JUPITER PRIME

Consulting • Training • Advisory

جوبيتر برايم للاستشارات والتدريب

COURSE BROCHURE

Advanced AI Engineering for Technical Professionals Training Course

13 – 24 September 2027
Munich

jupiterprime.uk

info@jupiterprime.uk

Registered in the United Kingdom and Saudi Arabia

Advanced AI Engineering for Technical Professionals Training Course

Ref: 42_1071 **Date:** 13 – 24 September 2027 **Location:** Munich **Fees:** 11100 GBP

AI Engineering for Technical Professionals is a 2-week advanced training course for engineers, IT professionals and technical specialists who will build and document a deployable AI solution design for a real technical problem in their organisation. Prototype notebooks are easy to produce, but production systems that ingest live data, call language models and stay accurate under change demand disciplined engineering. Jupiter Prime Training runs this AI Engineering for Technical Professionals course for hands-on technical staff.

Why AI Engineering for Technical Professionals Matters Now

Three shifts shape this discipline: foundation models moving AI from bespoke research into everyday software components, data volumes from plant, network and customer systems outgrowing manual pipelines, and operations teams expecting models to be versioned, tested and monitored like any other service. An organisation that engineers AI well ships reliable features and reuses components across projects; one that does not accumulates fragile prototypes, silent model decay and unexplained failures. The course aligns engineering controls with ISO/IEC 42001, the AI management system standard, and the NIST AI Risk Management Framework NIST AI RMF.

Who Gains Most from AI Engineering for Technical Professionals

- Specialists who design and maintain data ingestion, transformation and feature pipelines
- Specialists who train, tune and evaluate supervised and deep learning models
- Specialists who integrate language models, retrieval and APIs into applications
- Specialists who run infrastructure, deployment and observability for AI services
- Specialists who translate operational and engineering problems into AI requirements

A platform or engineering squad gains most when its data engineers, model developers and operations staff attend together, because a deployable AI system only works when pipelines, models and serving infrastructure are designed against the same contracts and monitoring signals.

AI Engineering for Technical Professionals Capabilities

Participants work through the full engineering lifecycle, from raw data to a monitored service, writing and reviewing the artefacts each stage requires.

- Build reproducible data pipelines with validation, lineage and feature stores — data platform owners
- Apply supervised learning and deep learning architectures to structured, image and text problems — engineering leads
- Design language model integrations using retrieval-augmented generation, tool calling and guardrails — product owners
- Test models and generative systems with offline benchmarks, evaluation harnesses and human review — quality assurance
- Specify MLOps pipelines for training, versioning, deployment and rollback — operations and release managers
- Calculate serving cost, latency and accuracy trade-offs for architecture choices — technical budget holders

Key AI Engineering for Technical Professionals Terms

- MLOps — practices and tooling that automate training, testing, deployment and monitoring of machine learning models as reliable software services.
- Feature store — a managed repository that serves consistent, versioned input features to both model training and live inference.
- Retrieval-augmented generation — a pattern where a language model answers using passages retrieved from approved documents, improving accuracy and traceability.
- Embedding — a numeric vector representing the meaning of text, images or records so similar items sit close together.
- Evaluation harness — an automated test suite that scores model or prompt outputs against reference cases and quality metrics.
- Model drift — degradation of a deployed model's performance as live data or relationships move away from training conditions.

AI Engineering for Technical Professionals Day by Day

Day 1: The AI Engineering Landscape

- Classical, Deep and Foundation Model Approaches — Matching Method to Problem
- AI System Reference Architecture — Data, Model, Serving and Monitoring Layers
- Build, Fine-Tune or Call an API — Engineering Trade-Offs
- Technical Problem Framing Canvas — Inputs, Outputs and Success Metrics
- Manufacturing and Telecom Use Case Walkthroughs — Sector Patterns

Day 2: Data Pipelines and Feature Engineering

- Batch and Streaming Ingestion Patterns — Pipeline Topology Choices
- Schema Validation and Data Quality Tests — Catching Bad Inputs Early
- Data Lineage and Versioning — Reproducible Training Sets
- Feature Store Design — Training and Serving Consistency
- Labelling Strategies and Weak Supervision — Building Training Data at Scale

Day 3: Supervised Learning in Practice

- Linear, Tree-Based and Ensemble Models — Baseline Selection
- Cross-Validation and Hyperparameter Search — Robust Model Choice
- Handling Imbalanced and Missing Data — Production Realism
- Interpretability With SHAP and Partial Dependence — Engineering Diagnosis
- Experiment Tracking — Comparing Runs and Artefacts

Day 4: Deep Learning Architectures

- Convolutional Networks for Vision Inspection — Defect Detection
- Sequence Models and Transformers for Signals and Text — Architecture Choice
- Transfer Learning and Fine-Tuning — Reusing Pretrained Weights
- Training Stability, Regularisation and Mixed Precision — Compute Efficiency
- GPU Resource Planning — Training Time and Cost Estimates

Day 5: Language Model Integration

- Prompt Templates and Structured Outputs — Reliable Application Responses
- Tool Calling and Function Interfaces — Connecting Models to Systems
- Model Selection Across Hosted and Open-Weight Options — Cost and Control
- Guardrails, Content Filters and Fallbacks — Safe Application Behaviour
- Token Budgets and Caching — Latency and Spend Management

Day 6: Retrieval-Augmented Generation Systems

- Document Chunking and Metadata Strategies — Retrieval Quality
- Embedding Models and Vector Indexes — Similarity Search Design
- Hybrid Search and Re-Ranking — Precision of Retrieved Context
- Citation and Grounding Checks — Traceable Answers
- Access Control on Retrieved Content — Permission-Aware Responses

Day 7: Evaluation and Testing of AI Systems

- Offline Metrics for Classification, Regression and Ranking — Release Criteria
- Generative Output Evaluation Harnesses — Faithfulness and Relevance Scores
- Human Review Protocols and Golden Datasets — Quality Baselines
- Robustness and Adversarial Testing — Edge Case Coverage
- A/B and Shadow Deployment Tests — Live Performance Evidence

Day 8: MLOps and Deployment

- CI/CD Pipelines for Models and Prompts — Automated Promotion
- Model Registry and Versioning — Controlled Releases and Rollback
- Containerised Serving and Autoscaling — Throughput and Latency
- Edge and On-Premise Deployment Options — Data Residency and Connectivity
- Infrastructure as Code for AI Workloads — Repeatable Environments

Day 9: Monitoring, Drift and Operational Risk

- Prediction and Data Drift Monitoring — Retraining Triggers
- Observability for Language Model Applications — Traces, Costs and Errors
- Incident Runbooks for AI Services — Diagnosis and Recovery
- ISO/IEC 42001 and NIST AI RMF Engineering Controls — Model Risk Evidence
- Technical Documentation and Model Cards — Handover and Audit

Day 10: Practising the Deployable AI Solution Design

- Problem Framing Applied to a Participant's Technical Challenge — Scope Decision
- Pipeline and Model Design Applied to Its Data — Architecture Choice
- Evaluation Plan Applied to the Proposed System — Release Criteria
- Deployment and Monitoring Applied to Target Infrastructure — Operating Model
- Deployable AI Solution Design Assembly — Architecture, Tests, Runbooks and Documentation

Applying AI Engineering for Technical Professionals After the Course

- Add schema validation and data quality tests to one existing training pipeline and publish a lineage record for its current model version.
- Build an evaluation harness with a golden dataset for one language model feature and run it on every prompt or model change.

AI Engineering for Technical Professionals Questions FAQ

What is AI engineering for technical professionals, and why does it matter now?

It is the discipline of building, deploying and operating machine learning and language model systems as dependable software. It matters now because foundation models and growing data volumes push AI into core products, where fragile prototypes create outages and hidden errors.

Which roles gain most from AI engineering for technical professionals training?

Experienced technical specialists gain most: data engineers, software and machine learning engineers, platform and operations staff. You benefit if you build pipelines, train models, integrate language models or keep AI services running.

How does advanced AI engineering for technical professionals differ from an AI for business course?

It is hands-on and technical rather than strategic. A business course covers use cases, value and oversight, while this course builds pipelines, trains deep learning models, integrates retrieval-augmented generation and sets up MLOps, testing and monitoring.

What do you bring back from AI engineering for technical professionals training?

You bring back a deployable AI solution design for a technical problem. It documents the architecture, data pipelines, model choice, evaluation plan, deployment approach, monitoring and runbooks your squad can implement.

AI Engineering for Technical Professionals: The Way Forward

The deployable AI solution design first informs whether a technical problem proceeds to build and on which architecture. Data engineers use it to set pipeline contracts, model developers to fix evaluation criteria, and operations staff to provision serving and monitoring. After the first production release, revisit the evaluation harness, since real user inputs expose cases the original tests missed.