

ARTIFICIAL INTELLIGENCE

ADVANCED

5 DAYS

Forward Deployed Engineering: Delivering Enterprise AI in Customer Environments

A five-day role-conversion programme for organisations building forward deployed capability from existing delivery talent. Covers discovery, solution selection, messy client environments, enterprise integration, prototyping to production, security review, executive communication and adoption. Ends in a defended end-to-end engagement simulation. Assumes existing engineering competence and prior LLM application experience.

Why Forward Deployed Engineering?

Role conversion, not career change

Built for organisations turning solution architects and delivery engineers into forward deployed teams.

Delivery skill, not just AI skill

Assumes engineering competence and teaches everything that surrounds it.

Constraints are the curriculum

Legacy systems, restricted access, compliance gates and stakeholder politics are the content, not caveats.

Ends in a defended engagement

Capstone runs a full client simulation, defended to a panel playing sceptical stakeholders.

What you'll be able to do

- ✓ Explain the forward deployed model and how it differs from solutions architecture and consulting
- ✓ Run customer discovery that separates the stated problem from the real one
- ✓ Select the right technical pattern, including deciding when AI is the wrong answer
- ✓ Scope engagements with measurable success criteria and honest risk assessment
- ✓ Work productively inside undocumented, legacy and access-restricted environments
- ✓ Integrate with enterprise data sources, identity systems and line-of-business platforms
- ✓ Prototype at speed without creating debt that blocks the production path
- ✓ Harden a prototype into a reliable, observable, cost-controlled service
- ✓ Prove a solution works against client acceptance criteria and defend the numbers
- ✓ Survive client security review, procurement and compliance assessment
- ✓ Communicate trade-offs credibly to both engineers and executives
- ✓ Manage scope change, blocked access and competing stakeholders
- ✓ Own adoption after go-live and measure real workflow impact
- ✓ Plan handover, exit and the productisation of repeated patterns

Who this is for

- Solutions architects moving from advisory to implementation ownership
- Delivery and consulting engineers in systems integrators and consultancies

- Pre-sales and solutions engineers taking on build responsibility
- AI engineers moving into customer-facing delivery roles
- Technical leads establishing an internal AI delivery function

Curriculum

01 The Forward Deployed Model

- What the role actually is, and how it differs from solutions architecture, consulting and pre-sales
- Why vendors and integrators are staffing embedded delivery teams
- Engagement archetypes: proof of value, pilot, production build, platform enablement
- Ownership of the outcome rather than the deliverable, and what that changes
- An honest look at where the role is real and where the title has been inflated
- **Hands-on:** map your own client portfolio to engagement archetypes

02 Customer Discovery and Problem Framing

- Stakeholder mapping: who decides, who blocks, who is never in the room
- Separating the stated problem from the underlying one
- Interview technique for eliciting workflow reality rather than aspiration
- Quantifying current-state cost, volume and pain before proposing anything
- Disqualifying bad use cases early and why that builds credibility
- **Hands-on:** run a discovery session against a live brief and produce a discovery document

03 Solution Selection and Pattern Choice

- Matching problem shape to technical pattern: retrieval, agents, classical approaches, plain software
- Recognising when AI is genuinely the wrong answer and saying so
- Framing cost, latency and accuracy trade-offs in business terms
- Build against buy against configure decisions inside a client estate
- **Hands-on:** propose and justify an architecture for the brief you scoped

04 Scoping, Success Criteria and the Engagement Plan

- Writing success criteria the client will accept and you can actually meet
- Timeboxing, milestone design and building in decision gates
- Risk assessment: technical, data, access, organisational and political
- What to promise, what to caveat and what to refuse
- Producing an engagement plan a client sponsor can approve

05 Working in Someone Else's Environment

- Getting productive fast in undocumented and unfamiliar codebases
- Legacy constraints, technical debt and systems nobody currently owns
- Restricted access, delayed credentials and working around blockers
- Air-gapped, VPC-isolated and sovereign deployment environments
- Respecting client conventions instead of importing your own
- **Hands-on:** orient in an unfamiliar codebase and produce an architecture note

06 Enterprise Data Access and Integration

- Connecting to CRM, ERP, ticketing, warehouses and internal APIs
- Authentication patterns and acting correctly on behalf of a user
- Inheriting existing permissions so a system cannot overshare
- Data quality reality: what the client believes against what the tables contain
- Event-driven integration and where webhooks beat polling
- **Hands-on:** integrate against a realistic set of messy enterprise sources

07 Rapid Prototyping Under Constraint

- Building fast without creating debt that blocks the production path
- Deciding deliberately what is throwaway and what is foundation
- Demo discipline: showing real behaviour rather than a happy path
- Managing expectations set by an impressive prototype
- **Hands-on:** ship a working prototype against the scoped brief

08 Building the Production Path

- Hardening a prototype: error handling, retries, fallbacks and degradation
- Observability, tracing and instrumentation the client team can operate
- Cost controls, quotas and preventing consumption surprises
- Performance, concurrency and realistic load in the client's context
- Deployment into the client's own pipelines and platform standards

09 Evaluation and Proving It Works

- Translating client acceptance criteria into a measurable evaluation suite
- Establishing a baseline so improvement can actually be claimed
- Golden datasets built from the client's real cases, not invented ones
- Presenting results honestly, including where the system underperforms
- Defending numbers to a sceptical technical audience
- **Hands-on:** build and present an evaluation against your prototype

10 Security Review, Compliance and Procurement

- What a client security review asks and how to be ready before it starts
- Data residency, retention, processing agreements and impact assessments
- Model, vendor and connector supply chain assessment
- Producing audit evidence: logs, decision records and control documentation
- Navigating procurement, legal review and the timelines they impose
- **Hands-on:** complete a security questionnaire for the system you built

11 Communicating with Technical and Executive Audiences

- Explaining trade-offs to engineers without condescension
- Explaining the same decision to an executive without distortion
- Handling the sceptical CTO and the over-enthusiastic sponsor
- Saying no, and saying not yet, without losing the engagement
- Written communication: status, decisions and escalations that get read
- **Hands-on:** present the same architecture decision to two different audiences

12 Managing Ambiguity, Scope and Politics

- Requirements that change because the client learned something
- Competing stakeholders with incompatible definitions of success
- Blocked access, absent decision-makers and stalled engagements
- Recognising a failing deployment early and what to do about it
- Escalation paths and when to involve your own leadership
- Protecting your own team from an unbounded engagement

13 Adoption Ownership

- Why go-live is the midpoint of the engagement rather than the end
- Enablement, documentation and training for the client's own team
- Feedback loops that capture real usage and real failure
- Measuring workflow impact rather than usage counts
- Feeding findings back into product and model roadmaps
- **Hands-on:** build a ninety-day adoption plan

14 Handover, Exit and Scaling the Model

- Knowledge transfer that survives your departure
- Runbooks, architecture decision records and operational documentation
- Deciding when the engagement should end
- Recognising repeated patterns and productising them
- Building an internal forward deployed practice: hiring, ramping and staffing
- Metrics for the practice itself: utilisation, time to value, expansion

15 Capstone: End-to-End Engagement Simulation

- Receive an ambiguous business brief with incomplete information
- Run discovery, frame the problem and propose an architecture
- Build a working system against a realistic constrained environment
- Produce evaluation results, a security response and an adoption plan
- Defend the engagement to a panel playing sceptical client stakeholders
- Receive structured feedback on both technical and delivery performance