

Quoting rebuilt — and the order system out of the *owner's head*

A UK wholesale plant nursery where quoting a customer meant a fifteen-minute juggle between a website and a spreadsheet, and where an order was a folder on a shared drive. Fixed without a line of custom software — by choosing the right tools, connecting them properly, and running them alongside the old way until the old way was not needed.

SECTOR	Wholesale horticulture — a UK plant nursery supplying the trade
ROLE	Business strategist, working inside the business rather than advising it
REPLACED	Website-and-spreadsheet quoting · orders kept as folders in Documents across shared drives
APPROACH	No bespoke application — off-the-shelf products chosen, tailored and connected
STACK	WooCommerce · Pipedrive · courier portals · Zapier · Make · custom code integrations
RESULT	£1m to £3.2m in two years, with the office unchanged at three; margin from 5% to 17.6%
TIMELINE	The quoting, CRM and order management rebuild implemented inside the first 3–4 months
ENGAGEMENT	Two years in the business — the systems first, then running operations and fulfilment on them

01 Executive summary

This one is different from the other case studies in this set. It was not a project delivered from outside a business — it was two years spent inside one, doing the jobs before rebuilding them. **The rebuild itself took three to four months of that.**

The nursery sells plants wholesale: a catalogue that runs to roughly ten thousand lines, customers who ring up and ask for a price on a list of species and sizes, and an order that then has to be picked, loaded and delivered. When the work started there were four people and about £1m a year going through the business. Two years later there were fifteen people and £3.2m — and the margin had moved from a near-break-even 5% to 17.6%.

Where those extra people went matters. They went outside — onto the nursery and the logistics side, to physically handle the volume. **The office stayed at three.** The same three people took a business from £1m to £3.2m, which is only possible if the administrative work per order collapses.

Two bottlenecks did most of that. The first was quoting: a price on the phone meant searching the website for what was in stock, cross-checking a spreadsheet for what it should cost, then writing the quote and emailing it — about fifteen minutes, and the whole thing again if the customer changed their mind. The second was subtler and more dangerous: the order system had no system at all. An order was a folder in Documents, sitting on a shared drive across the laptops — and everything the folders did not hold, which was most of what mattered, sat with the owner. Nobody else could run a day without asking him.

There was no visibility either: nothing tracked what had been quoted to whom, beyond whatever each salesperson kept by hand about their own leads.

None of it was fixed by building software. The quoting method, the CRM and the order management system were all introduced — chosen from what already exists rather than commissioned. The only thing already in place was the website, and the advice there was to populate it with every product line and its price. The fix was the right tools, configured properly, and connected — so a salesperson quotes directly from the website's catalogue inside the order system's back end, the customer and the quote land in the CRM, follow-ups send themselves, and a quote that gets paid populates the order flow with nothing typed twice — three separate products behaving as one seamless system.

All of it went in inside the first three to four months. The remaining two years were spent running operations and fulfilment on top of what had been built — which is the part that matters when you are judging whether a system actually works. It is easy to install something and leave; living with it for two years, in the seat, is what proves it.

Costed at the wages the business actually pays, standing still would have run to roughly **£107,000 a year** — an office hire it would have had to make, and the peak-season enquiries it could not have quoted. Section 12 sets out the arithmetic.

Quoting was taken first because it feeds everything else; from there the work moved downstream through fulfilment, every stage owned by a role. Quoting now takes two to three minutes, on the call — and the heaviest work was taken out of the day rather than shared around, so the business could carry far more volume without quality slipping.

15 → 2–3

minutes to build a quote

675 hrs

of sales time back on the
phones, a year

3–4 mths

to implement all of it, start
to finish

£1m → £3.2m

in two years, with the same
three people in the office

02 Fifteen minutes for one price

A wholesale customer does not ask for one plant. They ask for a list — species, sizes, quantities — and they want a number before they get off the phone.

Answering that meant working through the list twice: once against the website to see what was actually available, and again against a spreadsheet to find what each line should be priced at. Then the quote was written up and emailed. About fifteen minutes of a salesperson's time, per quote, with the customer either waiting or being rung back later — and if they changed a size or a quantity, most of the work happened again from the top.

The cost was not only the time. A quote that takes fifteen minutes gets rationed: customers get rung back instead of answered, small enquiries feel like they are not worth doing, and a salesperson spends the day on clerical work instead of on the phone selling.

THE SECOND COST, AND THE WORSE ONE

Every price came out of a spreadsheet or out of somebody's memory. Across ten thousand product lines, no salesperson can hold that — so what a customer was charged depended on who picked up the phone, what they remembered, and how much they wanted to close the sale. **Underpricing was not a risk; it was a certainty.**

And there was a third cost at the other end of the chain. A quote the customer accepted and paid for had to be re-entered by hand, line by line — the same information, a second time, with all the transcription errors that implies.

Underneath all of it, nothing was tracked. There was no record of what had been quoted, to whom, for how much, or whether anyone had followed it up — beyond whatever each salesperson kept by hand about their own leads. **A business cannot manage what it cannot see:** no pipeline, no conversion rate, no list of quotes going cold, and no way to know whether a quiet month was a demand problem or a follow-up problem.

03 The website became the price book

The unlock was not a piece of software. It was a piece of advice: put every product line on the website, with its price.

Once the whole catalogue existed in one place and was priced, it stopped being a shop window and became the single source of truth. Those prices are the rules — and they were embedded into the quoting system, so a quote is priced from the catalogue rather than from whatever the person building it happens to remember.

WHAT THAT PROTECTS

A salesperson no longer needs to know what ten thousand product lines cost — which is not a reasonable thing to ask of anyone — and, just as importantly, **they cannot underprice**. The number comes from the catalogue, not from memory or from a judgement made under pressure with a customer waiting on the phone.

That is where the margin moved. A price rise reaches every quote the moment it reaches the website, instead of waiting to reach every salesperson's memory — and the difference between what a strong day and a weak day gets charged for the same plant simply disappears.

04 Nothing was built — everything was chosen

Three things went into this business: a quoting method, a CRM, and an order management system. Not one of them was written from scratch.

The website was already there — a WooCommerce store — and once it held every product line at its price it became the thing everything else could quote from. The quoting method was a plug-in: the right one, found and configured, so a salesperson prices straight off that catalogue. Pipedrive came in as the CRM and was then tailored and built out until it ran the orders too, which is not what it is sold as but is what it turned out to be very good at.

Dispatch was pulled in the same way: the external courier portals the business already books through are fed from Pipedrive, so a job going out is booked from the order rather than typed into a courier's website by somebody reading off a screen.

The connecting was done with Zapier and Make, plus custom code integrations wherever those could not reach far enough — which is the only code written anywhere in this project, and it is glue rather than product. Pipedrive sits in the middle of all of it as the core CRM and order management tool; everything else is fed from there.

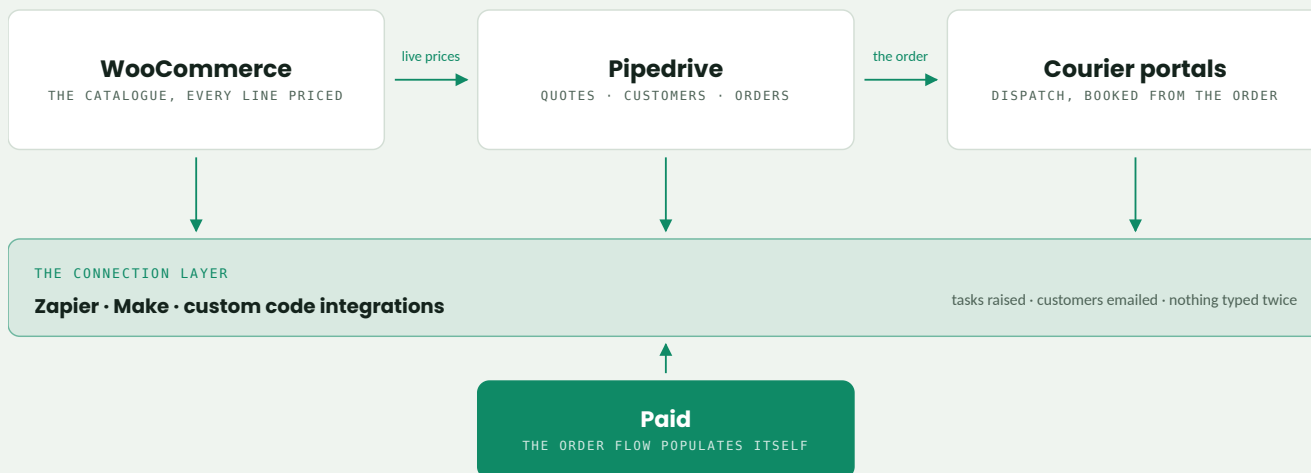
Then all of it was connected, so that what had been three separate products behaves like **one seamless system**: a price on the website becomes a quote, a quote becomes a customer record, a paid quote becomes an order, and the order moves through fulfilment — without anyone carrying information from one screen to another.

NO BESPOKE APPLICATION — EXISTING PRODUCTS, CONNECTED UNTIL THEY BEHAVE AS ONE SYSTEM

ALREADY IN PLACE

INTRODUCED, THEN TAILORED INTO ORDER MANAGEMENT

EXTERNAL, FED BY PIPEDRIVE

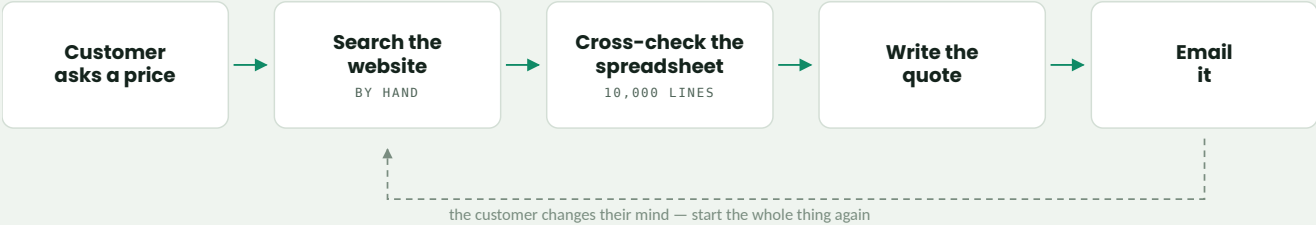


No bespoke application. An existing WooCommerce catalogue, Pipedrive brought in and tailored into order management, the courier portals fed from it, and a connection layer of Zapier, Make and custom integration code holding it all together.

A salesperson now quotes directly from the website's catalogue inside the order system's back end. The customer's details and the quote save into the CRM together. Quotation follow-ups send themselves. And when a quote is paid, it populates the order flow — the same lines, the same prices, nothing re-keyed.

05 What that removed

BEFORE one price, on the phone — about fifteen minutes



AFTER two to three minutes of a person, then nobody touches it again

automatic — nobody re-types anything



a change is a product line swapped and re-submitted, not a rebuild

The quote, before and after. A person is involved for the first two steps; the rest of the chain runs on its own.

Two to three minutes rather than fifteen, and an amendment is a product line swapped and re-submitted rather than the whole quote rebuilt. But the larger saving is further down the chain: **the double entry is gone**. A paid quote used to be re-typed into the order system by hand — every line, every price, every customer detail, with all the transcription errors that implies. Now it is the same record moving forward.

WHAT THAT TIME IS WORTH

Doing the arithmetic

In the busy season the sales team can take **fifteen to twenty leads in a day**. At fifteen minutes a quote that is three and three-quarter to five hours of pure quoting before anybody answers a telephone — which is why, under the old method, they simply could not quote them all and take calls in the same day. At two to three minutes it is under an hour. In the quiet period it is eight to ten a day, and the same arithmetic applies at a smaller scale.

Quiet period	8–10 leads a day
Busy season	15–20 leads a day
Averaged across the year	≈ 13 a day
Over roughly 250 working days	≈ 3,250 quotes
Time saved on each — 15 min down to 2–3 min	≈ 12 minutes
Selling time returned, every year	≈ 675 hours

That is the better part of half a full-time role, handed back to the people whose job is to be on the phone rather than in a spreadsheet — and it is only the quoting half. The re-keying that disappeared when a paid quote started populating the order flow by itself is on top of it, as is every hour that used to go on chasing quotes by hand.

WHY THAT IS THE CONSERVATIVE NUMBER

675 hours counts only the minutes saved on quotes that were being written anyway. It does not count the enquiries that never got quoted at all — and in the busiest weeks of the year, when fifteen to twenty leads a day was arithmetically more than a day of quoting, **there were always some**. Those were not admin hours lost. They were revenue.

Nor does it count the second-order effect. When a quote costs three minutes instead of fifteen, small enquiries stop being quietly rationed — the ones that are not obviously worth the effort get answered too, and some of them turn out to be worth having.

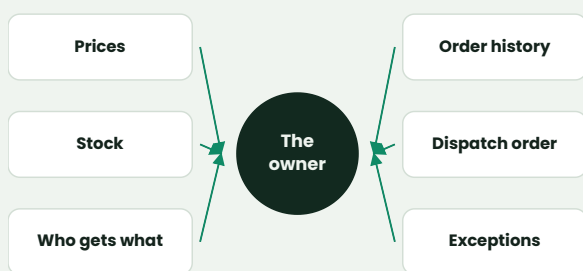
06

Out of folders, out of the owner's head

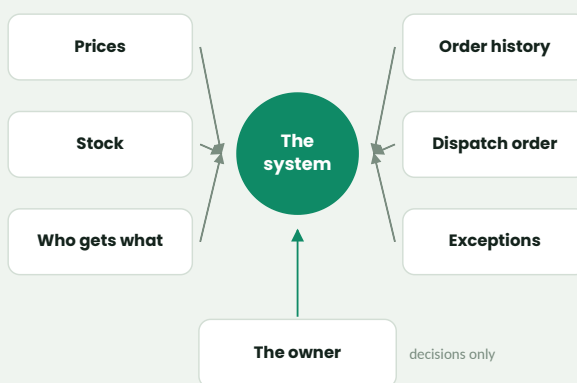
The quoting problem was the visible one. The order side was the one that would have stopped the business growing — because there was no order system at all.

An order was a folder in Documents, on a shared drive, across the laptops. That is not a system; it is a filing habit, and it only works while the number of orders is small enough for people to remember which folder holds what. Everything the folders did not hold — what a particular customer pays, what is really in stock as opposed to what the list says, who gets the good material when there is not enough, what went wrong last time, the order things go out in — lived in one person's memory. That works at four people. At fifteen it means every day queues behind one man, and the business cannot be away from him.

BEFORE



AFTER



Every question used to route through the owner. Now the answers live in the system, and he is consulted for decisions rather than for facts.

Getting it out of his head was mostly not a technical job. It was sitting in the seat, watching which questions actually got asked, and writing the answers down somewhere the rest of the team could reach them — then making the system the place those answers live, so they stay current without anybody maintaining a document.

The same was true at the front of the chain: nothing tracked a quote once it had gone out. A salesperson's own handwritten list of who they had quoted was the whole of the sales record, so the business had no pipeline it could look at and no idea which quotes were going cold.

The place all of it lives now is the CRM. It came out of the box as a sales tool, and it was tailored and built out from there into something the business runs its orders on: stages that mirror how work actually moves, each one owned by a role, each one naming the action that role has to take next. **Nobody has to ask what happens now — the system says so.**

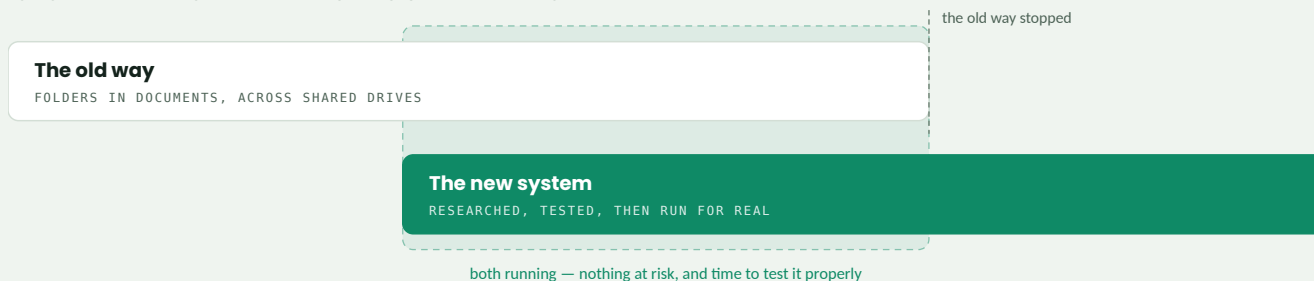
07

Chosen, tested, and run alongside

There was no need to invent an order management system. There are good ones already built. The work was finding out which, proving it on this business's real orders, and getting it in without stopping the business for a day.

So: research what exists, shortlist, and test the candidates against the way this nursery actually works rather than against a feature list. Then put the chosen one in **while the old way carried on running** — the same orders going through both, so nothing was at risk and the new system could be judged on real work rather than on a demo. Only once it had earned it was the old way stopped.

NO DOWNTIME — THE OLD WAY KEPT RUNNING UNTIL THE NEW ONE HAD EARNED IT



Research and select › test › run both › retire the old way

The old way was never switched off and hoped for. It ran until the new system had proved itself on the same orders.

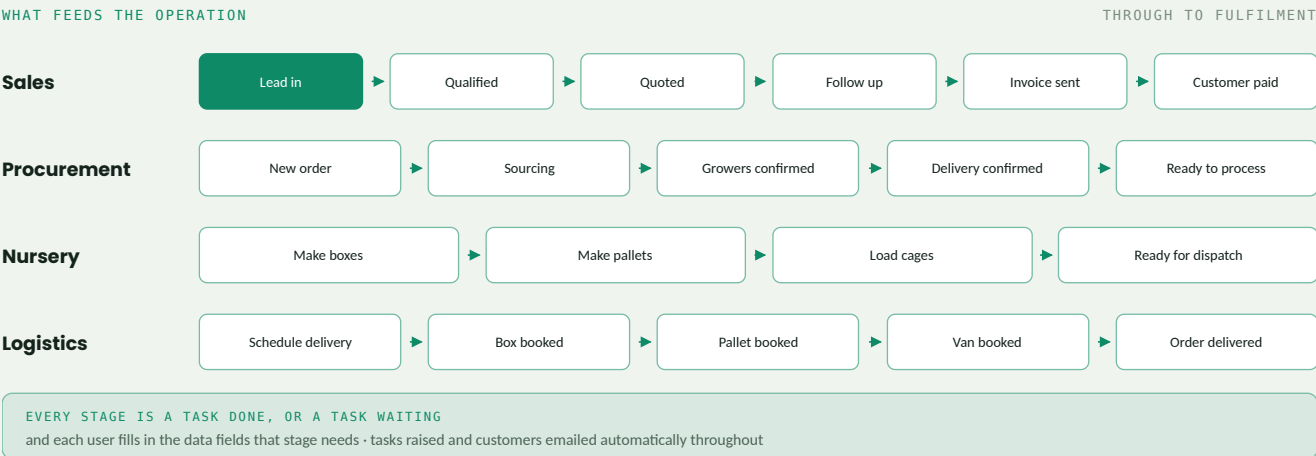
WHY THIS MATTERS MORE THAN IT SOUNDS

Most operational software fails at the changeover, not at the feature list. A business that cannot afford a bad week will not risk one — so it stays on the folders and the spreadsheet forever. **Running both removes the risk that stops most firms ever starting.**

08 Start at what feeds the operation

The order the work was taken in was not arbitrary. Start with what feeds the business — the quote — then follow the same job downstream, stage by stage, to the point it is delivered and closed.

Five roles touch a job on the way through: the sales team, the order processors, procurement, the nursery team and logistics. Each has its own pipeline, and each sees their own stages and nothing else.



The pipelines, condensed. Each role's stages, in order, on one system — with each person seeing only their own. The real pipelines carry more stages than are shown here.

Every stage is either a task that has been completed or a task waiting to be done — that is the whole grammar of the system, and it is why nobody has to be told what happens next. As a person moves a job along their pipeline, they fill in the data fields that stage needs, so the record arrives at the next role already carrying what that role is about to ask for.

WHY THIS ORDER, AND WHY IT FITS

Fixing the front of the chain first is what makes the rest of it possible: everything downstream is working from a clean, priced, unambiguous order rather than from something typed out twice. And every seat was done before it was rebuilt, which is what stops a system being an outsider's idea of the work. The people who answer the phone know which questions customers actually ask; the people in the yard know which part of a dispatch goes wrong. A rebuild designed from a whiteboard solves the problems that are easy to describe. A rebuild designed from the seat solves the ones that are actually costing money.

THE PART CUSTOMERS ACTUALLY FEEL

Tasks and customer emails are automated the whole way along. The person who owns the next step is told it is theirs, and the customer hears from the business at every stage without anybody remembering to send anything. **Service stops depending on how busy the office happens to be that day.**

Taking the front of the operation first also means the work pays for itself immediately, and the next piece is funded by the last — nothing here needed a budget approved up front against a plan nobody could check.

DELIBERATELY CONDENSED

The stages shown are a shortened version of the real pipelines. The full set is the business's own operational detail and stays with them — but **the shape is the point, not the specifics**: one pipeline per role, every stage a task, every task with an owner and the data it needs attached to it.

09 The binder, and the courier portal

The single heaviest piece of admin in the business was not the quoting. It was getting a day's dispatch into the couriers' platforms.

The outside team used to work from paper. The office printed a written list of the day's jobs, put the invoices in a binder, and handed it over. The team worked down the list, and then the binder came back to a desk — where **an office person typed every single consignment into the courier's platform by hand**. Order details, addresses, box and pallet counts, one entry at a time, at roughly six or seven minutes each.

On a day dispatching thirty to forty pallets, that is most of an office person's afternoon spent re-entering information the business already had — twice, if you count writing the list in the first place.

Pallets on a busy dispatch day	30–40
Manual entry into the courier platform	6–7 min each
Office time, on a busy day	≈ 3–4½ hours
Office time on it now	None

What replaced it

The order management system went onto the outside team's phones. They open the app and see the day's workload immediately — no list, no binder, nothing printed. They record the number of boxes and pallets against the order as they make them up, move it to **ready for dispatch**, and the connection into the courier portals uploads the consignment in seconds.

It happens at the moment the outside team processes the order, done by the person who actually handled the goods. Nobody carries paper back to a desk, and nobody re-types anything — which also removes the errors that come from reading someone else's handwriting into a courier's address field at four o'clock.

10 Before and after

QUOTING AND PRICING

	BEFORE	AFTER
Pricing a customer's list	Search the website, then cross-check a spreadsheet, then write it up	Quoted straight from the website catalogue , inside the order system
Time per quote	About fifteen minutes	Two to three minutes , on the call
A day of leads in peak season	15–20 enquiries — unquotable in a day while also answering the phone	All of them, the same day , in under an hour of quoting
A customer changing their mind	Most of the work again from the top	A few clicks — the quote is live, not a typed document
Knowing what is in stock	Check the website and hope	In front of the person quoting as they build the list
Where prices lived	A spreadsheet, and the owner's memory	The website — every product line, priced, and the rule the quote follows
Risk of underpricing	Whatever the salesperson remembered or judged on the call	Not possible — the price comes from the catalogue, not from memory
Applying a price change	Tell everyone and hope it sticks	Change it on the website ; every quote from that moment follows
Tracking what was quoted	A salesperson's own handwritten note of their leads	Every quote on the board — value, stage, age and whose it is
Knowing the pipeline	Nothing to look at	Live — what is out, what is going cold, what converted

ORDERS, SERVICE AND THE BUSINESS

	BEFORE	AFTER
Where an order lived	A folder in Documents, on a shared drive	One system , with a stage, an owner and a next action
A paid quote becoming an order	Re-typed by hand into the order system	It populates itself — same lines, same prices, nothing re-keyed
Knowing whose job is next	Ask, or assume	The task appears for the role that owns it
What an employee sees	Whatever folder they could find	Their role, and nothing else — set up per user, per role
Chasing a quote	Remember to, or don't	The CRM follows up on its own
Keeping the customer informed	Whoever remembered, when they had a minute	Emailed at every stage , automatically, all the way to delivery
Getting the day's work to the outside team	A written list and the invoices, printed, in a binder	On their phones — the day's workload, visible immediately
Recording boxes and pallets	Written down, then re-entered at a desk	Entered once , by the person who made them up
Booking a courier	An office person typing every consignment in by hand, 6–7 minutes each	Uploaded in seconds the moment the order is processed
Putting the new system in	—	Run alongside the old way until it had proved itself; no downtime
Running a day without the owner	Not really possible	Normal — the answers are in the system, not in his head
Adding a salesperson	Months of learning the prices and the customers	They quote from day one against the same information
Small enquiries	Quietly rationed — not worth fifteen minutes	Worth answering , because answering costs three minutes

11 Value delivered

TASK	BEFORE	AFTER	RETURNED
Building a quote	~15 min per quote	2–3 min, on the call	≈ 675 hrs/year
Re-quoting a change	Most of it again	Swap a line, re-submit	Every revision
Turning a paid quote into an order	Re-typed by hand	It populates itself	Every order
Chasing quotes	Manual, or not at all	Automatic from the CRM	Every quote
Answering "what does X cost?"	Ask the owner	It is on the screen	Constant
Training a new salesperson	Months of absorbed knowledge	Quoting from day one	Every hire

WHAT ACTUALLY CHANGED

The business stopped queueing behind one person.

A firm can survive with the operating knowledge in one head at four people. At fifteen it cannot — every day waits on him, no one else can cover, and growth stops at the limit of what one person can hold. **Writing that knowledge into a system is what made the next ten people, and three times the order volume, possible.**

- **Selling time went back to selling.** Roughly 675 hours a year moved out of clerical assembly and back onto the phones — and small enquiries stopped being quietly rationed because they were not worth the effort.
- **Margin stopped depending on who quoted.** Once every line was on the website at its price, that became the rule the quoting system follows — so nobody can underprice, and nobody has to carry ten thousand prices in their head. One price list applied the same way by everyone is the difference between 5% and 17.6% far more than any single price rise is.
- **Service stopped being a function of how busy the office was.** Tasks are raised for the person who owns the next step and customers are emailed at every stage automatically — so nobody falls through a gap on a busy week, and the customer always knows where their order is.
- **Dispatch stopped costing an afternoon.** On a thirty-to-forty pallet day, typing consignments into the courier platforms by hand took three to four and a half hours of office time. That is now zero — the consignment uploads itself when the outside team processes the order on their phone.
- **The double entry disappeared.** A paid quote used to be re-typed into the order system line by line. Now it is the same record moving forward — hours of admin, and a whole class of transcription error, simply gone.

VALUE DELIVERED, CONTINUED

- **The sales pipeline became visible for the first time.** Quotes stopped living in individual salespeople's notebooks. What is out, what it is worth, what is going cold and what converts are now facts the business can look at rather than impressions.
- **The changeover cost nothing.** The new system ran alongside the old one on the same orders until it had earned the switch, so the business never had a bad week paying for the improvement.
- **The business became transferable.** Knowledge in a system can be handed to a new hire, checked, corrected and improved. Knowledge in a head cannot — and because each role's screen holds only that role's work, a new person is useful in an afternoon rather than a season.
- **Throughput went up without the office growing.** The heaviest, most time-consuming work was removed rather than distributed. The business went from four people to fifteen, but that hiring went outside — onto the nursery and the logistics side, to handle the physical volume. **The office stayed at three people throughout.** The same three carried more than three times the revenue.
- **And quality held while it happened.** The customer's experience did not degrade under the extra volume, because the chasing and the updates no longer depended on somebody in the office having a spare ten minutes.
- **Growth stopped being capped by the owner.** £1m to £3.2m, without the person who knew everything becoming the reason it could not go further.

THE TEST OF ALL OF IT

Fifteen to twenty leads in a day, all quoted the same day.

That is what peak season looks like now, handled by the same three people in the office who were there at £1m. Under the old method it was arithmetically impossible — fifteen minutes each is the better part of a working day of pure quoting, before anyone picks up a telephone. **The enquiries that used to go unanswered in the busiest weeks of the year are the ones the business was most likely to win.**

12

What it is worth, in money

Two of the savings in this document are measurable from figures the business already knows. Costed at the wages actually being paid, they come to roughly fifteen thousand pounds a year on their own.

THE RATES USED

Outside team	£14 / hour
Procurement	£15 / hour
Sales and office	£15 / hour — assumed

THE BREAKDOWN

SAVING	THE ARITHMETIC	HOURS / YEAR	AT THE RATE
Quoting 15 min down to 2-3 min	3,250 quotes × ≈12 min saved	≈ 675	≈ £10,125
Courier consignment entry manual, then none at all	80 busy dispatch days × 35 consignments × 6.5 min	≈ 300	≈ £4,500
Office and sales time returned	from these two line items alone	≈ 975	≈ £14,600

Quote volume is derived on page 07; the consignment figures on page 12. The only assumptions here are the office hourly rate and eighty busy dispatch days a year — change either and the rest follows.

WHAT IT IS WORTH, CONTINUED

What that figure deliberately leaves out

Everything above is time that can be counted from volumes the business already tracks. Several of the larger savings cannot be, and none of them are in the total:

- **Re-entering a paid quote as an order.** Every won job used to be typed in a second time, line by line. That is now zero, and it happens on every order the business takes.
- **Chasing quotes by hand**, and remembering to — now automatic from the CRM.
- **Keeping customers informed.** Emails at every stage that somebody used to have to write, on top of their actual job.
- **Procurement and the nursery team** working from an order that arrives already complete, instead of chasing the office for what it should have said.
- **Errors.** Mistyped addresses, wrong pallet counts, a consignment booked against the wrong order — each one a phone call, a re-delivery, or a customer who does not come back.
- **The enquiries that never got quoted at all** in the weeks when fifteen to twenty leads a day was more than a day of quoting. That is not a cost line. It is lost revenue.

THE COMPARISON THAT MATTERS

≈ £14,600 a year, recurring, from two line items — against an implementation that took three to four months and no bespoke software at all. Whatever the licences and the setup came to, it is not close.

Figures are illustrative and built from the volumes and rates given by the business. They are meant to be checked against its own numbers, not taken on trust.

WHAT IT IS WORTH, CONTINUED

The other side of it — what standing still would have cost

The savings above are one way of reading the numbers. The more useful way, for anyone deciding whether to do this, is the opposite one: **what would this business have had to spend to reach £3.2m on the old processes?**

One: it would have had to hire

At today's volume, run the old way, the two measured tasks alone consume this much office time:

	OLD METHOD	NOW	DIFFERENCE
Quoting — 3,250 a year	≈ 812 hrs	≈ 135 hrs	677 hrs
Courier consignment entry	≈ 300 hrs	0	300 hrs
Extra office hours needed	≈ 1,112 hrs	≈ 135 hrs	≈ 975 hrs

Nine hundred and seventy-five hours is **the better part of half a full-time office role** — and that is from two tasks. Add re-entering every paid quote as an order, chasing quotes by hand and writing the customer updates, and it goes past a whole person.

≈ 975 hours at £15 an hour	≈ £14,600 in wages
Employer NI, pension, holiday cover	+ ≈ 18%
The hire that did not have to happen	≈ £17,000 a year

These are the same hours as the saving on the previous page, seen from the other direction — they do not add together. But the avoided cost is the larger figure, because a saved hour costs a wage while a hired hour costs a wage plus everything on top of it.

WHAT IT IS WORTH, CONTINUED

Two: and it would still have lost work

Hiring solves the hours. It does not solve the arithmetic of a peak-season day. Fifteen to twenty leads at fifteen minutes each is three and three-quarter to five hours of pure quoting — and a salesperson also has to answer the telephone, deal with the orders already sold, and go home. **Some proportion of those leads simply never got quoted**, in exactly the weeks of the year when demand was highest.

That is not an admin cost. It is revenue the business never competed for. What it comes to depends on two numbers only the business knows:

Enquiries a day left unquoted at peak	4 — illustrative
Peak days in a year	≈ 60
Enquiries never quoted	≈ 240
Conversion rate	25% — illustrative
Average order value	£1,500 — illustrative
Revenue never competed for, a year	≈ £90,000

The three marked figures are placeholders — replace them with the business's own and the answer moves. The point does not: at a tenth of that number it is still several times the admin saving, and unlike the admin saving it compounds, because an unquoted customer buys from somebody else and stays there.

WHAT IT IS WORTH, CONTINUED

THE TWO ADDED TOGETHER

THE COST OF STANDING STILL EVERY YEAR

The office hire that would have been needed ≈ £17,000

Revenue never competed for at peak ≈ £90,000

Not making the change

≈ £107,000 a year

The two lines are different kinds of number and are worth reading as such: the first is a cost avoided, the second is revenue never earned — at this business's margin that is nearer £16,000 of profit in the year it happens. It is also the line that compounds, because a customer quoted by somebody else does not come back the following season.

AGAINST WHAT

Three to four months of implementation, no bespoke software, and products the market already sells. **Set against roughly £107,000 a year of standing still** — and the office carried three times the revenue with the same three people, which is the same fact stated a third way.

13

What this one proves

The other case studies in this set are software built from scratch. This one is the opposite, and it matters more than it looks: **the right answer here was not to build anything.**

Writing a bespoke quoting tool and a bespoke order system would have been the more impressive-sounding engagement, and it would have been the wrong one — slower, more expensive, and two more systems to maintain. What it actually needed was one piece of advice about the catalogue, the right plug-in, an honest search through what already exists, and the discipline to test the result on real orders before trusting it. The only code written was integration glue, where Zapier and Make could not reach far enough on their own.

That last part is a skill in itself. Choosing well from what is already on the market, and migrating onto it without giving the business a bad week, is not a lesser job than building something — it is usually the one that actually gets adopted.

Most people who build operations software have never run the operation. They arrive, ask what the process is, are told the tidy version of it, and build that — which is why so much trade software is technically fine and practically useless. The version people describe in a meeting is never the version they actually do at four o'clock on a Friday with a customer waiting.

Doing every seat before rebuilding it is where that gap gets learned. It is also where the commercial instinct comes from: knowing which bottleneck is worth attacking first is a business judgement, not a technical one, and getting it wrong means building something elegant that saves nobody any money.

The work itself was quick — three to four months to get the quoting method, the CRM and the order management system in and connected. What came after was the unusual part: two years running the operation on them, in the seat, watching every place they were awkward and fixing it. **Most people who install a system never have to live with it.**

THE PATTERN, IN ONE LINE

Do the job. Find where the hours and the margin actually leak. Fix it with the smallest thing that works — often something you already own. Then do the next one.

No app

bespoke — existing products, chosen and connected

3–4 mths

to build it — then two years living with it

3 in the office

unchanged, through 3× the revenue

14

Could your operation use this?

Two questions worth asking about your own business: how long does it take to give a customer a price, and what happens to the week if one particular person is away?

If the answers are "too long" and "not much gets done", the problem is the same one this nursery had. The knowledge that runs the business is in the wrong place — in a spreadsheet nobody else can read, or in a head nobody else can open — and everything queues behind it.

For this nursery the answer, costed at its own wages, was roughly **£107,000 a year** — a hire it would have had to make, and the peak-season work it could not have quoted for. It fixed both in three to four months without commissioning a single piece of software.

Getting it out is not usually a big software project. Very often it is not a software project at all — it is the tools you are already paying for, configured properly and connected to each other. The first question worth answering is not "what should we build?" but "**what have we already got that isn't switched on?**"

WHERE IT USUALLY STARTS

An operations audit. A walk through your working week, a map of every place a person is being used as a database, and a costed list of what could be moved into a system — with the numbers attached, so you can decide whether any of it is worth building.

byHoratio

AI & operations software for real-world businesses

horatio@byhoratio.com

BYHORATIO.COM