RIGHTTACK CONSULTING · ERP SELECTION & SYSTEMS

How to spend two million dollars and still not know what is in your warehouse

A field guide to choosing an ERP system, written from the operations chair — the one you have to sit in for the three years after the vendor has been paid and gone home.

There is a moment in the life of a growing company when the spreadsheets stop being a workaround and start being the business. Inventory lives in one person's head and three tabs. Nobody can say what is on hand without walking to the rack. The month-end close is an act of archaeology. Somebody says the word "ERP," everyone nods, and a very expensive clock starts ticking. This is a field guide to the next eighteen months — so you come out the other side with a system you run, rather than one that runs you.

Enterprise software is sold the way weddings are sold: on the day, not the marriage. The demo is bright, the salesperson is charming, and the number on the slide is a fraction of what you will actually spend. Almost none of the failure happens in the software. It happens in the decisions you make before and around it — what you buy, why, how much you bend it, and whether anyone checked that it can produce the one report you needed all along. Get those right and a mediocre system serves you well. Get them wrong and the best system on the market becomes a two-million-dollar way of not knowing what is in your warehouse.

First, make sure it is actually you who needs this

Before you shop, work out whose problem you are solving. Often the pressure for ERP comes from finance — the CFO wants a clean close, auditable numbers, one ledger. That is a legitimate need. It is also a different need from yours, and a system chosen to make the close elegant can quietly make the shop floor miserable. Operations lives in inventory accuracy, receiving, work orders, lead times and what to build next. Finance lives in the general ledger. The same three letters mean two different machines depending on who is describing them.

So say out loud, early, whose requirements win where they conflict — because they will conflict. If you cannot articulate the operational problem in a sentence a warehouse supervisor would recognise, you are not ready to buy. You are ready to keep talking.

The beauty pageant

Then comes the parade. Three or four vendors, each with a demo environment built by people whose entire job is to make the software look inevitable. The data is clean because they cleaned it. The workflow is smooth because they scripted it. Nobody in that demo has ever had a receiving clerk scan the wrong lot at 4pm on a Friday.

> **The demo is not the product. It is a movie of the product, directed by someone who wants your money.**

You cannot un-see a good demo, so protect yourself before you watch one. Write down your hardest real-world cases first — the split shipment, the partial return, the engineering change mid-build, the product with forty options — and make each vendor run *your* ugly transactions in front of you, not their pretty ones. And treat "been there, done that" as a feature: a system your people have run before starts with a base of tribal knowledge you do not have to buy twice. Familiarity is worth real money on a implementation, and it never shows up on the comparison grid.

Buy for the company you are, not the one in the pitch

Every vendor will size you up and sell you up. The tier above always has more modules, more scale, more future. And the tier above is where implementations go to sink, because you inherit the complexity of a company ten times your size while still having the staff and the discipline of the company you actually are.

> **Buy the ERP for the company you are today, not the one in your five-year plan.**

You can grow into a system, or grow out of one, and the second is cheaper than being crushed by the first on day one. Match the tool to your current transaction volume, your current headcount, your current process maturity. A system you can fully stand up and actually use beats a more powerful one you can only half-implement — because a half-implemented ERP is not a partial win. It is a new set of spreadsheets, with a licence fee attached.

Configure. Do not customise.

This is where the money leaks, quietly, for years. Every ERP arrives with a set of standard processes baked into it — a normal way to receive, to pick, to invoice, to close. Your team will look at each one and say, but we do it differently. Sometimes that difference is your competitive edge and worth protecting. Almost always it is just habit, and you are about to pay a developer to enshrine a habit in code.

> **You are not competing on your accounts-payable process. Do not pay to customise it.**

Ask of every requested customisation a single question: does a customer pay us more because we do this our way? If the honest answer is no — and for AP, receiving, GL and most of the back office it is no — then change your process to match the software, not the software to match your process. Customisation is not just the build cost. It is the upgrade you can never take, the consultant you can never fire, the reason version 12 is a project instead of a patch. Turn the clever advanced features off, too, until the boring ones are running clean. You can switch capability on later. You cannot un-break a go-live that tried to do everything at once.

The budget is some function of pi

Take the number the vendor gave you. It covers licences and a tidy implementation, and it is fiction — not because they lied, but because their number cannot include your data, your customisations, your backfill, your overtime, or the four months you will run two systems in parallel while nobody trusts the new one yet.

> THE ONLY BUDGETING FORMULA THAT HAS EVER HELD
>
> Take the vendor's all-in number. Multiply by pi. *That* is your budget.
>
> Software is the cheap part. The expensive parts are your own people's time, the data, and the double-running. None of them are on the quote.

Budget the internal cost most of all — the best people you have will be pulled onto this, and they still have day jobs. Software that is under-resourced on the customer side fails at a rate the vendor is always happy to blame on the customer. They are not wrong. Fund it as the operational project it is, or do not start it.

Data migration is where projects go to die

Here is the part everyone underestimates and nobody enjoys. Your data is worse than you think. The item master has duplicates, dead SKUs, wrong units of measure and costs nobody has updated since a previous system. Pour that into a new ERP and you have paid two million dollars to be precisely as confused as before, faster.

Migration is not an IT task to be handed down. It is an operational reckoning — the first honest inventory of your own business you may ever have taken — and it belongs to operations, not to the implementer. Decide deliberately what comes across and what dies in the old system. Clean the item master before it moves, not after. And do not let anyone tell you the tool will sort it out; the tool will faithfully carry your mess across the river and set it down on the far bank.

> **Type the first purchase orders in by hand. You will learn more about your own business in a week than in the whole selection.**

Somewhere in a mid-tier ERP implementation there is a person who chose to key the opening transactions manually rather than trust an import — and came out understanding the data model, the failure points and the real state of the inventory better than any consultant on the project. The slow way is sometimes the fast way. This is one of those times.

You always go live halfway through

You will not be ready. Nobody ever is. There is no clean Friday where the old world ends and the new one begins fully formed on Monday; you go live with known gaps, a stack of workarounds and a team that is tired before it starts. Plan for that instead of pretending it away.

Which means budgeting for the part nobody puts on the plan: hypercare. The weeks after go-live where transactions are wrong, people are slow, and the shop floor quietly reverts to the spreadsheet unless someone is standing there stopping them. Staff that period deliberately — a war room, daily triage, the power users freed from their day jobs to hold hands — because this is exactly when a system is abandoned. Not because it failed, but because nobody was resourced to get it over the first four ugly weeks. Cut over a fiscal boundary if you possibly can, keep the old system readable but read-only, and accept that a productivity dip is the price of admission. A dip you planned for is a schedule. A dip you did not is a crisis.

Test the reporting before you sign, not after

The whole point of this was to know things — what is on hand, what is late, what it cost, what to build next. And reporting is invariably the last thing anyone checks, discovered months in, when someone finally asks for the one number the whole exercise was meant to deliver and the system cannot produce it without a custom report and another invoice.

So test it first. Before you sign, write down the ten reports you actually run the business on, and make the vendor produce every one from their own demo data. If a number you manage by cannot come out cleanly, that is not a reporting gap to fix later. That is the wrong system, and you have just found out while it is still free to walk away.

None of this is really about the software. Choosing well, sizing to who you are, refusing the vanity customisations, funding the internal cost, owning the migration, surviving the go-live, proving the reports — every one of those is an operational decision made under a sales process engineered to rush it. The system is a tool. Whether it leaves you knowing what is in your warehouse, or two million dollars poorer and still walking to the rack to check, is decided almost entirely off the screen.

And notice what all of it has in common. Not one of these is a box you tick at contract signing. Each is a discipline you have to hold across eighteen months, inside a live business that will not stop shipping while you install its nervous system — usually while the vendor, the CFO and the calendar are all pulling the other way. Knowing the pitfalls is the easy half. Steering the thing between them, in the right order, on the resourcing you actually have, is the work.

That is the work I do.

WHERE THIS GOES NEXT

Install the people before the tool →

The planning, visibility and risk method — the operating discipline an ERP is supposed to serve, and cannot replace. Or start a conversation: rtackconsulting.com/contact