AI Engineering Handbook (EN)

By Florence J. and Rachel L.

Understanding AI Engineering Systems

Software engineering is undergoing a profound transformation.

Over the past few decades, software systems have been built primarily from deterministic programs. Engineers define system behavior by writing logic, control flow, and data structures. The core challenge of software engineering has been managing complexity: how to build scalable architectures, how to keep distributed systems stable, and how to maintain reliability as systems grow.

But with the rise of large language models, the basic composition of software systems is changing. More and more applications are no longer driven solely by code; instead, they are centered around models: models generate content, retrieval systems provide memory, tools extend capabilities, and evaluation systems measure quality. Software is no longer just executing instructions; it is collaborating with probabilistic models.

AI engineering is becoming the new software engineering.

If software engineering is the discipline of building logical systems, then AI engineering is the discipline of building cognitive systems. Engineers are no longer just writing programs; they are designing systems composed jointly of models, data, and software.

The goal of this book is to draw a field map for this new domain.

If AI engineering is the new software engineering, we need to answer a fundamental question:

How are modern AI systems actually built?

To answer this, the book breaks AI systems into five layers:

Foundations → Training → Models → Inference → Applications

From the fundamentals of Transformers and scaling laws, to model training and behavior shaping; from model architectures to inference systems; and then to agent systems that power complete AI applications — these layers together form the AI Engineering Stack.

Within this framework, many familiar concepts take on new meanings:

- LLM is a probabilistic simulator.
- RAG is the system's memory structure.
- Agents are the system's orchestration layer.
- Evaluation is testing for AI systems.

Building AI systems is, in essence, a continuous process of engineering trade-offs: latency vs quality, compute vs data, dense models vs MoE, and tool use vs model reasoning.

These choices are not isolated. Training strategy affects inference efficiency, model architecture determines system latency, and RAG design changes evaluation methodology. Every layer of an AI system influences the others.

This book is not just an interview question bank, and not a framework usage guide. It aims to build something more fundamental:

A systematic engineering perspective for AI engineering.

What you learn here is not just answers, but a way to understand how modern AI systems are built.

AI Engineering Stack

```
AI Engineering Stack

Applications
    Agents

Inference Systems
    Serving
    Compression

Model Architectures
    GPT
    MoE
    Multimodal

Training Systems
    Pretraining
    Mid-training
    Post-training

Foundations
    Transformers
    Tokenization
    Scaling Laws
```

Chapters