



PAPER • CCG INTELLIGENCE • 10 SEPTEMBER 2026

Businesses Are Full of Work That Software Should Be Doing

Over the last six months I have sat in a lot of businesses drowning in work a computer should have taken over years ago. A property firm running a large portfolio off a single spreadsheet. A retailer managing procurement inside a project-management app it had quietly outgrown. An accounting practice building the same client document pack by hand, every time, from information it already held. None of them came to us asking for AI. They asked for the work to stop.

I wrote here recently that we won't sell you the robot, and that the half of an automation project nobody demonstrates is the half that decides whether it works. This is that other half: what the technology looks like when you build it around a business instead of selling it one.

The first thing you learn is that the work worth automating hides. It almost never sits where the org chart would put it. It's in the spreadsheet someone has reconciled by hand for years. It's in the head of the one person who knows how the stock really gets counted and which supplier to phone when it doesn't, so when they go on leave the process goes with them. And a surprising amount of it lives in WhatsApp, where the enquiry arrives, the quote goes

out, and the only record is a chat thread until the phone gets lost. The vendors sell automation for the tidy, visible process. The real money is usually in the other two, and you only find it if you go and look.

A couple of examples, names left out. The accounting practice came to us with one job: reconciling bank transactions against their retail clients' till records, slow and manual and quietly hated. We built it, it went into daily use, and then the thing happened that happens almost every time. Once people watch one job disappear, they start pointing at the next one. That practice has since asked us for four more. Their client document packs are now generated from a template library, with AI lifting the details off whatever the client sends in. Their clients' staff get payslips and tax certificates back over WhatsApp in seconds, no app, no login, because everyone already knows how to use WhatsApp.

Another was a retailer who genuinely couldn't see their own stock. Now a purchase order comes up with last month's and last year's sales sitting next to it, the goods-received note fills itself in from the supplier's invoice, and the low-stock warnings come from how fast things actually sell rather than a number somebody guessed years ago.

None of this started with a product. We didn't arrive with a platform to install. We sat in the business, found the specific thing bleeding time or money, and built around how the place already works, instead of asking everyone to bend around software made for someone else. That is more or less the whole method. The rest is things we've learned the hard way. Keep the first build small and let the business tell you what comes next. Expect the bottleneck to move rather than disappear, because it always does. Make sure a real person owns the process once software is running it. And meet people where they are: if the team lives on WhatsApp, that's where the tool goes; if the data can't leave the building, the AI runs inside it.

What has genuinely changed is speed. A brief on Monday can be something people are using the following week. We start every build on an AI-assisted pipeline with our own standards and parts, and a person takes over the moment the machine reaches the edge of what it does well. Weeks instead of quarters isn't a slogan. It's the difference between a business waiting a year for a system and a business using one before the next month-end.

But let me be straight about the part that matters most, because it's where I agree completely with my colleagues on the change side of CCG. The build is the easy half. The real work starts the day you ship, when you meet the exceptions, the ownership questions, and the person quietly keeping the old spreadsheet going just in case. That is why we stay on after go-live rather than handing over and disappearing, and why the technology and the change discipline sit under one roof here.

Find the inefficiency, build the fix, make it stick, and then go and find the next one.

If you would like help with Doing Change Better, we would welcome the conversation. Get in touch with us.

