AI in Mining

A 5 Day Hands-On Workshop

Expression of Interest

For Mine Planning & Technical Services Teams

Prepared by IMC Mining

March 2026

1. Executive Summary

Mining companies make billion-dollar decisions in spreadsheets. Block models with millions of records are summarised in pivot tables. Cost models with hundreds of interlinked formulas live in workbooks maintained by a single person. Schedule optimisation results are copy-pasted between Excel files that nobody fully understands. Everyone knows this is fragile, but nobody knows what the alternative looks like in practice.

This workshop shows your team the alternative — and they build it themselves.

IMC Mining is offering a 5 day intensive, hands-on workshop that teaches mine planning engineers, technical services professionals, and geoscientists how to move heavy computation out of spreadsheets and into robust, auditable, database-backed systems — while keeping Excel as the familiar interface for summaries, reports, and team consumption. Participants leave with working software, not slide decks.

What makes this workshop unique is that it is delivered by practitioners who have already solved this problem at scale. IMC Mining's own platforms — MiningIQ and MineCost — process multi-million block models, run first-principles cost engines, and then generate the Excel outputs that analysts and managers expect: formatted workbooks with VBA macros, embedded charts, NPV summaries, and equipment fleet reports. The workshop teaches your team the same patterns.

2. The Real Problem: Spreadsheets at Scale

Spreadsheets are brilliant tools for thinking, exploring, and presenting. They are terrible tools for managing large-scale mining data, enforcing calculation integrity across teams, and auditing how a number was derived six months ago. The mining industry has outgrown them — but hasn't yet replaced them.

What Goes Wrong

- **Hidden complexity:** A typical mine cost model has hundreds of interlinked formulas across dozens of sheets. AI tools like Cowork and Windsurf can now ingest a directory of spreadsheets, deconstruct every formula and cross-sheet reference, and map out what the workbook is actually doing. When we run this analysis, we routinely find formulas that reference the wrong cell after a row insertion three years ago.
- **Single-person dependency:** If the person who built the master workbook leaves, the model becomes a black box. There is no version history, no test suite, and no way to verify that a change didn't break something downstream.
- **Scale limits:** Block models with 2+ million records, 30 years of drilling data, and daily dispatch logs cannot be meaningfully processed in Excel. Teams resort to sampling, aggregating, or simply ignoring data that doesn't fit.
- **No audit trail:** When a board paper says the NPV is $1.2B, can you trace that number back through every assumption, every discount rate, every cost escalation? In a spreadsheet, the answer is almost always no.

What Should Replace Them (and What Shouldn't)

The Key Insight

The answer is not "stop using spreadsheets." The answer is "stop computing in spreadsheets." Move the heavy lifting — the cost engine, the risk model, the schedule optimisation — into proper code with version control, unit tests, and database storage. Then generate the Excel files your teams already know how to read. Spreadsheets become the presentation layer, not the calculation layer.

This is exactly how IMC Mining's production systems work. MineCost runs a first-principles cost engine in Python/Django, then exports 20+ sheet .xlsm workbooks with VBA macros for validation, dropdown menus for data entry, and formatted summaries for management. MiningIQ processes pit optimisation results through its FlowOpt engine, then generates analyst-ready spreadsheets with equipment fleets, price decks, and NPV calculations that plug directly into the cost model. The database is the source of truth; the spreadsheet is the delivery format.

3. What the Workshop Teaches

Every lab exercise follows the same pattern that IMC uses in production: compute in Python, store in a database, deliver through a familiar interface (Excel, dashboard, or both). By the end, your team will be able to:

- **Audit existing spreadsheets with AI:** Use tools like Cowork and Windsurf to ingest a directory of workbooks, map every formula and cross-sheet reference, identify disconnects, circular references, and potential errors — then produce a plain-English report of what the spreadsheet is actually doing.
- **Build database-backed calculation engines:** Move cost models, risk models, and schedule summaries out of Excel and into Python with Pandas, backed by proper data storage. Every calculation is versioned, testable, and auditable.
- **Generate Excel outputs programmatically:** Use openpyxl to create formatted workbooks — with VBA macros, dropdown validation, conditional formatting, and embedded charts — that look exactly like the spreadsheets your teams are used to. The difference: the numbers come from a single source of truth, not a chain of VLOOKUP formulas.
- filters, and real-time recalculation — the modern alternative to spreadsheet what-if analysis.
- **Use AI tools for document analysis and team collaboration:** Leverage tools like Cowork and Windsurf to interrogate company documents, SOPs, and geotech reports — no custom development required.
- **Deploy to the cloud:** Put your dashboards and tools on AWS so your whole team can use them, with proper security and cost controls.

4. How IMC Already Does This in Production

This workshop is not theoretical. Every pattern taught is drawn from IMC Mining's production platforms. Here is what those platforms actually do:

MineCost — The Spreadsheet Replacement Pattern

MineCost is a Django web application that replaces the traditional Excel cost model with a database-backed engine. But critically, it still speaks Excel:

- **Scenario Export (.xlsm):** The ScenarioExporter class generates complete macro-enabled workbooks with 20+ sheets: Project_Data, Schedule, Cost_Tree, Equipment, Rosters, Fleet data, Price Deck, Unit Costs, Capital, and Financial inputs. The workbook includes VBA modules for dropdown validation, data checking (errors flagged in red, warnings in orange), and roster preview calculations.
- **Financial Summary Generation:** Reads scenario results from the database, calculates yearly aggregations with proper weighted averaging for grades and recovery rates, and writes formatted summary workbooks with embedded PNG charts (waterfall diagrams, schedule timelines, financial profiles).
- **Output Data Export:** Generates structured output workbooks with Financial Summary, Cost Tree, Schedule Summary, Fleet Data, Unit Costs (with dynamic divisor columns), NPV, and DCF/NPC Tree sheets — all from a single database query, not a chain of spreadsheet formulas.
- **Round-Trip Compatibility:** Analysts can download a .xlsm, make adjustments in Excel, and re-upload. The system preserves the VBA macros and validation logic through the round trip.

MiningIQ — From Pit Optimisation to Analyst Workbook

MiningIQ handles the other end of the pipeline: strategic mine planning outputs delivered as analyst-ready spreadsheets:

- **Analyst Spreadsheet Generator:** A 2,000+ line module that takes pit optimisation results or MILP scheduling scenarios and generates complete cost model scaffolds — 22 sheet sections including Project Setup, Schedule, Revenue, Equipment fleets, Activity costs, Labour, Admin, Capital, and Overheads. These plug directly into MineCost for detailed costing.
- **Simplified Analyst Cost Model:** A wizard-driven workflow that lets an analyst go from pit optimisation output to a fully imported MineCost scenario in under 30 minutes. The system auto-populates schedule and grade data, generates a starter spreadsheet with only the 8 core input sheets that need editing, and backfills 13 supporting tables with valid defaults. Round-trip compatible: download, edit in Excel, upload, run.
- **Drillhole Data Exports:** Geological data stored in the database is exported to formatted Excel workbooks matching industry-standard formats with proper unit conversions and coordinate system transformations.
- **Haulage Simulation Outputs:** Physics-based truck cycle time calculations (rimpull curves, anticipatory braking, corner speed reduction) are computed server-side and delivered as formatted results — not as Excel solver add-ins.
- **Enterprise Financial Reporting:** Full income statement, balance sheet, and cash flow generation with Bayesian risk analysis, multi-currency support, and scenario versioning. The financial model lives in a database; the board paper is generated from it.

The Pattern in Every Module

Database stores the truth. Python does the calculation. Excel (or Streamlit) delivers the result. This is the pattern your team will learn to build.

5. The Bigger Picture: Custom Solutions vs. Proprietary Software

There is a deeper argument behind this workshop that goes beyond spreadsheets. The mining industry spends hundreds of millions of dollars annually on proprietary software licences for pit optimisation, truck simulation, drillhole management, and scheduling tools — and much of that spend buys generality that individual operations don't need.

The Generality Tax

Proprietary pit optimisation and scheduling tools must handle every commodity, every mining method, every jurisdiction, and every edge case. That generality is expensive to build, expensive to licence, and it makes the software slow and complex. A site engineer working a single copper-gold open pit doesn't need a tool that also handles underground coal, iron ore beneficiation, and alluvial diamond recovery. They need a fast, focused tool for their operation.

What IMC Has Proven

Using the same open-source tools taught in this workshop, IMC Mining has replicated core functionality from several major proprietary platforms — not as toy demos, but as production systems processing real mine data:

Proprietary Category	What IMC Built	Performance
Pseudo-flow pit optimisation	FlowOpt: Hochbaum's Pseudo-Flow algorithm with implicit structure arc generation. Built in under 2 hours of development time using AI-assisted coding.	7 seconds for over 50M structure arcs that historically took over 45 minutes. Same optimisation quality as costly systems, orders of magnitude faster.
Pit optimisation + scheduling	Bienstock-Zuckerberg (BZ) Lagrangian decomposition hooked into FlowOpt's Pseudo Flow output, adding the 4th dimension of time to the pit optimisation.	Integrated pit-schedule optimisation that most operations run as two separate, disconnected steps with different vendors.
Truck simulation	Accurate haulage physics engine: rimpull/retarder curves, anticipatory braking, corner speed reduction, road-class rolling resistance. Supports OEM truck specs (Komatsu, CAT).	Real-time cycle time calculation across 40+ routes. No per-seat licence required. Linked to design shells and digitised haul routes and flows to cost model.
Drillhole data management	Direct laboratory API integration for data import, plus 7 automated QA/QC checks (standards, blanks, duplicates, precision, grade-length curves) that run continuously, not as manual spot-checks.	Automated QA/QC that catches errors in hours, not months. Export to all major geological modelling platforms.
Simplified cost modelling	Analyst Cost Model wizard: auto-populates from pit optimisation results, generates a starter spreadsheet with 8 editable core sheets, backfills 13 supporting tables. Upload and run.	From optimisation output to a fully imported financial scenario in under 30 minutes. Activity-based costing for rapid what-if analysis.

Why This Is Now Possible

Two things have changed that make this achievable for site engineering teams, not just software companies:

- **Open-source algorithms are production-ready:** Pseudo-flow pit optimisation, Bienstock-Zuckerberg scheduling, Monte Carlo simulation, Bayesian inference, and physics-based haulage modelling are all implementable using publicly available algorithms and libraries (NumPy, SciPy, Numba, PyMC3, scikit-learn). The mathematical foundations are well-documented and freely available. You do not need to licence them.
- **AI removes the programming bottleneck:** The barrier was never the algorithm — it was the programming skill to implement it. With AI-assisted coding, a mining engineer who understands the domain can build a working pit optimiser, haulage model, or cost engine without being a professional software developer. The engineer brings the domain knowledge; AI brings the code scaffolding. As Section 7 explains, the engineer must still verify every output — but the heavy lifting of writing code is no longer the constraint.

What This Means for Your Team

The mining industry doesn't need more skilled programmers. It needs skilled engineers and geologists who can create custom solutions for their particular operation. Solutions that are fast because they only solve the problems that matter to your site. Solutions that are transparent because your team built them and understands every assumption. Solutions that are free from per-seat licensing because they're built on open-source foundations.

This workshop teaches your team how to start building those solutions. Not in theory — by actually building them, using the same tools and patterns that IMC used to build a 7-second pit optimiser, a physics-based truck simulator, an automated QA/QC pipeline, and a 30-minute analyst cost model workflow.

6. Who Should Attend

Audience	Current Reality	What They'll Gain
Mine Planning Engineers	Running pit optimisation tools, then copying results into Excel for NPV comparison. Optimisation churn without risk-adjusted decision framework.	Risk-led scenario comparison, automated NPV pipelines, 30-minute analyst cost models, surrogate models for real-time cut-off guidance.
Technical Services / Cost Engineers	Maintaining interlinked Excel cost models. Single-person dependency. Manual reconciliation between planning and finance.	Database-backed cost engines that generate the same Excel outputs their stakeholders expect. The MineCost pattern: compute in Python, deliver in .xlsm.
Geologists & Engineers	Manual QA/QC spot-checks. Data trapped in Access databases and legacy formats.	Automated QA/QC pipelines, AI-powered document analysis, export to major geological modelling platforms.
Heads of Tech Services	Concerned about key-person risk, audit trail gaps, and the cost of getting a number wrong.	Strategic understanding of the spreadsheet-to-database transition. Live demo of AI auditing existing workbooks. Attend Day 1 for leadership alignment.

7. Workshop Structure — Day by Day

The workshop is modular: a 3-day core track covers foundations through deployment, with optional Days 4–5 for deeper capability. Every half-day includes a hands-on lab.

Day 1: Foundations — See the Problem, Start the Solution

Morning — AI-Powered Spreadsheet Audit

- Live demo: take an existing mine cost spreadsheet and run an AI audit using Cowork/Windsurf. Watch it deconstruct every formula, map cross-sheet references, and identify errors and disconnects.
- Discussion: what this reveals about the spreadsheets your team relies on today
- Environment setup: Python 3.11+, PyCharm + AI coding assistants, Git version control
- Claude API: prompt patterns for code generation, data wrangling, and spreadsheet analysis

Afternoon — Mining Data Essentials

- Block models, tidy data principles, and why Excel breaks at mining scale
- Lab 1: Exploratory Data Analysis with Claude & Pandas — load a real block model, profile grades, visualise spatial distributions. Compare: how long would this take in Excel?
- Lab 2: Statistical modelling warm-up — distributions, correlations, and variable relationships in mining datasets using NumPy & Pandas

Day 2: Decisions — Risk Models & Interactive Dashboards

Morning — Bayesian Risk Modelling

- Why single-point NPV estimates are dangerous. Expert probabilities, asymmetric flexes, and the case for risk-adjusted decisions.
- Deterministic enumeration vs. Monte Carlo: when to use each
- Lab 3: Build a first-principles risk model — define input distributions, run simulations, extract P10/P50/P90 and 'Prob. ≥ Base' metrics. Uses the same PyMC patterns as MiningIQ's enterprise finance module.

Afternoon — Interactive Visualisation & Excel Generation

- Building dashboards with Streamlit: sliders, filters, and real-time recalculation
- Generating Excel outputs with openpyxl: formatted workbooks, VBA macros, embedded charts. The MineCost ScenarioExporter pattern.
- Lab 4: Build an interactive risk dashboard AND generate a formatted Excel summary workbook from the same data source. Two outputs, one source of truth.
- Demo: The 30-minute analyst cost model — watch a pit optimisation output flow through a wizard into a starter spreadsheet, edit a handful of cost assumptions, upload, and get a full NPV and cashflow analysis. This is the MiningIQ Analyst Model workflow that was built based on client feedback for rapid what-if scenarios.

Day 3: AI Tools & Cloud Deployment

Morning — AI-Powered Document Analysis with Cowork

- Live demo: Cowork ingesting a directory of mine documents — SOPs, geotech reports, cost assumptions — and answering questions about them with context
- Hands-on: participants use Cowork and Windsurf on their own documents to extract insights, cross-reference data, and identify inconsistencies
- Lab 5: Use AI tools to audit a complex interlinked cost model, produce a documented overview of every formula and assumption, and flag potential errors

Afternoon — Cloud Deployment

- Docker containers and AWS EC2/S3 deployment paths
- Lab 6: Deploy your Streamlit dashboard and Excel generator to AWS EC2 so the whole team can access it
- Cost controls, IAM permissions, and security basics for mining companies

Days 4–5: Extended Track (Optional)

Day 4 — Scale & Reliability

- Data engineering with Parquet: columnar storage for large block models and time-series data
- Security deep-dive: AWS Parameter Store, secrets management, data residency
- Governance: versioning, audit trails, and the case against “final_v3_REAL_final.xlsx”

Day 5 — Capstone Build Day

Teams choose one capstone project and build a deployable tool in a single day:

- **Surrogate Model:** Train an ML model on optimisation runs to build a real-time cut-off grade assistant. Same patterns as MiningIQ’s FlowOpt module.
- **Automated QA/QC Pipeline:** Build an end-to-end drillhole data pipeline with automated quality checks (standards, blanks, duplicates) and formatted Excel reporting. The MiningIQ drillhole pattern.
- **Risk-Led Financial Model with Excel Output:** Build a full model that ingests schedule data, applies Bayesian flexes, calculates Expected NPV with P10/P50/P90 — and generates a formatted .xlsm workbook with VBA validation. The complete MineCost pattern, end to end.

8. The AI Reality Check — Why Engineers Still Matter More Than Ever

This workshop does not pretend that AI will write perfect code for mining engineers. It won’t. What it will do is remove the heavy lifting — scaffolding a cost model, wrangling a block model into tidy format, generating boilerplate data pipelines — so that engineers can focus on what actually matters: making sure the numbers are right.

What We’ve Learned Building MiningIQ and MineCost with AI

Across thousands of AI-assisted code solves using Claude, Gemini, and OpenAI, we have found that over 50% require careful manual correction before they are production-ready. The AI always wants to find a solution — even when it doesn’t have enough context to find the right one.

What AI Gets Wrong

The failure modes are consistent across every major AI coding tool. Understanding them is essential for any engineer who will use AI in their workflow:

- **Hardcoded values:** AI will frequently insert literal numbers where a variable or database lookup should be. A discount rate of 8% appears as 0.08 in the code instead of reading from the project's configuration. A gold price of $1,950/oz is baked into a calculation instead of pulling from the price deck. These are silent errors — the code runs, the output looks reasonable, but the assumption is frozen and wrong.
- **Fallback defaults that mask failures:** When AI encounters a missing value or an edge case, it will often insert a "reasonable" default rather than raising an error. A missing recovery rate becomes 95%. A null ore density becomes 2.7 t/m³. The code runs without complaint, but the output is built on fabricated data. In mining, where a wrong assumption can cascade through an entire NPV calculation, this is dangerous.
- **Context errors:** AI struggles with domain-specific context. It will apply a surface mining cost structure to an underground operation. It will assume annual discounting when the model uses mid-year convention. It will treat all tonnes as dry tonnes when the process requires wet. These errors require a mining engineer to catch — no amount of prompt engineering eliminates them entirely.
- **Confident but wrong:** AI does not say "I'm not sure." It produces code that looks professional, is well-commented, and runs without errors — but silently implements the wrong logic. A cut-off grade calculation that ignores stockpile costs. A haulage model that doesn't account for road gradient. A revenue calculation that double-counts by-product credits. The code looks right. The answer is wrong.

What This Means for Engineers

The biggest shift for mining engineers going forward is not in writing code — AI can handle most of the heavy lifting there. The shift is in verification. The engineer's role becomes: define the problem clearly, let AI generate a first draft, and then carefully audit every assumption, every hardcoded value, every fallback default, and every domain-specific calculation to ensure correctness.

This is a skill that must be taught. It is not intuitive. Engineers who are new to AI-assisted development tend to either trust the output completely (dangerous) or reject AI entirely (wasteful). The workshop teaches the middle path: use AI aggressively for scaffolding and acceleration, but treat every output as a draft that requires engineering review.

How the Workshop Addresses This

- **Day 1 — Prompt discipline:** Participants learn how to give AI sufficient context to reduce (not eliminate) errors. Specific prompt patterns for mining calculations, including providing units, referencing data sources, and specifying what should NOT be hardcoded.
- **Day 2 — Verification patterns:** Every lab includes a "verify the AI" step where participants systematically check AI-generated code for hardcoded values, default assumptions, and domain errors. This becomes second nature by Day 3.
- **Day 3 — AI tool verification:** The Cowork document analysis lab teaches participants to cross-check AI-generated insights against source data. Participants learn to spot when AI tools summarise incorrectly or miss context, building the same sceptical eye they need when reviewing AI-generated code.

- **Day 5 — Real-world capstone:** Capstone projects are graded not just on whether the code runs, but on whether the assumptions are correct. Teams present their verification process alongside their tool.

9. No Black Boxes — Every Solution Must Be Open to the Team

The spreadsheet problem and the AI problem share the same root cause: knowledge trapped in one person's head. When a cost model lives in a single engineer's Excel file, the team is one resignation away from a black box. When an engineer builds a Python solution but keeps it on their laptop with no documentation and no version control, the problem is exactly the same — just in a different language.

> **IMC's Non-Negotiable Rule**
>
> We do not allow black box solutions. Every line of code built during the workshop — and every tool participants build afterward — must be stored in a shared repository (Bitbucket, GitHub, or equivalent) where every team member with appropriate security clearance can access, read, and modify it. Mining companies should not accept individual engineers creating black boxes, whether in Excel or Python.

Version Control as a Core Skill

Day 1 of the workshop includes Git and Bitbucket setup for a reason: version control is not optional. It is the foundation that makes everything else safe. With a shared repository, every change is tracked, every assumption is visible, and every team member can see exactly what changed, when, and why.

- **Full team visibility:** Every engineer on the team has access to the codebase. No single person owns a critical calculation. If someone is on leave, their colleagues can review, run, and modify the code.
- **AI-powered documentation on demand:** If an engineer departs, the remaining team faces a code handover — but with modern AI tools, this is no longer the nightmare it once was. Any team member can point Claude, Copilot, or Windsurf at the repository and ask: "What does this code do? Document every module, every assumption, every input and output." Within minutes, the AI produces a comprehensive overview that would have taken days to write manually. The code is no longer a black box; it's instantly readable.
- **Change tracking and audit trail:** Git provides a complete history of every modification. When a board paper asks "when did the discount rate change from 8% to 10%?", the answer is in the commit log — with the author, the date, and the reason. Try getting that from a spreadsheet.
- **Branch-and-review workflow:** Engineers can experiment on branches without affecting the production model. Changes are reviewed by peers before merging. This is the same workflow used by every major software company and is simple to teach.

Security Without Silos

Openness does not mean no security. The workshop teaches proper access control: repository permissions, branch protection rules, and role-based access so that the right people can see and modify the right code. The principle is simple: open to the team, controlled by policy, auditable by management. This is how IMC operates MiningIQ and MineCost internally, and it is how we train workshop participants to operate.

10. Data Security — An Honest Conversation

If an engineer gives an AI tool access to the operating budget for a mine, is that data secure? This is a question that every mining company must ask — and one that the workshop addresses directly, because getting it wrong has serious consequences.

What You Need to Know

AI tools like Cowork, Windsurf, Claude, and Copilot process data by sending it to cloud servers. The security of that data depends entirely on your subscription plan, your provider's data retention policies, and whether your organisation has negotiated enterprise-grade protections. The details differ by provider and change frequently, but the core questions are always the same:

- **Is your data used for model training?** On consumer plans, it may be. On enterprise and commercial plans (Claude for Work, Claude Enterprise), major providers explicitly exclude your data from training. This distinction matters enormously for proprietary mining data.
- **How long is your data retained?** Retention periods range from zero (with enterprise zero-data-retention agreements) to 30 days or longer on standard plans. For sensitive financial models and production data, your organisation needs to understand exactly where the data goes and how long it persists.
- **Is there an audit trail?** Some AI tools do not yet capture activity in compliance logs or data exports. If your organisation operates under regulatory requirements, this is a gap that must be assessed.
- **What about on-premises alternatives?** For the most sensitive data, the workshop covers deployment patterns using AWS with proper IAM controls, data residency configurations, and security boundaries — keeping computation within your organisation's cloud tenancy rather than routing through third-party AI services.

How the Workshop Addresses This

We do not gloss over data security. The workshop includes a dedicated session on establishing clear policies for your team:

- **Classification framework:** Which data can be processed through cloud AI tools, and which cannot. Operating budgets, grade control data, and board-level financial models may require different handling than public geological reports or generic SOP documents.
- **Enterprise plan requirements:** What to look for in your AI provider's terms — data retention policies, training exclusions, zero-data-retention options, and compliance certifications.
- **Anonymisation techniques:** How to sanitise datasets for use in AI-assisted analysis while preserving the analytical value. Workshop labs use anonymised sample data by default, and we teach participants how to prepare their own data safely.
- **Self-hosted deployment:** For teams that need full control, the cloud deployment labs (Day 3 afternoon, Day 4) teach participants to run their tools on their own AWS infrastructure, keeping sensitive data within the organisation's security perimeter.

The Practical Position

AI tools are extraordinarily powerful for mining engineering — but they are not yet a black box you can trust with your most sensitive data by default. The workshop teaches engineers to use them aggressively where appropriate, and to know exactly where the boundaries are for their organisation. This is a governance decision, not a technology decision, and it must be made explicitly.

11. What Makes This Workshop Different

1. We've Already Built What We're Teaching

Every lab exercise is derived from production code. The Excel generation uses the same openpyxl + VBA macro embedding as MineCost's Scenario Exporter. The Bayesian risk model uses the same PyMC3 patterns as MiningIQ's enterprise finance module. The QA/QC pipeline mirrors MiningIQ's drillhole module. Participants are learning patterns that have been battle-tested across real mining projects.

2. You Keep Spreadsheets — But Fix Them

This is not a "spreadsheets are bad" workshop. Spreadsheets are an excellent delivery format. The workshop teaches participants to generate them programmatically from trusted data sources — with VBA macros, dropdown validation, conditional formatting, and embedded charts — so that the spreadsheet your manager opens looks exactly like what they expect, but the numbers behind it are auditable, versioned, and correct.

3. AI as Audit Tool, Not Replacement

The workshop introduces AI as something practical and immediate: a tool that can read your existing spreadsheets and tell you what they're doing, where they might be wrong, and how they connect to each other. This isn't aspirational — tools like Cowork and Windsurf can do this today. Day 1 starts with a live demonstration on a real mine cost spreadsheet.

4. Domain-Specific, Not Generic

The datasets are block models, not iris flowers. The cost models are WBS-driven equipment fleet calculations, not generic spreadsheets. The risk models use asymmetric flexes and expert probabilities, not textbook normal distributions. Every example is grounded in how mining companies actually make decisions.

	Generic AI/Python Training	IMC Workshop
Datasets	Iris, Titanic, toy examples	Block models, cost trees, equipment fleets, drilling data
Instructors	Software trainers	Mining software engineers who build production systems
Spreadsheet story	"Excel is dead, use Python"	"Keep Excel as your delivery layer, but compute properly"
Post-workshop	slide deck	Working tools: risk dashboard, Excel generator, QA/QC pipeline, cloud deployment
AI integration	ChatGPT demos	Spreadsheet audit with Cowork, AI-assisted coding, document analysis, API integration
Backed by	Generic curriculum	MiningIQ + MineCost production codebases

12. Technology Stack

Participants work with a curated, production-proven technology stack:

Layer	Technologies	Production Use
Core	Python , Git, PyCharm, AI assistants	Foundation of MiningIQ and MineCost
Data	Pandas, NumPy, SciPy, Parquet	MiningIQ block model processing, MineCost cost engine
Risk & ML	PyMC3, Monte Carlo, scikit-learn	MiningIQ enterprise finance risk module
AI Tools	Cowork, Windsurf, Claude API, Gemini	Document analysis, spreadsheet audit, AI-assisted coding
Excel Generation	openpyxl, VBA macro embedding	MineCost ScenarioExporter, MiningIQ Analyst Spreadsheet Generator
Dashboards	Streamlit, Plotly	MiningIQ financial reporting and geological model dashboards
Deployment	Docker, AWS EC2/S3, IAM	Production stack for both platforms

13. Delivery Options & Pricing

Option	Duration	Format	Ideal For
Core Workshop	5days	On-site	Teams wanting the spreadsheet-to-database transition with immediate practical output
Executive Briefing	Half day	On-site or remote	Leadership alignment: live demo of AI spreadsheet audit, strategic case for the transition

Pricing is available on request and depends on group size (recommended 8–16 participants), location, and whether client-specific datasets and spreadsheets will be incorporated into the labs.

All workshop materials, code repositories, templates, and Excel generation scripts remain with the client.

14. The Real Cost of AI Tooling

> **⚠ Reality Check**
>
> Production-grade AI tooling is not free. IMC currently spends over $1,000 per week across Anthropic (Claude), Windsurf, Gemini, and OpenAI subscriptions. Engineers planning to adopt these tools need to understand the cost implications and secure appropriate budgets before committing.

A common misconception: many engineers assume AI tools are like the free tier of ChatGPT — a convenient assistant at no cost. The reality is starkly different. The AI coding tools, large context windows, and enterprise features demonstrated in this workshop all sit behind paid tiers that add up quickly across a team.

For a team of engineers adopting AI-assisted development, realistic budgets are $200–500 per user per month across multiple platforms. This covers:

- **AI coding assistants** — Windsurf Pro, GitHub Copilot, or Cursor ($20–60/month each)
- **Large language model APIs** — Anthropic Claude, OpenAI GPT, Google Gemini ($20–200/month depending on usage)
- **Enterprise desktop tools** — Claude Pro/Max, ChatGPT Plus/Team ($20–200/month per seat)
- **Hosting and infrastructure** — Cloud compute, databases, CI/CD pipelines (variable)

Governance is non-negotiable. Without controls, costs spiral quickly. Individual engineers signing up for ad-hoc subscriptions leads to duplicated tools, zero visibility on spend, and no ability to assess ROI. The workshop covers practical governance:

- **Management sign-off** — AI tooling budgets must be approved like any other operational expenditure
- **Centralised procurement** — one team manages subscriptions, not individual engineers
- **Per-project allocation** — budgets allocated by project with clear deliverables, not open-ended exploration
- **Usage monitoring** — regular review of API usage dashboards and seat utilisation
- **Tiered access** — not every engineer needs the most expensive tier; match tools to roles and tasks

The workshop includes a practical session on setting up cost controls, monitoring dashboards, and building the business case that engineers need to take back to management.

15. Prerequisites

- A laptop with admin rights to install software (Python, PyCharm, Git)
- Basic familiarity with mining terminology (block models, cut-off grades, NPV)
- Some exposure to Python is helpful but not required — Day 1 starts from environment setup
- Optional but highly recommended: bring your own spreadsheets (anonymised cost models, schedule summaries) for the Day 1 AI audit exercise

IMC will provide a pre-workshop checklist and setup guide two weeks before the engagement.

16. Next Steps

- Expression of Interest: Reply to confirm interest and preferred dates
- Scoping Call: 30-minute call to understand your team's current tools, spreadsheets, and objectives
- Proposal: Tailored proposal with pricing, logistics, and optional client-data integration
- Delivery: Workshop delivered on-site at your facility

Contact: Stewart Lewis — slewis@imcm.au

IMC Mining