Forward Deployed Engineer (FDE) Readiness Bootcamp: From Client Discovery to Production Deployment

Duration: 5 days/ 40 hours

Day 1: FDE Mindset, Customer Discovery and Problem Framing

Duration: 8 Hours
Concepts and Demonstrations: 1.5 Hours
Practical Activities: 6.5 Hours

Module 1: Understanding the Forward Deployed Engineer Role

- Evolution and purpose of the FDE role
- FDE versus traditional software engineer
- FDE versus solution architect and technical consultant
- Working at the intersection of:
    - Customer requirements
    - Product capabilities
    - Software engineering
    - Deployment and operations
- End-to-end FDE engagement lifecycle
- Characteristics of a successful FDE:
    - Technical depth
    - Customer empathy
    - Speed and adaptability
    - Ownership and accountability
    - Comfort with ambiguity
- FDE engagement scenarios and common challenges

Module 2: Business and Customer Discovery

- Understanding the customer's industry and operating model
- Stakeholder identification and stakeholder mapping
- Understanding current-state workflows
- Identifying operational pain points
- Distinguishing symptoms from root problems
- Asking effective technical and business questions
- Conducting discovery interviews
- Capturing functional and non-functional requirements
- Identifying assumptions, constraints and dependencies
- Recognising incomplete or conflicting requirements

Module 3: Problem Framing and Outcome Definition

- Converting business problems into engineering problem statements
- Creating clear problem and opportunity statements
- Defining scope and boundaries

- Prioritising requirements using value, risk and feasibility
- Defining success metrics and acceptance criteria
- Establishing baseline performance
- Identifying measurable KPIs
- Preparing a value hypothesis
- Creating an initial implementation backlog
- Managing uncertainty and changing priorities

Hands-On Labs

Lab 1: Customer Discovery Simulation

Participants receive an incomplete client requirement and conduct a simulated discovery meeting.

Activities include:

- Identifying stakeholders
- Preparing discovery questions
- Interviewing simulated business and technical stakeholders
- Documenting pain points
- Identifying missing information
- Capturing technical and business constraints

Lab 2: Problem-Framing Workshop

Participants convert the discovery information into:

- Current-state workflow
- Problem statement
- Target-state workflow
- Assumptions and constraints
- Business-impact hypothesis
- Success metrics
- Initial product backlog

Lab 3: Client Playback Presentation

Teams present their understanding back to the simulated client and receive feedback on:

- Clarity
- Accuracy
- Business alignment
- Questioning approach
- Handling ambiguity

Day 2: Solution Architecture, Data and Integration Engineering

Duration: 8 Hours
Concepts and Demonstrations: 1.5 Hours
Practical Activities: 6.5 Hours

Module 1: Translating Requirements into Architecture

- Moving from problem statement to solution architecture
- Identifying major solution components
- Functional and non-functional architecture
- Selecting an appropriate technology stack
- Build-versus-buy decisions
- Designing for rapid experimentation and future scale
- Monolithic, modular and microservice approaches
- Synchronous and asynchronous communication
- Event-driven solution patterns
- Handling legacy-system constraints
- Designing for extensibility and maintainability

Module 2: API and Enterprise Integration

- REST API fundamentals
- API contracts and interface design
- Authentication and authorisation
- Integrating external and enterprise systems
- API gateways and service boundaries
- Error handling and retries
- Timeouts and circuit-breaker concepts
- Rate limiting
- Idempotency
- API versioning
- Webhooks and event-based integrations
- Testing integrations using mock services

Module 3: Data Architecture and Processing

- Identifying source systems
- Structured and unstructured data
- Data ingestion and transformation
- Data validation and quality checks
- Relational and NoSQL storage selection
- Data models and schema design
- Batch versus real-time processing
- Data lineage and traceability
- Handling sensitive data
- Data access and retention considerations

Module 4: Security-by-Design

- Identifying security requirements during discovery
- Authentication versus authorisation
- Role-based access control
- Secrets and credential management
- Encryption in transit and at rest

- Secure API integration
- Input validation
- Audit logging
- Privacy and data-governance considerations
- Basic threat modelling

Hands-On Labs

Lab 1: Architecture Design Challenge

Teams design a solution for the business case identified on Day 1.

The architecture must include:

- User-facing application
- Backend service
- Integration layer
- Database or data platform
- Authentication
- Cloud deployment environment
- Logging and monitoring components

Lab 2: API Integration Lab

Participants:

- Consume a mock customer API
- Authenticate API requests
- Validate responses
- Handle errors and timeouts
- Transform received data
- Store processed data
- Expose a new API endpoint

Lab 3: Architecture Review Board

Each team presents its architecture and defends:

- Technology choices
- Scalability approach
- Security controls
- Integration strategy
- Data design
- Key trade-offs

Lab 4: Threat-Modelling Exercise

Participants identify:

- Assets
- Entry points
- Trust boundaries
- Potential threats

- Recommended mitigations

Day 3: Rapid Solution Engineering and AI-Assisted Development

Duration: 8 Hours
Concepts and Demonstrations: 1.5 Hours
Practical Activities: 6.5 Hours

Module 1: Rapid Prototyping for FDE Engagements

- Prototype versus proof of concept versus production solution
- Selecting the smallest valuable solution
- Vertical-slice development
- Prioritising high-risk technical assumptions
- Building for early customer feedback
- Managing technical debt during rapid delivery
- Reusable components and accelerators
- Feature flags and configuration-driven development
- Keeping prototypes production-aware

Module 2: Production-Quality Engineering Practices

- Clean and modular code
- Separation of concerns
- Configuration management
- Environment-specific settings
- Error-handling strategies
- Structured logging
- Input validation
- Unit and integration testing
- Code reviews
- Branching and version-control practices
- Documentation for maintainability
- Defining “done” for an FDE deliverable

Module 3: AI-Assisted Engineering

- Using AI coding assistants responsibly
- Generating code scaffolding
- Code explanation and refactoring
- Generating test cases
- API documentation assistance
- Debugging with AI tools
- Reviewing AI-generated code
- Preventing hallucinated dependencies and insecure code
- Protecting customer information when using AI tools
- Human validation and approval controls

Module 4: Applied AI Solution Patterns

Depending on the client's technology environment, this module can include:

- LLM-enabled application architecture
- Prompt design and structured outputs
- Retrieval-augmented generation
- Tool and function calling
- Connecting AI applications with enterprise APIs
- Context management
- Guardrails and output validation
- Evaluation datasets
- AI response quality measurement
- Human-in-the-loop workflows
- Cost and latency considerations

Hands-On Labs

Lab 1: Vertical-Slice Prototype

Teams build a working solution containing:

- User interface or API consumer
- Backend service
- Business logic
- Database integration
- External API integration
- Validation and error handling

Lab 2: AI-Assisted Development Challenge

Participants use an approved AI coding assistant to:

- Generate initial code scaffolding
- Refactor an existing component
- Create unit tests
- Identify a defect
- Improve technical documentation

Participants must document what was AI-generated and how it was validated.

Lab 3: AI-Enabled Feature

Teams add an AI-enabled capability such as:

- Document summarisation
- Knowledge assistant
- Natural-language query interface
- Classification or routing
- Recommendation support
- Automated report generation

Where AI is outside the client's FDE framework, this lab can be replaced with an advanced integration or data-processing challenge.

Lab 4: Code and Design Review

Teams review another team's solution for:

- Code quality
- Security
- Testability
- Maintainability
- Performance
- Correctness
- Production readiness

Day 4: Cloud Deployment, DevOps, Reliability and Production Support

Duration: 8 Hours
Concepts and Demonstrations: 1.5 Hours
Practical Activities: 6.5 Hours

Module 1: Production Deployment Fundamentals

- Development, testing, staging and production environments
- Application packaging
- Container fundamentals
- Dockerfile design
- Environment variables and secret handling
- Cloud deployment models
- Infrastructure-as-code concepts
- Deployment configuration
- Blue-green and rolling deployment concepts
- Release verification
- Rollback planning

Module 2: Continuous Integration and Delivery

- Automated build workflows
- Running tests in a delivery pipeline
- Code-quality checks
- Dependency and vulnerability scanning
- Artifact creation
- Environment promotion
- Approval gates
- Deployment automation
- Release notes and change documentation
- Post-deployment validation

Module 3: Observability and Reliability

- Logs, metrics and traces
- Structured application logging

- Health checks
- Service-level indicators
- Service-level objectives
- Dashboards and alerts
- Performance monitoring
- Capacity and scalability considerations
- Failure modes and graceful degradation
- Backup and recovery considerations
- Reliability and operational readiness checklist

Module 4: Troubleshooting and Incident Response

- Structured troubleshooting methodology
- Reproducing defects
- Isolating application, network and data issues
- Reading logs and stack traces
- API failure investigation
- Database-performance investigation
- Root-cause analysis
- Incident communication
- Workarounds versus permanent fixes
- Post-incident review
- Converting field issues into product feedback

Module 5: Customer Adoption and Operational Handover

- Deployment runbooks
- User documentation
- Support models
- Knowledge-transfer sessions
- Training customer teams
- Managing production acceptance
- Capturing feedback from users
- Driving adoption
- Escalation paths
- Handover to operations and support teams

Hands-On Labs

Lab 1: Containerisation Lab

Participants:

- Create a Dockerfile
- Containerise the application
- Configure runtime variables
- Manage secrets securely
- Run the complete solution locally
- Test container health

Lab 2: Deployment Lab

Teams deploy their application to a controlled cloud or Kubernetes sandbox.

Activities include:

- Build and package
- Configure environment
- Deploy the application
- Verify service connectivity
- Perform smoke testing
- Document rollback steps

Lab 3: Monitoring and Dashboard Lab

Participants configure:

- Application logs
- Health endpoints
- Basic metrics
- Monitoring dashboard
- Operational alerts

Lab 4: Production Incident Simulation

Each team receives one or more injected failures, such as:

- Authentication failure
- API timeout
- Incorrect configuration
- Database connectivity issue
- Performance degradation
- Invalid data
- Dependency failure

Teams must:

- Triage the incident
- Communicate status
- Identify the root cause
- Implement a fix
- Validate recovery
- Produce a short incident report

Day 5: End-to-End FDE Capstone and Deployment Readiness Assessment

Duration: 8 Hours
Concepts and Assessment Briefing: 2 Hours
Capstone and Evaluations: 6 Hours

Module 1: Managing an FDE Engagement

- Balancing business value and technical feasibility
- Managing scope under limited time
- Handling changing customer requirements
- Communicating delivery risks
- Prioritising production blockers
- Working with product and engineering teams
- Escalating appropriately
- Capturing reusable solution patterns
- Converting field feedback into product improvement
- Managing customer expectations
- Executive and technical communication

Module 2: Business Impact and Value Realisation

- Connecting the solution to customer outcomes
- Defining pre- and post-deployment measurements
- Measuring adoption and usage
- Evaluating operational improvements
- Identifying cost, time, quality or risk benefits
- Preparing business-impact reports
- Demonstrating ROI hypotheses
- Defining the next phase of implementation
- Creating a scale-up roadmap

Module 3: Capstone Client Scenario

Teams receive a new, time-bound customer scenario containing:

- Business background
- Initial requirements
- Stakeholder profiles
- Existing systems
- APIs and sample data
- Security constraints
- Deployment environment
- Conflicting priorities
- Production issue or change request

Capstone Activities

Stage 1: Discovery and Scoping

- Teams must:
- Conduct a simulated client interview
- Identify the core problem
- Define scope
- Identify assumptions and risks
- Define success metrics
- Prepare a prioritised backlog

Stage 2: Architecture and Planning

Teams produce:

- Solution architecture
- Integration design
- Data flow
- Security approach
- Deployment plan
- Test strategy

Stage 3: Build and Integration

Teams:

- Build a functional solution
- Integrate at least one external service
- Add validation and error handling
- Implement tests
- Add logs and health checks

Stage 4: Deployment and Incident Handling

Teams:

- Package and deploy the solution
- Perform smoke testing
- Respond to an injected production issue
- Resolve or mitigate the issue
- Update the operational documentation

Stage 5: Client Demonstration and Executive Playback

The final presentation must cover:

- Customer problem
- Proposed solution
- Working demonstration
- Architecture
- Security and reliability
- Deployment approach
- Business-impact metrics
- Risks and limitations
- Recommended next steps
- Scale-up roadmap