EARTHSCAN

WHITEPAPER 17 PAGES / FEBRUARY 2025

MLOps for Subsurface AI: Reproducibility, Versioning and Handover

Most subsurface AI programmes are run as research and die as research. They produce a notebook that beats the baseline, a deck that impresses a steering committee, and a model nobody can rebuild a year later. This whitepaper is a field guide to the alternative: running a multi-year subsurface ML programme as engineering. Drawn from a three-phase, roughly twenty-month formation-evaluation engagement with a mid-sized Middle East carbonate operator, it covers the four disciplines that decide whether a trained system survives — data versioning with DVC, experiment tracking with Weights & Biases, CI/CD and serving, and bit-level reproducibility through seeding and frozen checkpoints — and then the discipline almost every programme skips: the handover that puts the model, the pipeline, the runbooks and the know-how inside the operator's in-house ICT team. We argue that the handover is not a closing courtesy. It is the acceptance test for everything built before it, and a programme that cannot be handed over was never engineered in the first place.

Tannistha Maiti
Senior AI Researcher

Tarry Singh
Founder & CEO

MLOPS / SUBSURFACE-AI / REPRODUCIBILITY / DATA-VERSIONING / CI-CD / MODEL-HANDOVER

CONTENTS

A subsurface model is not the file you train. It is the entire apparatus required to retrain it on demand, explain why it predicted what it did, and run it next year after the geoscientist who built it has moved on. Most operators acquire the first thing and assume they have bought the second. They have not. The gap between a model that works in a notebook and a model an in-house team can own is an engineering gap, and it is the gap where multi-year subsurface AI programmes quietly fail.

Why this is an engineering problem, not a research one

Over roughly twenty months, working with a mid-sized Middle East carbonate operator across three phases, we built a formation-evaluation system on high-resolution borehole image logs: bedding-plane and fracture detection, dip and azimuth regression, vug analysis, and well-to-well correlation. The modelling was hard. The modelling was also not the part that determined whether the operator would still have a working system at the end.

The research literature measures success in a metric on a held-out set. An operational programme is measured differently. Can you reproduce last quarter's result to the decimal? Can you tell which of forty checkpoints is the one running on the asset? When a new batch of wells arrives, can the operator retrain without you in the room? These are not modelling questions. They are versioning, provenance, and handover questions — and they are answered by infrastructure, not by a better architecture.

> **THE 15 PERCENT RULE**
>
> A trained model is roughly fifteen percent of a production subsurface AI system. The other eighty-five percent — ingestion, versioning, experiment tracking, CI/CD, reproducibility scaffolding, serving, and the handover runbooks — is the part that decides whether the fifteen percent survives contact with a second year and a second team.

This whitepaper walks the eighty-five percent. It is written for the IT lead, the CTO, and the MLOps engineer who will inherit a subsurface model and have to keep it alive — and it is organised around the four disciplines that decide whether they can.

Discipline one: version the data, not just the code

Subsurface teams instinctively version code. Almost none version data, and in subsurface ML the data is where the entropy lives. The raw input — apparent dip and azimuth picks, well radius, the binary wireline log file, the interpreter's PDF — arrives well by well, gets normalised, gets cut into overlapping patches, gets augmented, gets split into train, validation, and test. By the time a model trains, the bytes it sees are four or five transformations removed from anything an engineer can point at. If you cannot name the exact dataset a run consumed, you cannot reproduce the run, and you cannot defend the prediction.

We solved this the way it has to be solved: every dataset was a first-class, immutable, content-addressed artefact. We used **DVC** for data and model versioning sitting alongside **Git** for source, so that a commit hash pinned not just the code but the exact processed dataset and the model weights it produced. Each generated dataset was named with a hexadecimal UUID rather than a human label like `final_v2`, because human labels lie and hashes do not. A run referenced its dataset by that UUID; the UUID referenced a frozen set of patches; the patches traced back through the augmentation recipe to the specific wells and depth intervals that produced them.

This mattered because the dataset itself moved constantly. The core sinusoid dataset grew from roughly 900 image-and-ground-truth pairs to over 55,000 — a 65-fold expansion driven by overlapping-patch generation and augmentation. One reservoir interval alone went from 236 raw patches to 4,212 after augmentation, lifting sinusoid-bearing patches from 19 to 2,046. Without versioning, "the 55,000-pair dataset" is a meaningless phrase, because there were dozens of them. With versioning, every model was permanently bound to the one it actually learned from.

> **DATA QC IS A VERSIONED GATE, NOT A VIBE**
>
> Of ten wells received in one intake, two were excluded before training — both carried abnormal static binary-log value ranges that fell outside the normal 0–255 band and defeated normalisation, and one of the two had additionally been acquired with a different image-logging tool whose response was not directly comparable. That exclusion is a data decision with model consequences, so it lives in the dataset's

> provenance record. A future engineer reading the lineage sees eight wells and the recorded reason for the other two. Silent exclusion is how irreproducibility enters a programme.

The operational layer on top of this was a data-management dashboard that itself went through twelve numbered versions over the engagement. Versioning is not a one-time setup. It is a habit that has to hold for the life of the programme, and the only way it holds is if the tooling makes the versioned path the easy path.

Discipline two: track every experiment, including the failures

A subsurface model is the survivor of a search, not the product of a recipe. The search space here was large by intent. Backbones spanned ResNet-10 through ResNet-34; optimisers covered SGD, Adam, and AdamW; learning rates were swept from 0.001 down to 0.0005; encoder and decoder depth, feed-forward dimension, loss composition, and class weighting were all in play. Across this space we ran hyperparameter sweeps with **Weights & Biases** as the experiment tracker, logging every run's configuration, metrics, and artefacts to a single shared project so that no result existed only in someone's terminal scrollback.

The discipline that separates an engineering programme from a research one is logging the failures with the same rigour as the wins. The first supervised runs overfit — constant prediction on the validation set despite a matched class distribution across splits. That is a result. It drove a specific, recorded set of changes: single-channel input, a smaller backbone than ResNet-18, a different sinusoid loss, augmenting only the sinusoid-bearing patches rather than the whole set, and incorporating well angle. None of those decisions would be defensible a year later if the failing runs that motivated them had been deleted. An experiment tracker is a lab notebook that cannot be edited after the fact, and that immutability is the point.

The search converged on a configuration the operator's team could read off a single record: a ResNet-10 backbone trained from scratch with no pretrained weights, four encoder and four decoder layers, feed-forward dimension 1024, AdamW at an optimal learning rate of 0.0004, dropout 0.2, a combined focal-and-L1 loss with class-loss weight 5 against parameter-loss weight 1, an inference probability threshold of 0.5, and early

stopping after 40 epochs without improvement. That is not a paragraph of folklore. It is a tracked, reproducible artefact — and when the operator's engineers needed to retrain on new wells, it was the starting point they inherited, not a starting point they had to rediscover.

The case study's real argument isn't faster retrains — it's that the gap between 'model is stale' and 'model is fresh again' is where decisions ran on quietly degrading predictions. Drag 'today' across a year of operating life: under the manual queue, drift hid for months and a retrain took six-plus weeks, so the model's staleness sawtooth grows long teeth; under the agentic loop, drift surfaces in days and a retrain runs overnight, so the teeth collapse. The orange band is the silent-drift exposure the loop removes — the window in which five of 18 requeued models had drifted into a combined $4.2M of misallocated infill capital. The retrain cycle times (6 weeks → overnight, ~40×), the drift-detection cut (months → days), +22% accuracy, the $4.2M figure and ~40 assets are the case study's own; the week-by-week staleness curve shape, the year-long retrain cadence and the vertical weeks-stale scale are schematic, drawn to argue the gap rather than chart a measured series.

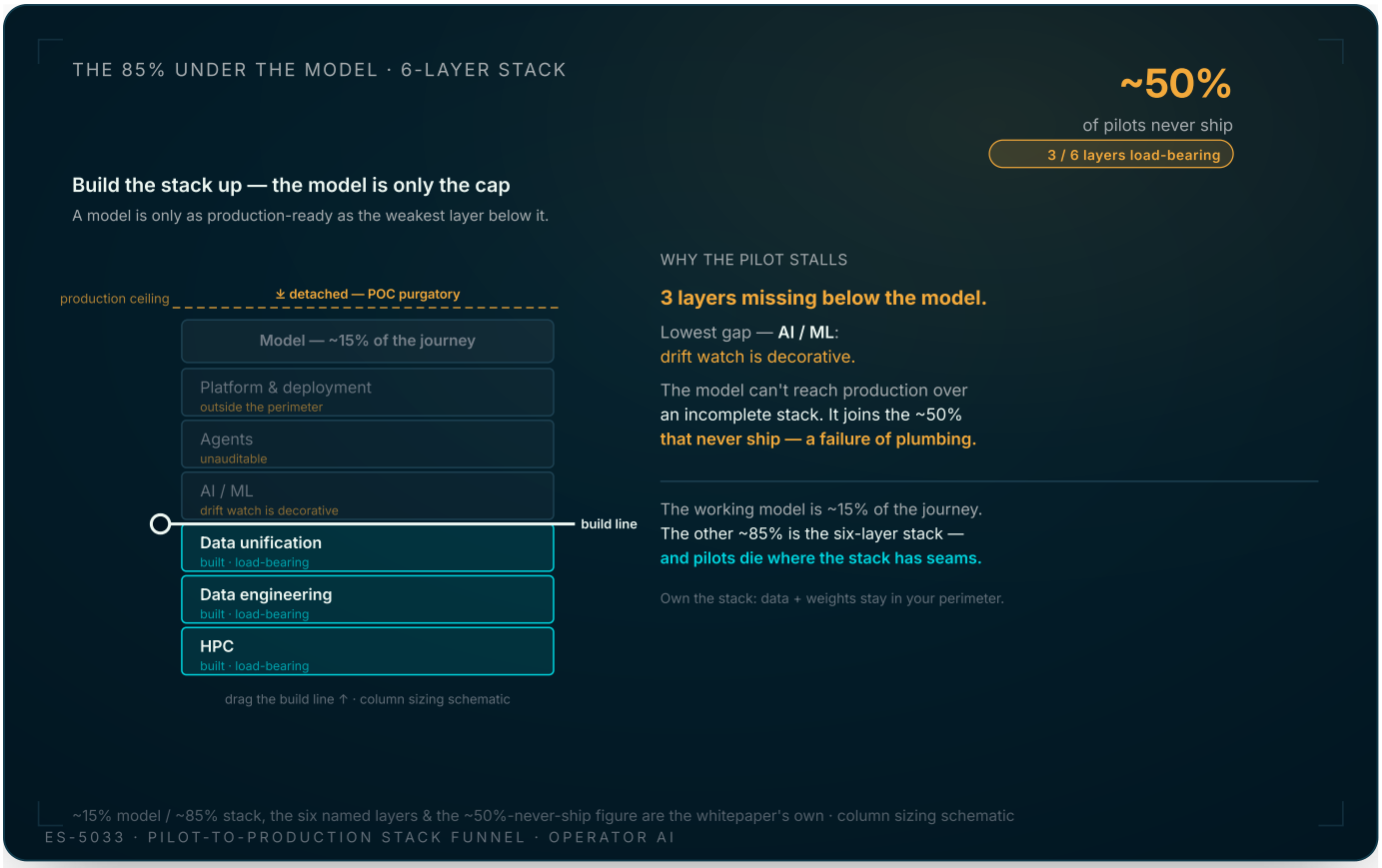
The drift-and-staleness picture above is the reason experiment tracking has to outlive the build phase. A model is not finished when it ships; it begins decaying the moment the formation it sees in production diverges from the formation it trained on. The team that owns the model needs the same tracking discipline the build team used, because the

retrain that refreshes a stale model is itself an experiment that has to be logged, compared, and version-pinned. Hand over the model without handing over the tracking habit, and the operator inherits a system they can run exactly once.

Discipline three: CI/CD and the path to serving

Most subsurface ML stops at a checkpoint and a Streamlit demo. Production needs the rung above that: continuous integration that catches a broken pipeline before it reaches the asset, and a serving path that turns a frozen checkpoint into something a geoscientist can run without a data scientist standing behind them. We treated this as an explicit phase rather than an afterthought — the programme's own phase ladder named "model rollback and serving" and "production CI/CD" as Phase 3 deliverables, distinct from the Phase 1 and Phase 2 modelling work.

The lifecycle we built and handed over was the full loop, not a fork of it: data exploration, feature engineering, training and experimentation, evaluation against geoscientist-validated ground truth, a production-ready frozen model, CI/CD, drift and staleness monitoring, and a feedback loop back to data. A model registry sat at the centre, so that "the model running on the asset" was an addressable, promotable object rather than a file path someone remembered. The serving surface was a set of applications running inside the operator's perimeter on a custom MLOps control plane, with the underlying compute provisioned on-premise rather than rented from a cloud that would also be ingesting the operator's subsurface data.



Pilots don't stall because the model is weak. The working model is only ~15% of the journey; the other ~85% is a six-layer engineering stack (HPC → Data engineering → Data unification → AI/ML → Agents → Platform/deployment), and a project ships only when every layer below the model is built to production grade. Drag the build line up the load-bearing column: with all six built the model reaches the production ceiling; with any gap below it the model detaches into POC purgatory — the ~50% that never ship. The ~15%/~85% split, the six layers and the ~50% figure are the whitepaper's own; the equal-sixths column sizing is schematic.

The funnel above is the attrition every operator should expect and most do not plan for. A pilot that produces a validated metric has cleared one gate of several. Ingestion with provenance, a versioned model registry, a CI pipeline, a serving layer inside the security perimeter, drift monitoring, and a feedback loop are all gates between a working notebook and a workflow on the asset — and a programme that skips any of them ships a model that cannot be operated. The discipline here is to scope CI/CD and serving in from the first phase as line items with acceptance criteria, not to bolt them on once the modelling "is done." Modelling is never done; serving is what makes that survivable.

Discipline four: reproducibility you can defend in a review

Reproducibility in subsurface AI is not an academic nicety. A dip and azimuth prediction feeds a structural model that feeds a drilling decision worth millions. When a reviewer asks why the model placed a fracture at a given depth, "the network said so" is not an answer. The answer has to be reconstructible: this dataset UUID, this frozen checkpoint, this seed, this threshold, this depth-conversion arithmetic.

Several things make subsurface reproducibility harder than it looks, and each has to be engineered around explicitly:

- **Irreducible measurement error has to be separated from model error.** At the image resolution in play, a single image-log pixel corresponds to about 3 centimetres of depth, so a ±3 cm uncertainty is baked into the input before the model does anything. When a prediction sits 3 cm off a ground-truth pick, that is the instrument, not the network — and the localisation tolerance that defines a true versus false positive has to be chosen and recorded against that physical floor. We swept tolerances at 2, 4 and 6 cm precisely so the metric could be read against the instrument, not in spite of it; report a number without naming the tolerance and the metric is meaningless.
- **Frozen checkpoints have to be self-describing.** A checkpoint whose filename encodes its own provenance — learning rate, epoch count, backbone, loss, and a timestamp — tells an engineer what it is without a lookup. "The 0.0004 / ResNet-10 / focal-plus-L1 checkpoint trained at this timestamp" is unambiguous; `best_model_final.pt` is a future incident.
- **Thresholds are part of the model.** A focal-loss configuration tuned for recall ran at a 0.5 confidence threshold; a cross-entropy configuration tuned for precision-and-recall ran at 0.9. A checkpoint shipped without its threshold is a checkpoint that will be run wrong.
- **Determinism has to survive the data growth.** Because the dataset itself was non-stationary — growing from 8 to 11 to 14 to 16 wells over the phases, where moving from 8 to 11 wells improved depth, dip, and azimuth error by roughly 0.007 MAE — every reported number is meaningless unless it is pinned to the well-count and dataset version that produced it. Reproducibility and versioning are the same discipline viewed from two angles.

> **THE REPRODUCIBILITY TEST**
>
> The honest test of a subsurface ML programme is this: hand a new engineer the repository, the dataset UUID, and the checkpoint name, and ask them to reproduce last quarter's headline number to the decimal, without talking to the person who

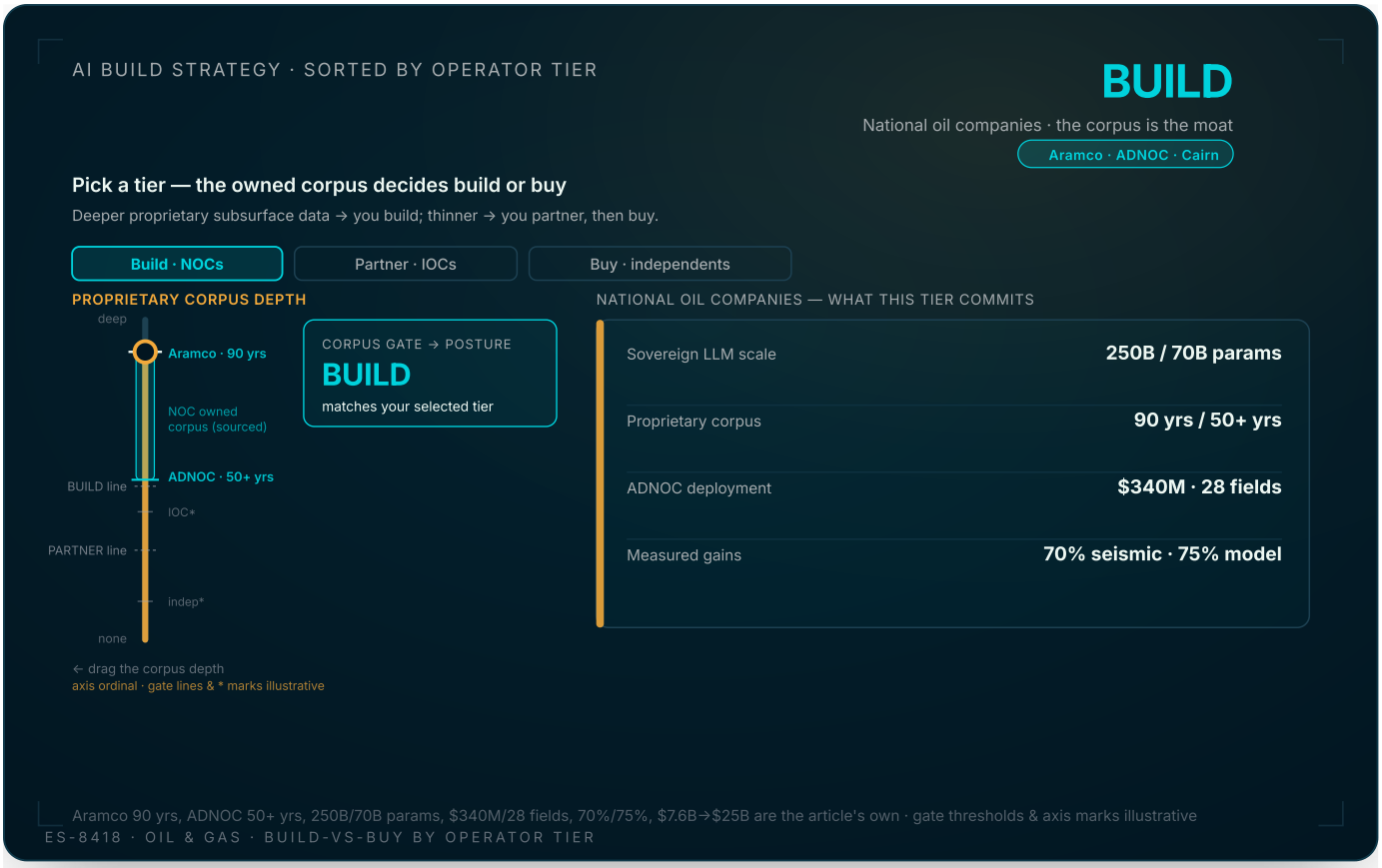
> produced it. If they can, the programme is engineered. If they cannot, it is research wearing an operations costume — and it will not survive the handover.

The final mile: handover to the operator's ICT team

Everything above exists to make one thing possible: handing the system to the operator's in-house ICT team and walking away without the system dying. This is the discipline most programmes skip, and skipping it is why so many "successful" pilots are quietly dead within a year of the consultants leaving.

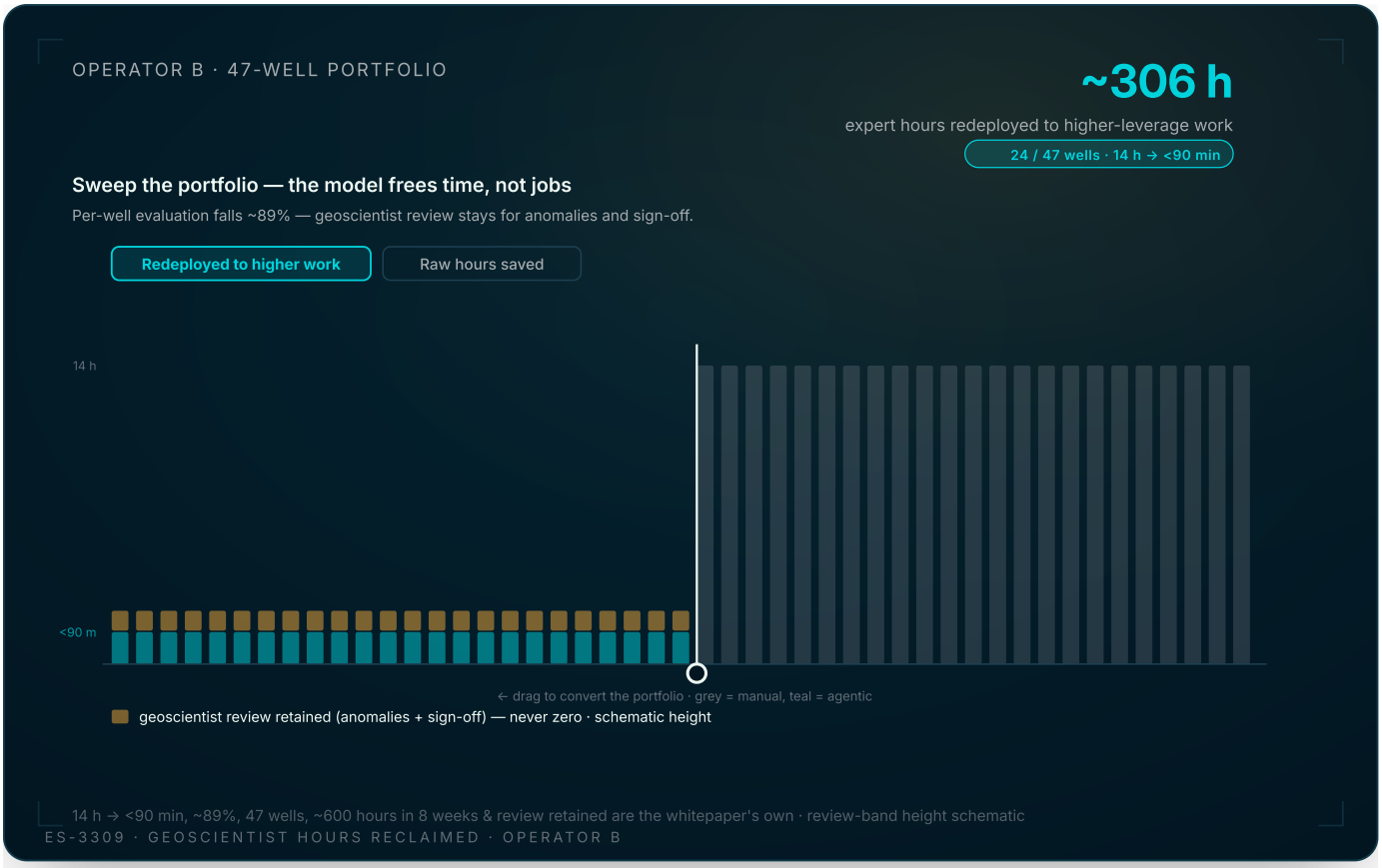
Handover is not a documentation drop. We treated it as the acceptance test for the whole programme, and structured it three ways. First, **deliverable packaging**: each of the three production capabilities — vug detection, bedding-and-fracture detection, well-to-well correlation — was handed over as a complete unit comprising the versioned dataset, the frozen model, the architecture, the output format, and the runbooks. A model without its dataset and its operating documentation is not a deliverable; it is a liability with good metrics.

Second, **a real choice about operating posture.** We did not assume the operator wanted to run everything themselves on day one. We costed three honest scenarios with their genuine trade-offs: a fully self-operated path, where the team retrains independently on their own on-premise GPU hardware on a cadence measured in days; the current managed path, where retraining inside the existing arrangement runs in minutes to hours; and an off-premise managed path, where retraining is handled externally on a two-to-three-week cadence. The point of laying out all three is that capability transfer is a dial, not a switch — and the operator, not the vendor, sets it.



In 2026 the AI build-vs-buy split in oil & gas is sorted by operator tier, and the deciding variable is the depth of the proprietary subsurface corpus an operator owns. Pick a tier — NOCs (Build), Western IOCs (Partner), mid-tier independents (Buy) — and the panel reconfigures to that tier's posture, named operators and the article's own commitments. The orange ladder is the single argument: the deeper the owned corpus (the sourced NOC band runs from ADNOC's 50+ years to Aramco's 90 years), the further toward BUILD a tier sits. Drag the corpus-depth marker — or step tiers with the chips / arrow keys — and the recommended posture snaps to the band the depth lands in. Named operators, the NOC corpus depths, model sizes ($340M / 28 fields, 250B / 70B params), the 70% / 75% gains and the $7.6B→$25B market are the article's own; the corpus-depth axis, the gate thresholds and the IOC / independent marker positions are illustrative.

Third, and most important, **people, not just artefacts.** Software hands over cleanly; the judgement to operate it does not. The programme deliberately built local capability alongside the system, training a cohort of 55 young professionals — 15 of them local nationals, drawn from regional universities as in-country capability building — so that the know-how to run, debug, and extend the platform lived inside the region rather than departing with the delivery team. A handover that transfers files but not judgement produces shelfware with a maintenance contract. The infrastructure disciplines in this whitepaper are what make the judgement transferable: a versioned dataset, a tracked experiment history, a CI/CD pipeline, and a reproducible checkpoint are teachable. A model that only one person understands is not.



The engineering layer didn't replace the geoscientist — it moved expert time off rote evaluation onto judgment. On Operator B's 47-well portfolio, per-well Well-Log-Detection evaluation fell from 14 hours to under 90 minutes (~89%), reclaiming ~600 expert hours in 8 weeks — while geoscientist review was explicitly retained for anomaly cases and final sign-off. Sweep across the portfolio: each 14-hour bar collapses to a <90-minute sliver, the reclaimed-hours total fills toward ~600, and an orange 'review retained' band rides every converted well and never shrinks to zero. All headline numbers are the whitepaper's own; the review-band height is schematic (review is retained but its hours aren't quantified).

The return on all of this is not abstract. The same engineering that makes a system reproducible and handoverable is what frees expert time at scale — interpretation work that took a geoscientist hours per well collapses to a reviewed, agentic workflow, with the expert retained for anomalies and sign-off rather than for mechanical picking. But that dividend only persists if the operator can keep the system running. A productivity gain that evaporates when the build team leaves was never a gain. It was a loan.

What good looks like

For an IT lead or CTO evaluating a subsurface AI programme — one you are buying, one you are running, or one you are trying to rescue — the questions that matter are not about model architecture. They are about whether the programme was engineered to be owned:

- Is every dataset content-addressed and immutable, with data-QC exclusions recorded as provenance rather than applied silently?
- Is every experiment, including the failures, tracked in a shared, immutable record that the operator's team can read and extend?
- Are CI/CD, a model registry, and a serving path inside the security perimeter scoped as first-phase deliverables with acceptance criteria — not bolted on at the end?
- Can a new engineer reproduce a headline number to the decimal from the repository, a dataset UUID, and a checkpoint name alone?
- Does the handover transfer datasets, frozen checkpoints, runbooks, and judgement — with the operating posture chosen by the operator, not assumed by the vendor?

If the answer to all five is yes, the operator owns a system. If the answer to any is no, they own a checkpoint and a countdown.

WHAT THIS WHITEPAPER ARGUES

1. A trained subsurface model is ~15% of a production system; the other 85% is versioning, tracking, CI/CD, reproducibility, and handover.
2. Version data, not just code — every dataset content-addressed and immutable (DVC + Git, hex-UUID dataset names), with QC exclusions recorded as provenance.
3. Track every experiment including failures in an immutable shared record (W&B); the retrain that refreshes a stale model is itself a tracked experiment.
4. Reproducibility means separating instrument error from model error, self-describing frozen checkpoints, shipped thresholds, and numbers pinned to dataset version.
5. The handover is the acceptance test for everything before it — transfer datasets, checkpoints, runbooks AND judgement, with operating posture chosen by the operator.

References

International Energy Agency, 2025 International Energy Agency. Energy and AI Special Report (2025). Missing internal expertise identified as the dominant adoption barrier across the energy sector. https://www.iea.org/reports/energy-and-ai

McKinsey & Company, 2025 McKinsey & Company. The State of AI (2025). Workflow redesign identified as the single strongest EBIT correlate of AI value capture. https://www.mckinsey.com/

Carion et al., 2020 N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko. End-to-End Object Detection with Transformers (DETR). ECCV 2020. Architectural basis for the set-prediction detection approach. https://arxiv.org/abs/2005.12872

Sculley et al., 2015 D. Sculley et al. Hidden Technical Debt in Machine Learning Systems. NeurIPS 2015. The canonical argument that ML code is a small fraction of a production ML system. https://papers.nips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

EARTHSCAN

MLOps for Subsurface AI: Reproducibility, Versioning and Handover

Published February 2025. Generated 2026-07-22.
https://www.earthscan.io/whitepapers/mlops-for-subsurface-ai