RIGHT TACK CONSULTING · INSIGHT

You will support it until you decide not to.

Every machine you have ever shipped carries an obligation that outlives the product. Most businesses never decide what that obligation is — they discover it, one telephone call at a time, when a part goes unavailable. The alternative is to decide once, publish it, and put a clock on it.

Somebody calls about a machine you sold in 1998. They want a board repaired. The part has not been made for nine years, your supplier stopped making the chip on it before that, and the engineer who knew the product retired. In the next ninety seconds someone in your business will decide what you owe this customer. If there is no policy, that decision is made by whoever picked up the phone, and it will be made differently the next time, by someone else, for a customer of a different size.

That is the actual failure. Not that the part is unavailable — that was inevitable and it is not a scandal. The failure is that the commitment was never decided, so it gets improvised under pressure, at the moment when the business has least room and the customer has most grievance. And because it is improvised it is also inconsistent, which is the thing customers remember and repeat to each other.

The support commitment is a design decision. It should be made once, in daylight, by people who are not currently being shouted at, and it should be written down in a form that has dates in it.

What a phase actually is

The unit of the decision is the phase. Not a marketing label — a declared level of support, stated per product, that says what a customer can expect and, more usefully, what they cannot. Four is the right number. Fewer and the middle of a product's life is undefined, which is where most of the installed base sits. More and nobody remembers them.

The mechanism is that the phase is entered on evidence, not on sentiment. The test is whether support is still economically feasible for both sides, and that resolves into four questions with real answers: can the supplier still provide service parts, what do new parts now cost, can field engineers still be trained on it, and how fast is the installed base itself declining through upgrades and natural migration. A product does not move to best-effort support because someone in a meeting felt it was old. It moves because one of those four answers changed.

> Support is not a feeling about a product's age. It is a commitment you can still fund, and the four things that decide whether you can still fund it are all measurable.

Here is the structure, with the clock that runs underneath it.

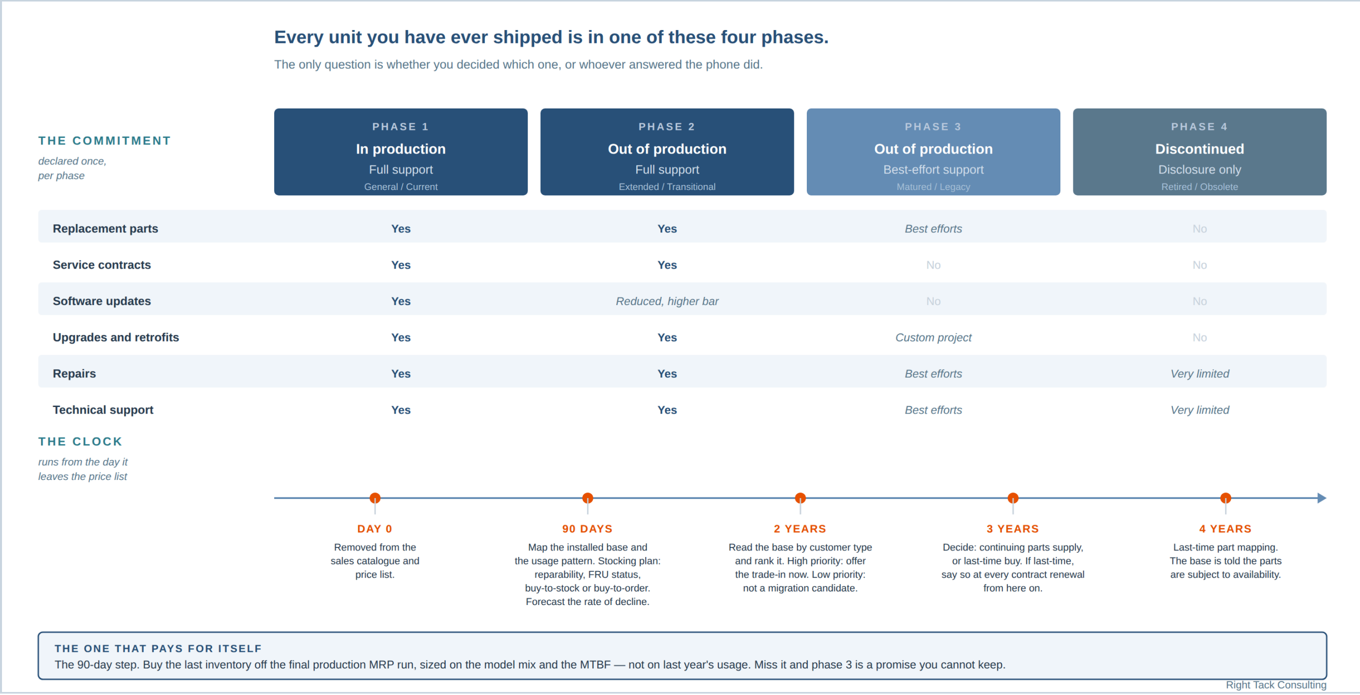

Figure 1. The commitment, declared once per phase, and the clock that runs from the day the product leaves the price list. The rows are what a customer can expect; the value of writing them down is as much in the "No" as in the "Yes". The ninety-day step is the one that funds everything to the right of it.

The threshold moves; the support does not stop

The most useful idea in this whole area is also the least intuitive, and it comes from the software side. When a product leaves current status, you do not stop fixing its bugs. You raise the bar for what counts as worth fixing.

Put a number on it and the policy becomes administrable. If the severity threshold for fixing a defect in a current product is 70, then in the next phase it is 100 — the same scale, a higher bar. Serious problems still get fixed. Cosmetic ones do not. Nobody has to argue about whether the product is "still supported", because that is the wrong question; the question is what severity now qualifies, and that has an answer written down in advance.

Two rules make the rest of it work, and both are worth stating publicly. A product stays in that phase at least as long as the warranty of the last copy you shipped — you cannot sell something on Friday and downgrade its support on Monday. And when a version moves down a phase, every version behind it that already met the criteria moves down too, so the tail does not accumulate silently.

At the bottom of the scale the commitment narrows to something quite stark, and stating it plainly is better than implying more. In the last phase you offer no development, best efforts only from technical support, no guaranteed upgrade path — and disclosure of any safety or data-integrity issue that comes to light. That last clause is not a service level. It is an obligation that does not expire, and it is the one thing you keep doing for a product you no longer support at all.

Free or paid is a policy decision, not a sales decision

Underneath the phases sits a distinction most businesses have never settled, which is why their own people cannot answer it consistently in front of a customer. An *upgrade* replaces the installed version with a newer one — new major version, new catalogue number, and the customer pays. An *update* fixes or improves what is already installed without replacing it — new minor version, and depending on its class it may cost nothing.

Classify the updates and the pricing question answers itself. A safety or data-integrity fix goes out free of charge, always, to everyone, whether or not they have a contract. A critical fix — a serious performance, reliability or usability defect — is published and available, and typically reaches a customer through the support contract they bought. A hotfix addresses one customer's situation, is not fully tested for general release, and is not theirs to redistribute; it also needs a decision, at the moment it is issued, about whether it will be carried forward into future releases, and if it will not, the customer is told so at the time rather than discovering it at the next upgrade.

The commercial consequence of the boundary is the part people miss. If updates arrive through a support contract, then the contract is the product and the update is its content — which means lapsing has to cost something, or the contract is a donation. That is what a break-in-cover rule does: an upgrade is available from one major version to the next, and if a customer is more than one version behind, they buy the current product rather than upgrading into it. Discount it if you like. But the rule has to exist, because without it the rational move for every customer is to let the contract lapse, wait, and buy the upgrade once at the end — and enough of them work that out.

> **Every one of these is a decision about revenue, made once, in a document — or a decision made repeatedly, under pressure, by whoever is on the call.**

The letter you will eventually have to write

The end of the sequence is a letter to people who have your equipment on their floor, telling them it is going out of support. It is the most-avoided document in the aftermarket and the one that most repays being written properly, because it is where the policy becomes real to the only audience that matters.

Three things make it work. Give twelve months, not thirty days: the recipient may have a capital cycle to get through and cannot act on short notice. Say why, honestly and specifically — a third-party component is no longer available from the vendor and the last-time-buy stock is nearly exhausted is a reason a customer can respect, where "product rationalisation" is not. And put named options on the table rather than sympathy: keep running it and accept that repairs may be slow, expensive or impossible; retrofit the machine with current technology; or trade it in. Attach an incentive to the two you would prefer, and give it an expiry.

The letter also has to be honest about the thing nobody likes writing down: if you attempt a repair and cannot complete it, the customer still owes you for the labour. Better on the page in February than discovered in a dispute in November.

The step that pays for itself

If only one thing from this survives contact with your calendar, make it the ninety-day step. When a product comes off the price list, you have one short window in which the last production run is still being planned and the tooling and the supply chain still exist. That is the moment to buy the final inventory — and to size it off the model mix and the mean time between failures across the base, rather than off last year's parts usage, which reflects the machines that happened to break, not the machines that are out there.

Miss that window and every commitment further down the scale becomes a promise you cannot keep. There is no recovering it later at a sensible price: you are then buying obsolete components on the broker market, at whatever the broker asks, to honour a service contract you sold at last year's rate. The businesses that end up here did not decide to; they simply never had a ninety-day step.

Where this is the wrong tool

This is a policy for a business with a large installed base of long-lived capital equipment and a service organisation attached to it. If your product is consumed, replaced on a short cycle, or sold as a subscription where the current version is the only version, most of this is machinery you do not need — the phases collapse into two, and the interesting question moves to renewal rather than to support.

It is also not a substitute for deciding whether the aftermarket is a business you want. The policy tells you what you owe and what it costs. It does not tell you whether service should be a profit centre, what your attach rate ought to be, or whether the installed base is worth harvesting or defending. Those are portfolio questions, they come first, and they are a different paper.

Why there is no method under this one

Every other paper here ends by pointing at a method — the way a client buys help building the structure it describes. This one does not, and the omission is deliberate rather than an oversight.

The temple I use to explain how a business is structured has no pillar for the installed base, and I have said in print that it does not. Order pursuit and after-sales service belong on the second drawing every business needs, the process map, which answers in what order the work happens rather than what capability must exist. Inventing a pillar here to give this paper somewhere to hang would be tidier and less true.

So take this one as it is: a set of decisions that have to be made by somebody, in a business that has been shipping long enough for the first calls to start arriving. Make them before the call, not during it.

WHERE THIS COMES FROM

A corporate support policy, written 2005–06 →

The phases, the clock, the severity threshold and the updates-versus-upgrades standard here are drawn from a global support policy and a software lifecycle standard I wrote and chaired the taskforce for, in a materials-testing instrument business with a large and elderly installed base. Twenty years on, the shape has not needed changing. Start a conversation: rtackconsulting.com/contact