a bill:

- **A long onboarding.** Before the software does anything useful, the business must configure it — mapping its real work onto generic fields, often with a paid consultant translating.
- **Poor adoption.** Staff open a tool that speaks a foreign language. The estimator sees "line items" where they think in "measured quantities". The therapist sees "resources" where they mean "the 2 o'clock with Amara". Software that doesn't speak your trade is software people avoid.
- **A permanent gap.** However hard you configure, the tool never quite reflects how the business really operates — so the spreadsheets that hold the real logic never fully go away.

The business bends itself to fit the software. It should be the other way round.

Why the usual answers fall short

Three approaches claim to close this gap. Each trades one problem for another.

Horizontal generic suites are built to sell to every industry, so they commit to none. Their strength — flexibility — is also the trap: a blank canvas that must be painted by the buyer. You get power in exchange for a project. The suite can eventually be shaped to your trade, but only after a long, expensive configuration that many businesses never finish, and that quietly rots as staff turn over and no one remembers why a field was set the way it was.

A pile of niche vertical tools goes the other way — software built specifically for salons, or for law firms, or for builders. It speaks the trade fluently, and on its own it fits beautifully. But each niche tool only covers one corner of the business. The salon booking app doesn't do the accounting; the legal practice tool doesn't run payroll; the site-management app doesn't manage stock. So the business ends up with a fluent tool for each corner and no connection between them — trading a fit problem for a coordination problem. (That coordination problem is the subject of a separate paper; here it's enough to note that fragmenting into niche tools solves fit by sacrificing everything else.)

Custom builds promise a perfect fit — software written for exactly one business. And they can deliver it, once. But a bespoke system is expensive to build, slow to change, and fragile to maintain. It fits the business as it was on the day it was specified, then falls steadily behind as the business grows. Most SMEs neither can nor should be in the software-development business.

So the growing business is caught: generic tools that fit no one, niche tools that fit one corner, or custom builds that fit once and then decay.

A different idea: one foundation, many industries

Industry-led software resolves the trade-off by separating what every business shares from what each industry needs.

Underneath, there is **one shared platform foundation** — the machinery every business relies on regardless of sector: a customer record, a chart of accounts, employees, tax rules, currencies, permissions, documents, approvals. This is built once, properly, at the platform level.

On top of that foundation sit **industry rule-packs** — the vocabulary, workflows and rules of a specific sector, arriving already in place. The rule-pack is what makes the software speak your trade on day one. It is not a configuration project you undertake; it is how the software shows up.

This is how you get the fluency of a niche tool and the breadth of a full platform at the same time — without fragmenting into a hundred incompatible products, and without asking each business to become its own software team.

How it works

The foundation is built once and consumed by every app. IXL CORE runs its applications — CRM, Accounting, Supply Chain, HR & Payroll, Operations and Marketing — on one shared foundation. The customer, the ledger, the employee list, the compliance rules: each is defined once and used

everywhere. No app keeps a private copy. This is what lets an industry rule-pack change how the *whole* business behaves, not just one screen.

Rule-packs add each industry's language, workflows and rules on top. A rule-pack shapes what the apps show and do, so the same underlying platform presents itself differently depending on the trade:

- **Professional services** — matters, engagements and billable time, with work-in-progress and time-based billing built in, so a law firm or consultancy sees its practice, not a generic "project".
- **Construction & contracting** — job costing against budget, progress claims and retentions held to completion, so a contractor's margin on each site is visible as the job runs, not discovered at the end.
- **Wellness & beauty** — appointments, therapists and rooms, packages, and staff commissions, so a spa or salon books its diary and pays its people the way the trade actually works.
- **Asset-heavy operations** — the full asset lifecycle, from acquisition through maintenance schedules to disposal, so businesses that run on physical plant and equipment can track condition, servicing and value over time.

The same shared foundation serves healthcare, manufacturing, retail & e-commerce, property & real estate, hospitality & leisure, asset management, nonprofits & NGOs, education & training, and logistics & distribution — each with its own rule-pack, all on one platform.

What it looks like in practice

Picture two businesses opening the software for the first time.

On generic software. The contractor logs in to a blank suite. There is no "retention", no "progress claim", no "job cost report" — only generic building blocks and a manual explaining how to assemble them. Before the software helps cost a single job, the contractor is booking consulting days to translate the trade into the tool. Weeks pass. Staff, handed a system that doesn't speak building, keep the real numbers in a spreadsheet.

On industry-led software. The contractor logs in and the software already knows what a job is, what a retention is, and how to cost work against a budget. The estimator sees measured quantities; the finance lead sees claims and retentions; the director sees margin per site — all in the language of the trade, from day one. There is no translation project, because the translation was already done in the rule-pack. And because it all sits on one foundation, the job that gets costed is the same job that gets invoiced, staffed and reported on across the rest of the business.

The outcome

When software arrives already shaped to the industry, the business stops paying the translation tax. Onboarding shrinks from a configuration project to a start date. Adoption improves because staff open a tool that speaks their trade, not a foreign one. The software finally reflects how the business actually operates — so the shadow spreadsheets can retire. And because the fit comes from rule-packs on a shared foundation rather than from a niche tool or a bespoke build, the business gets that fit *and* stays connected across its whole operation — sales, money, stock, people and work as one.

The question for any growing business is no longer *"how do we bend this software to fit us?"* It's *"why were we ever the ones doing the bending?"*

About IXL CORE

IXL CORE is the Business Operating System for growing businesses across Africa — one connected platform for customers, money, stock, people and work, shaped to your industry by rule-packs on a shared foundation, and built for how businesses in Kenya, South Africa and across the continent actually operate. Learn more at ixlcore.com, or start at app.ixlcore.com.