# Standing Up Subsurface AI Capability

Most operators who decide to build subsurface AI in-house frame the decision as a procurement question, which model, which vendor, which platform, and then discover that the model was the easy part. What they were actually deciding was whether to stand up a capability, and a capability is not a deliverable; it is a governed sequence of phases that each retire a specific class of risk before the next begins. This whitepaper sets out the operating model we run for that build-out: four phases, assess, pilot, scale, and operate, mapped onto the four project blocks the engagement is scoped in, each with its own roster of roles, its own exit gate, and its own staffing posture. We argue three things with three instruments. First, that the progression is genuinely governed and not merely sequential: each phase has a distinct staffing shape and a concrete exit test, the pilot against a curve-reconstruction fidelity gate at peak R-squared 0.9891, the operate phase against a monthly compute run-rate of 750 to 1800 EUR, and a gate skipped to save time does not save the risk it was meant to retire but forwards it downstream where it costs more to fix. Second, that the choice between an accelerated sixteen-week track at six full-time staff and one hundred eighty thousand EUR and a standard thirty-two-week track at four staff and one hundred thousand EUR is a concurrency-and-premium decision rather than a scope decision: the fast track does the same work sooner with more people for more money, not less work. Third, that the operate phase is a low standing run-rate whose floor is set by the served model's compute and is unmoved by how well the earlier phases were run, so the temptation to shortcut assess and pilot buys nothing at the stage where the programme actually lives. The deliverable is a way to plan, staff, and govern an in-house subsurface-AI build-out as the phased capability programme it is, rather than as a model handover with a support contract stapled to it.

**The EarthScan Team**
Research, engineering, and field

OPERATING-MODEL / CAPABILITY-BUILDOUT / IN-HOUSE-AI / SUBSURFACE / PHASED-DELIVERY / EXIT-GATES

CONTENTS

> "
>
> An operator that asks us to build subsurface AI in-house is not buying a model. It is standing up a capability, and a capability is built in a sequence of phases that each retire a specific risk. Skip one to save time and the risk does not vanish. It moves downstream and compounds.
>
> "

I

THE FRAME

# A capability is a sequence, not a deliverable

## What the operator is actually deciding

When an operator resolves to bring subsurface AI in-house, the decision usually arrives dressed as a procurement question. Which model, which platform, which vendor, what does it cost. That framing is comfortable because it treats the outcome as an object that can be specified, bought, and installed. It is also the framing that most reliably produces a disappointed operator eighteen months later, because the object was never the point. The model that reads a raster well log and reconstructs its curves is a real artefact, and building it well is hard, but it is the smallest and most transferable part of what the operator set out to acquire. What they set out to acquire was the standing ability to keep doing this after the first model ships, on the next archive, for the next asset, without re-hiring the capability from outside every time. That is not a deliverable. It is a capability, and a capability is built in a sequence.

This whitepaper is about that sequence. It is the operating model we run when the brief is not "give us a model" but "stand up the ability to build and run these models ourselves." We run it in four phases, and we hold to the discipline that each phase has its own roster, its own exit gate, and its own staffing shape, because the alternative, treating the whole build-out as one undifferentiated push toward a demo, is the failure mode we have watched sink otherwise well-funded programmes. The four phases are assess, pilot, scale, and operate, and they map cleanly onto the four project blocks the engagement is scoped and priced in.

The reason to insist on the phase boundaries, rather than letting the work flow continuously from first data look to production service, is that the boundaries are where risk is retired. Each phase exists to answer a question that the next phase's spending depends on. Assess answers whether the data and the problem admit an ML solution at all. Pilot answers whether a model can reach the fidelity the use case needs on a bounded slice. Scale answers whether that model can be turned into a service that clears real volume. Operate

answers whether the running service stays honest over time and inside its budget. A phase that has not answered its question has not earned the right to fund the next one, and a programme that funds the next one anyway is spending on a foundation it has not tested.

## Why in-house at all, and why that changes the shape

Before the sequence, the premise. An operator can rent subsurface AI per project, and for a single bounded task that is often the right call. The case for carrying the capability in-house rests on two things the upstream literature has made concrete: the gains from AI on upstream tasks are large enough to be strategic rather than incremental, and they recur across the asset base rather than resolving in a single engagement [1]. When a capability is strategic and recurring, renting it per project means paying the standing-up cost again on every project and never accumulating the institutional knowledge that makes the second build cheaper than the first. In-house changes the shape of the decision from "buy a result" to "build a machine that produces results," and a machine has to be assembled in an order.

That order is the subject of everything that follows. The rest of this document walks the four phases, shows the roles and the exit gate that belong to each, states the two delivery tracks the phases can be run at and what actually separates them, and then makes the case that the operate phase, the one the programme spends most of its life in, is a modest standing run-rate whose floor is set by the compute and is indifferent to how well the earlier phases were run. That last point is the one that disarms the most common and most expensive temptation in a build-out: the urge to shortcut the early phases to reach the operate phase faster.

II

THE SEQUENCE

# Four phases, each with its own gate

## Assess: the cheapest phase to fail in

The first phase is the one operators most want to skip, because it produces no model and feels like preamble. It is assess, and its entire job is to decide, on a small part-time core, whether the operator's data and problem admit a machine-learning solution, and if they do, to size every later phase against the worst case in the data rather than the average. The core here is deliberately small: a subsurface lead who owns the geological question, an ML architect who owns the modelling question, and a data engineer who owns whether the archive can be read at all. None of them is full-time on it. The phase is short by design and cheap by design.

It is cheap by design because it is the phase where a fatal problem is cheapest to discover. If the archive turns out to be un-trainable, if the scanned logs are too degraded, too inconsistent, or too sparse to support the model the use case needs, the operator wants to learn that before the full team stands up and starts drawing the salary that the later phases carry. The exit gate for assess is therefore a yes-or-no judgement rather than a number: is the archive trainable, and is the problem shaped like something a model can solve. A programme that treats assess as a formality and staffs the pilot before that question is answered has converted the cheapest possible failure into one of the most expensive, because it will discover the un-trainable data with a full team on the clock.

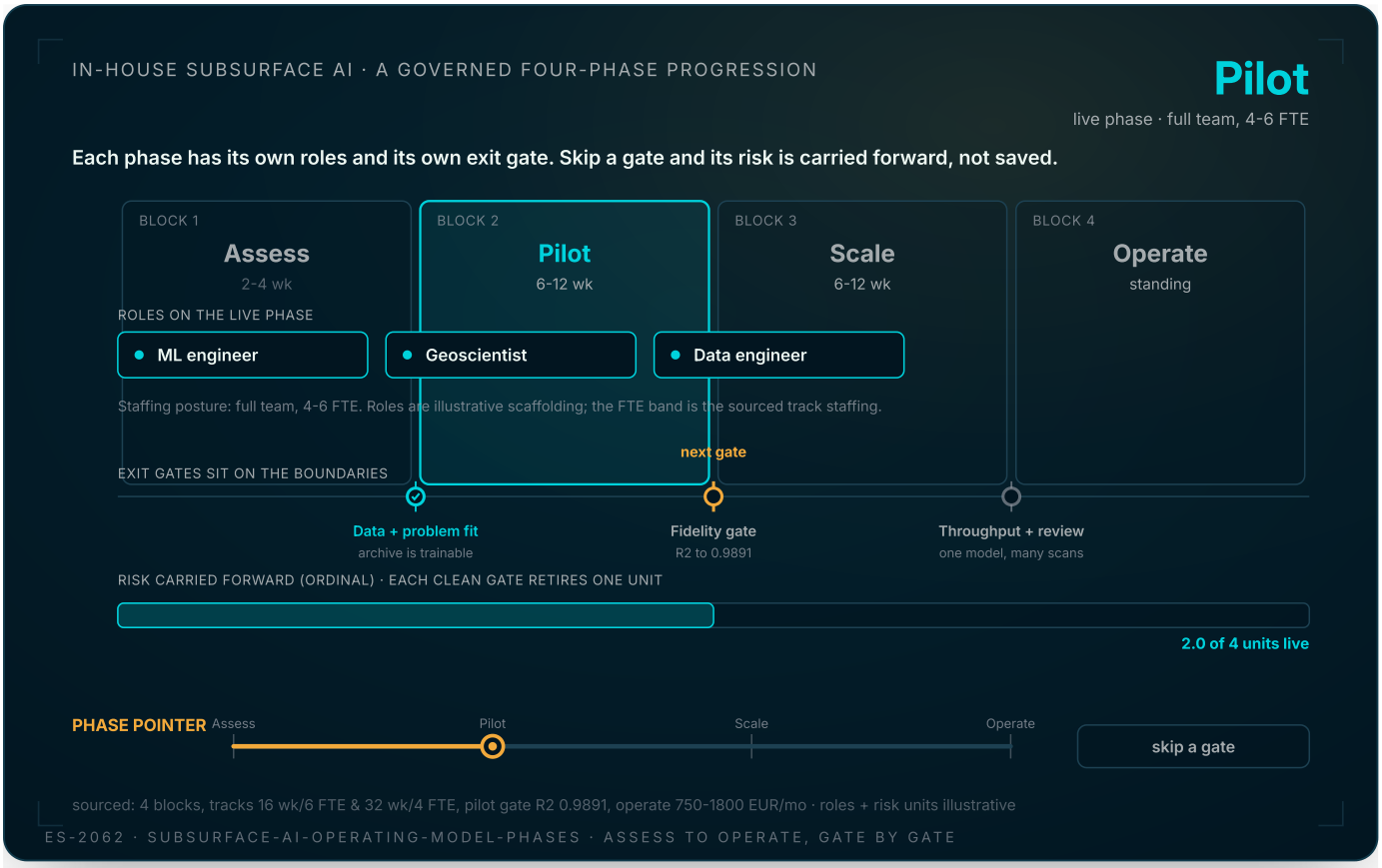
Assess also does the quiet work of sizing. The widest, worst-case member of a raster archive, not the typical one, is what sets the memory budget, the training time, and the throughput target for every phase that follows, and assess is where that worst case is found and measured. Get this wrong and the pilot is scoped against an average that the real data will breach; get it right and every downstream estimate has a defensible anchor. This is unglamorous and it is load-bearing, which is exactly the combination that makes it the phase most often shortchanged.

## Pilot: the first phase that can fail on a number

Pilot is where the full team stands up for the first time, typically four to six staff, and where the programme first spends real money. Its job is to build a working model on a bounded slice of the problem and to prove it against a stated fidelity gate. This is the first phase whose exit gate is a number rather than a judgement, and the discipline that matters most here is that the number is chosen and committed to before the pilot begins, not reverse-engineered from whatever the model happens to achieve.

For the digitisation work the gate is curve-reconstruction fidelity, and our own pilots have peaked at an R-squared of 0.9891 on the reconstructed curve against ground truth. The important thing is not the specific figure but the fact that there is one, and that it is measured on the quantity the operator actually cares about, the reconstructed curve, rather than on a convenient proxy like a segmentation mask overlap that can look excellent while the deliverable curve is wrong. A pilot that reports a strong mask score and no curve score has not passed its gate; it has changed the subject. The exit gate for pilot is a fidelity number on the deliverable metric, and a pilot that cannot state its gate has not piloted, it has demonstrated, which is a different and much weaker thing.

Pilot is also where the operator's own people should be inside the work rather than watching it, because the pilot is the first place the transferable judgement is built. The roster, an ML engineer, a geoscientist who owns ground truth, and a data engineer, is the roster whose skills the operator most needs to internalise, and a pilot run entirely by the vendor with the operator observing produces a passed gate and no transferred capability. The instrument below lays the whole sequence out so the phase-by-phase change in roles, gates, and staffing is legible at once, and so the cost of skipping a gate is visible rather than argued.




*Standing up in-house subsurface AI, read as a governed four-phase progression: assess, pilot, scale, operate, mapped onto the four project blocks the engagement was scoped in. Drag the phase pointer and the picture changes on three registers at once. The roles on the live phase turn over, because a data engineer's assess-phase job is not an MLOps engineer's scale-phase job. The exit gates sit on the boundaries between phases, each reading against a concrete test where one exists: the pilot fidelity gate against peak R-squared 0.9891, the operate gate against the 750 to 1800 EUR per month GPU run-rate. And the risk-carried track underneath falls one unit for every gate passed cleanly. The orange element is the only one that argues: the next exit gate the pointer is working toward. Toggle skip a gate and watch that risk not disappear but carry forward into every later phase, which is the whole case against compressing the sequence to save time. The four project blocks, the track staffing and timeline, the pilot R-squared gate, and the operate run-rate are sourced from the engagement archive; the role names and the ordinal risk units are illustrative scaffolding.*

Read the picture phase by phase and the governed structure is the point. The roster on the live phase turns over as the pointer advances, because the assess-phase data engineer and the scale-phase MLOps engineer are not the same job and should not be the same line in the plan. The exit gates sit on the boundaries between phases, each reading against a concrete test where one exists, the pilot against the fidelity R-squared and the operate phase against the monthly run-rate. And the risk track underneath falls one unit for every gate passed cleanly. Then toggle the skip control and watch what the argument actually is: the risk a skipped gate was meant to retire does not disappear from the track, it is carried forward into every later phase, because the un-trainable archive or the un-piloted fidelity

problem does not become true just because the plan stopped checking for it. Skipping a gate saves the calendar the gate would have taken and forwards its risk downstream, where it is more expensive to fix than it would have been to prevent.

> **THE GATE IS THE PRODUCT, NOT THE OBSTACLE**
>
> The most expensive decision in a build-out is to treat an exit gate as a delay to be managed rather than a risk to be retired. A gate skipped in assess to reach pilot faster does not remove the possibility that the archive is un-trainable; it defers the discovery to a phase with a full team on the clock. A gate skipped in pilot to reach scale faster does not make the fidelity acceptable; it ships an unproven model into an industrialisation phase that will amplify its errors. Govern the gates as the load-bearing part of the plan, because they are.

## Scale: turning a model into a service

Scale is the industrialisation phase. The piloted model exists and clears its fidelity gate on a bounded slice; scale turns it into a service that clears the operator's real volume with reviewable output. The roster turns over most sharply here, toward an MLOps engineer who owns the serving platform, a backend engineer who owns the pipelines around it, and a reviewer who owns the human check on production output. The exit gate for scale is not a new accuracy record. It is throughput and reviewability: can the service process the archive at the rate the operator needs, and is its output in a form a domain expert can check and correct rather than having to trust blind.

The property that makes scale economically sane is that one trained model serves every scan rather than being rebuilt per input. Because a single model, once trained, handles the full range of scan widths and conditions, the marginal cost of digitising one more log at scale is a serving cost and not a training cost, and serving costs collapse as throughput rises. That is what separates a scale phase that is affordable from one that is not, and it is why the scale gate is about throughput rather than about training a bigger or better model. A programme that reaches scale and finds itself retraining per input has not industrialised; it has multiplied the pilot.

## Operate: the phase the programme lives in

Operate is the standing phase, and it is where the programme spends most of its life. The delivery team does not stay stood up here. The right shape is a lean owner who holds the platform plus on-call ML and domain review, and the discipline is drift-watching and run-rate control rather than fresh construction. The exit gate for operate is not an exit at all in the usual sense; it is a standing pair of tests, that the run-rate stays inside budget and that the model does not drift out of fidelity as new data arrives. This is a portfolio to govern rather than a single asset to maintain, because a mature subsurface-AI programme carries more than one model family, the digitiser that reads the rock and, for long-lived projects, models that read very different things such as stakeholder and acceptance dynamics [2], and each has its own drift behaviour to watch.

The cost of operate is dominated by the served model's monthly compute. In our engagements that runs from 750 EUR per month on a high-end GPU tier to 1,800 EUR per month on an advanced one, plus the cost of the lean human review around it. The number to internalise is that this is a modest standing cost against the one-off cost of standing the capability up, and we will return to that comparison because it is the fact that disarms the temptation to shortcut the early phases.

III

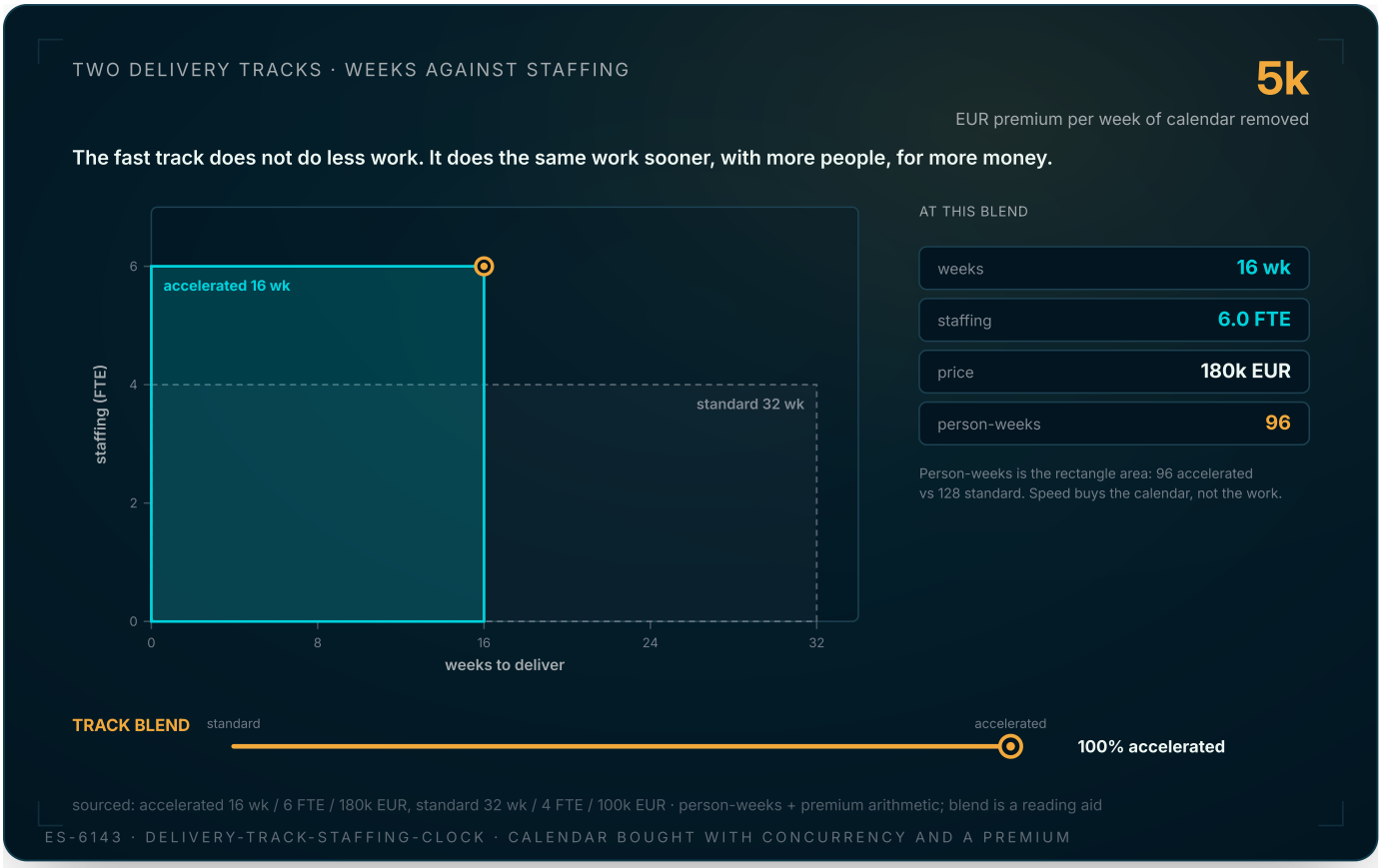
THE TRACKS

# Speed is bought with people and a premium, not with less work

## The two delivery tracks, and what actually separates them

The four phases can be run at two paces, and the engagement was priced at both. The accelerated track compresses the build-out into sixteen weeks with six full-time staff for one hundred eighty thousand EUR. The standard track runs the same build-out over thirty-two weeks with four staff for one hundred thousand EUR. The naive reading of these two numbers is that the accelerated track is a discount for urgency or that it somehow does less. Both readings are wrong, and getting the relationship right matters because the choice between the tracks is one of the first the operator makes and one of the most misunderstood.

What separates the tracks is concurrency and a premium, not scope. The accelerated track does not do less work; it does the same work sooner by putting more people on it at once, and it costs eighty percent more in total, not less. The calendar is bought by adding staff and paying a premium for the compression, and the total labour, measured in person-weeks, is actually higher on the fast track, not lower. That is the counter-intuitive part and it is worth seeing rather than asserting, because it reframes the decision from "cheap-and-slow versus expensive-and-fast" to "the same work, at a premium, for the calendar you need." The instrument below makes the trade visible on the two axes that actually move against each other.




*The two delivery tracks the engagement was priced at, read on the two axes that actually trade against each other: weeks to deliver on the horizontal, staffing in FTE on the vertical. Each track is a rectangle whose area is its person-weeks of effort. The accelerated track is short and tall (16 weeks, 6 FTE); the standard track is long and short (32 weeks, 4 FTE). The point the picture makes is that the accelerated rectangle is not the smaller one, and the accelerated price is the larger one: 180,000 EUR against 100,000 EUR, 80 percent more, for half the calendar. Speed is bought by putting more people on at once and paying a premium, not by doing less work. Drag the track blend and watch the calendar fall while the cost climbs; the orange element is the only argument on the plate, the premium in EUR you pay for each week of calendar the accelerated track removes. All four numbers per track are sourced from the engagement archive; person-weeks and the premium-per-week are arithmetic on those figures, and the blend between the two tracks is a reading aid, not a third priced option.*

Each track is a rectangle whose area is its person-weeks of effort: weeks along the horizontal, staffing up the vertical. The accelerated rectangle is short and tall, the standard one long and short, and the thing to notice is that the accelerated area is not the smaller one. The fast track carries more person-weeks and a higher price, one hundred eighty thousand EUR against one hundred thousand, for half the calendar. Drag the blend and the calendar falls while the cost climbs, and the orange marker reads the only argument on the plate: the premium in EUR you pay for each week of calendar the accelerated track removes. The right way to choose between the tracks is therefore to ask what a week of earlier delivery is worth to the asset, and to pay the premium only when that value clears it.

An operator who chooses the fast track expecting a discount has misread the economics; an operator who chooses it knowing they are buying calendar at a premium has made a defensible call.

> **“The fast track is not a discount for urgency. It is the same work, done sooner, with more people, for more money. Buy it when a week of earlier delivery is worth the premium, and not because it looks cheaper on the calendar.”**
>
> — FROM OUR ENGAGEMENT PLANNING NOTES

## Staffing the phases, not the calendar

The track decision sets the pace, but the staffing that matters is per-phase, not per-programme. A common planning error is to size a single team for the whole build-out and hold it flat across the phases, which over-staffs assess and operate and under-staffs the roster turnover that scale demands. The phase view corrects this. Assess wants a small part-time core. Pilot and scale want the full team, but a different full team, because the roles turn over between them. Operate wants a lean standing crew, not the delivery team retained out of momentum. Planning the staffing against the phases rather than against the calendar is what keeps the accelerated track's six full-time staff from being the wrong six in three of the four phases.

IV

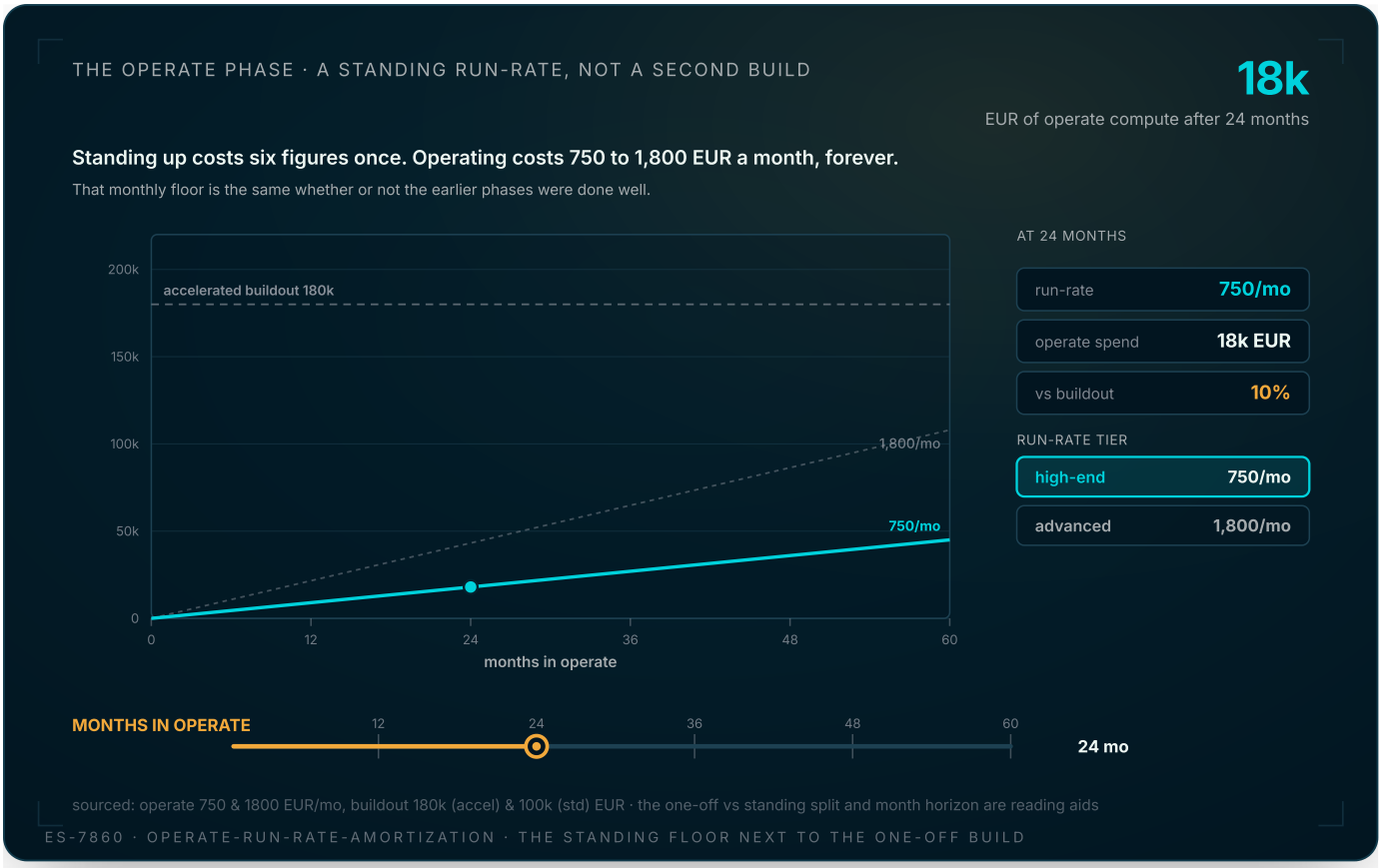
THE FLOOR

# Operate is cheap next to standing up, and shortcuts do not lower it

## The run-rate is a floor, and skipping phases does not move it

The single most expensive misjudgement we see in build-out planning is the decision to shortcut the assess and pilot phases in order to reach the operate phase faster. The reasoning is that the operate phase is where the value is realised, so the sooner the programme gets there the sooner it pays off. The flaw in that reasoning is that the operate phase is not a prize that arrives sooner if you run to it; it is a standing run-rate whose floor is set by the served model's compute and is completely indifferent to how well the earlier phases were run. Reaching it a month early by skipping a gate does not lower the monthly cost by a cent. It only raises the risk carried into the phase, because the model now running in production is one whose fidelity or whose data foundation was never properly tested.

The numbers make the point sharply. Standing the capability up costs six figures once, one hundred thousand EUR on the standard track or one hundred eighty thousand on the accelerated one. Operating it costs 750 to 1,800 EUR per month, and that monthly figure is the same whether the earlier phases were run carefully or rushed. The compute does not know or care how the model was built; it charges the same rent for a well-piloted model and a badly-piloted one. So the entire supposed benefit of shortcutting the early phases, reaching the cheap operate phase sooner, is an illusion: the operate phase was always going to be cheap, the shortcut does not make it cheaper, and the shortcut degrades the asset that runs there. The instrument below prices this honestly.




*The operate phase priced honestly against the phases that precede it. Months in operate run along the horizontal; cumulative EUR of operate compute climbs the vertical. Two run-rate lines rise from the origin: the high-end tier at 750 EUR per month and the advanced tier at 1800 EUR per month, both sourced. A dashed horizontal marks the buildout cost, toggling between the accelerated 180,000 EUR and the standard 100,000 EUR, so the reader can see how long the operate phase runs before its standing cost equals what standing the capability up cost once. The orange element is the only argument: the crossover, the month at which the cumulative run-rate first equals the buildout. At the low tier against the accelerated build that crossover is years out, which is the visual form of the claim that operating is cheap next to standing up, and that the monthly floor is set by the served model's compute regardless of whether the earlier phases were done properly, so skipping them saves nothing at this stage. The two run-rates and the two buildout costs are sourced from the engagement archive; the one-off-versus-standing split and the sixty-month horizon are reading aids on those figures.*

Months in operate run along the horizontal and cumulative operate compute climbs the vertical, with the two run-rate tiers rising from the origin and a dashed line marking the one-off buildout cost. The orange crossover is the argument: the month at which the cumulative operate run-rate first equals what standing the capability up cost once. On the low tier against the accelerated build that crossover is years out, which is the visual form of the claim that operating is cheap next to standing up. The floor is set by the compute, not by the quality of the build, so the shortcut that reaches this floor sooner saves nothing here and carries a degraded model into the phase. The honest planning posture is the opposite

of the shortcut: spend the assess and pilot phases properly, because they are where the risk is retired, and treat the low operate run-rate as the reward for a build done right rather than a destination to sprint toward.

## What good governance of this looks like

A build-out governed well has a small number of visible properties. The four phases are named in the plan and funded one at a time, each against the gate the previous phase passed. The exit gates are written down before their phase begins, and the ones that read against a number, the pilot fidelity gate and the operate run-rate, name the number in advance rather than discovering it after. The staffing is planned per phase and turns over deliberately between pilot and scale rather than being held flat. The track decision is made with the premium-per-week-of-calendar in view, not on the mistaken belief that the fast track is cheaper. And the operate run-rate is understood as a floor set by the compute, so no one proposes shortcutting the early phases on the theory that it lowers the cost of running the result. None of these is exotic. Together they are the difference between a programme that stands up a capability and one that ships a model with a support contract stapled to it.

### Assess before you build

- A small core, part-time: a subsurface lead, an ML architect, a data engineer
- The exit gate is a yes-or-no on whether the archive is trainable at all
- This is the cheapest phase to fail in, and the most expensive to skip
- It sizes every phase that follows against the widest, worst-case data

### Pilot against a real gate

- The full team stands up, four to six staff, for the first time

- The exit gate is a fidelity number, curve reconstruction to peak R-squared 0.9891
- A pilot that cannot state its gate has not piloted, it has demonstrated
- Passing here is what earns the right to spend on scale

## Scale is an industrialisation phase

- The roster turns over toward MLOps, backend, and review roles
- The exit gate is throughput and reviewable output, not a new accuracy record
- One trained model serving every scan is the property that makes scale cheap
- This is where a demo becomes a service the operator can run

## Operate is a run-rate, not a project

- A lean owner plus on-call, not the delivery team held indefinitely
- The standing cost is monthly compute, 750 to 1800 EUR, plus review
- The gate is drift and run-rate discipline, watched, not built once
- This floor is the same however well the earlier phases were run

### WHAT TO CARRY OUT OF THIS

1. Standing up in-house subsurface AI is a **four-phase** progression, assess then pilot then scale then operate, mapped onto the **4** project blocks the engagement is scoped in. Each phase has its own roster, its own exit gate, and its own staffing shape, and treating the build-out as one undifferentiated push toward a demo is the failure mode to avoid.
2. The exit gates are the product, not the obstacle. Some read against a number, the pilot against curve fidelity to peak R-squared **0.9891** and the operate phase against a **750** to **1,800** EUR monthly run-rate; others are yes-or-no, such as whether the archive is trainable at all. A gate skipped to save calendar forwards its risk downstream, where it costs more to fix.
3. The accelerated **16**-week, **6**-staff, **180,000** EUR track and the standard **32**-week, **4**-staff, **100,000** EUR track differ by concurrency and a premium, not by scope. The fast track does the same work sooner with more people for more money, so buy it only when a week of earlier delivery is worth the premium.
4. The operate phase is a low standing run-rate whose floor is the served model's compute, unmoved by how well the earlier phases were run. Reaching it early by skipping a gate does not lower the monthly cost; it only carries a degraded model into the phase the programme lives in.
5. Govern the build-out as a capability programme: fund phases one at a time against the previous gate, write the gates down before their phase begins, staff per phase with deliberate roster turnover, and read the run-rate as the reward for a build done right rather than a destination to sprint toward.

## Limitations

The four-phase structure and the roles named within each phase are the operating model we run in practice, but the role rosters and the ordinal risk track in the phases instrument are illustrative scaffolding rather than sourced quantities: the number of project blocks, the delivery-track timelines and staffing, the pilot fidelity gate, and the operate run-rate are the sourced figures, and the phase-by-phase split of weeks and the risk-unit accounting are presentational aids built to argue the sequence rather than to predict a specific plan. The two delivery-track figures, sixteen weeks at six staff for one hundred eighty thousand EUR

and thirty-two weeks at four staff for one hundred thousand EUR, are the tracks the engagement was priced at; the person-weeks and the premium-per-week-of-calendar in the staffing instrument are arithmetic on those figures, and the blend between the two tracks is a reading aid, not a third priced option, so an operator should treat intermediate blends as interpolation rather than as a quotable price. The operate run-rate of 750 to 1,800 EUR per month is the served-model compute cost from our engagements and does not include the lean human review around it or any storage, networking, or licensing the operator's own environment adds, so it is a floor on the running cost rather than a full total, and the one-off-versus-standing split and the sixty-month horizon in the amortization instrument are reading aids on the sourced figures. The fidelity gate of peak R-squared 0.9891 is a curve-reconstruction result specific to the digitisation work and to the deliverable metric it is graded on; a different task, a different metric, or a different archive will set its own gate, and the value here should be read as evidence that a pilot can and should commit to a stated fidelity number rather than as a target any programme will hit. Finally, the case for building in-house at all, and for treating the operate phase as a portfolio rather than a single asset, rests on the cited upstream and long-term-project literature and on our own engagement experience; it is a strategic argument about recurring, multi-family capability, and an operator whose need is a single bounded task may rationally rent rather than build.

## References


1. Koroteev, D., Tekic, Z. (2021). *Artificial intelligence in oil and gas upstream: Trends, challenges, and scenarios for the future.* Energy and AI. The upstream survey that frames why an operator would carry AI capability in-house rather than rent it per project, and the order-of-magnitude gains that justify the build-out. https://www.sciencedirect.com/science/article/pii/S2666546820300033

2. Buah, E., Linnanen, L., Wu, H., Kesse, M. A. (2020). *Can Artificial Intelligence Assist Project Developers in Long-Term Management of Energy Projects? The Case of CO2 Capture and Storage.* Energies, 13(23), 6259. The evidence that a subsurface-AI programme carries more than one model family, which is why the operate phase is a portfolio to govern and not a single asset to maintain. https://www.mdpi.com/1996-1073/13/23/6259

## Get the full whitepaper

This page is the long-form summary. The complete whitepaper adds the per-phase RACI for the four blocks, the exit-gate checklists we run at each boundary, the accelerated-versus-standard track worksheet with the premium-per-week arithmetic, and the operate-phase run-rate model with the review and environment costs the compute floor does not include.

EARTHSCAN

## A Phased Operating Model for Standing Up Subsurface AI Capability

Published November 2024. Generated 2026-07-22.
https://www.earthscan.io/whitepapers/a-phased-operating-model-for-standing-up-subsurface-ai-capability