

AI-Driven Development — 30-Day Challenge

Task 2 —

Part A — Theory (Short Questions)

1. Nine Pillars Understanding

Q1: Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better?

AI development agents are better for repetitive tasks because they save your time. Tasks like setup, boilerplate code creation, and environment configuration are handled quickly by the AI. This frees the developer's mind so they can focus on architecture, design, and system-level thinking—roles that truly belong to a system architect.

Q2: How do the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer?

The Nine Pillars provide a structured system that includes TDD, SDD, AI CLI, agents, multi-model workflows, evaluation, and more. Combined, these create deep understanding across multiple domains. This transforms the developer from a simple coder into an M-Shaped professional who has strong expertise in 2–4 domains such as architecture, AI tools, testing, planning, etc.

2. Vibe Coding vs Specification-Driven Development

Q1: Why does Vibe Coding usually create problems after one week?

Vibe coding is done without planning. It feels fast at the beginning, but after a week, problems start appearing. Code becomes messy, duplication increases, bugs appear, and even the developer cannot understand what they wrote earlier. Future changes become nearly impossible.

Q2: How does Specification-Driven Development prevent those problems?

SDD focuses on creating clear specifications first—what to build, how it will work, and what rules it must follow. When specifications are clear, the code becomes clean, maintainable, and predictable. This reduces bugs, keeps the system scalable, and makes long-term development easier.

3. Architecture Thinking

Q1: How does architecture-first thinking change the role of a developer in AIDD?

Architecture-first thinking transforms the developer from a coder into a system designer. Instead of focusing on line-by-line code, the developer thinks about the entire system—its flow, layers, interactions, and scalability. This mindset elevates them to a system architect level.

Q2: Why must developers think in layers and systems instead of raw code?

Thinking in layers and systems is essential because modern AI-based applications are complex. If a developer focuses only on raw code, the system will break. Layers (Models → IDEs → Agents) provide clarity, make debugging easier, and simplify future updates.

Part B — (CLI prompt + output)

The screenshot shows a terminal window with the Gemini logo at the top. Below it, there's a tips section and a detailed description of email validation rules. The main area is a command-line interface where the user is prompted for input. The status bar at the bottom indicates the user is in INSERT mode and provides file navigation information.

Part C — Multiple Choice Questions

1. Purpose of SDD?

Answer: B — Clear requirements before coding begins

2. Biggest mindset shift in AIDD?

Answer: B — Thinking in systems and giving clear instructions

3. Failure of Vibe Coding?

Answer: B — Architecture becomes difficult to extend

4. Advantage of AI CLI agents?

Answer: B — Handle repetitive tasks so developers can focus on design and problem-solving

5. M-Shaped Developer?

Answer: C — Deep skills in multiple related domains

Reflection

This task demonstrates how the developer's role is evolving in the AI-Native era. By using the Nine Pillars, SDD, and AI Agents, developers shift from merely writing code to thinking in systems. As a result, they become M-Shaped professionals with strong, multi-domain expertise suitable for modern AI-driven development.