

# **Day 2 AIDD**

## **30-Day Challenge**

### **Submission**

**Name:** Syed Muhammad Adil Ahmed

**Number:** 00494243

**Class Slot:** Friday (6:00 PM to 9:00 PM)

**Instructor:** Sir Hamzah Syed

## **Part A — Theory (Short Questions)**

### **1. Nine Pillars Understanding**

- Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better for your growth as a system architect?  
→ Using AI Development Agents like Gemini CLI helps me grow as a system architect because they handle all repetitive setup tasks such as creating environments, folders, files, and installations. These tasks usually take time and distract from real planning. When agents automate them, I can focus more on architecture, design, and decision-making — the work that actually builds my long-term skill.
  
- How the Nine Pillars help a developer become an M-Shaped Developer?  
→ The Nine Pillars give developers a complete framework to learn, test, plan, write specs, and use agents effectively. Because of this system, I can work across multiple domains — coding, testing, cloud, specs, and AI collaboration. This helps me grow from a single-skill developer into an M-Shaped Developer with deep knowledge in several areas.

### **2. Vibe Coding vs Specification-Driven Developmen**

- Why Vibe Coding creates problems after one week
  - Vibe Coding creates problems because it has no planning, no structure, and no clear specifications. After a few days, the code becomes confusing, hard to extend, and impossible to debug. There is no history or logic behind design decisions, so the project breaks quickly.
- How SDD prevents those problems?
  - Specification-Driven Development prevents these issues by defining everything before coding starts. Clear specs guide the AI and the developer, making the project structured, maintainable, and easy to extend. Because the system is planned first, no part of the project becomes messy later.

### 3. Architecture Thinking

- How architecture-first thinking changes the developer's role?
  - Architecture-first thinking shifts the developer from a code writer to a system designer. Instead of focusing on syntax, I think about components, behaviors, workflows, and how AI agents will execute tasks. This makes me responsible for planning, not typing.
- Why developers must think in layers and systems, not raw code?
  - Developers must think in layers because modern software is built from models, tools, and agents working together. Thinking in systems ensures everything connects properly and scales well. Raw code is only one part — the real strength comes from designing how the entire structure works.



## Part B — Practical Task (Screenshot Required)

Task:

ips for getting started:  
. Ask questions, edit files, or run commands.  
. Be specific for the best results.  
. Create GEMINI.md files to customize your interactions with Gemini.  
. /help for more information.

Write a one-paragraph specification for an email validation function. It must check that the email contains "@", has a valid domain extension like .com or .org, and returns clear error messages for invalid input.

The email validation function will accept a single string argument representing a user's email address. It will verify that the input is a properly formatted email by checking for the presence of a single "@" symbol, which must not be the first character. The function will also ensure the domain name contains at least one "." and that the top-level domain (e.g., .com, .org) consists of at least two letters. If the email is empty, missing an "@" symbol, or has an invalid domain structure, the function will return a specific, user-friendly error message detailing the exact validation failure; otherwise, it will confirm the email's validity.

Gemini CLI update available! 0.9.0 → 0.18.4  
Installed with npm. Attempting to automatically update now...

> Type your message or @path/to/file

## Part C — Multiple Choice Questions

1. What is the main purpose of Spec-Driven Development?

- A. Make coding faster
- B. Clear requirements before coding begins
- C. Remove developers
- D. Avoid documentation

2. What is the biggest mindset shift in AI-Driven Development?

- A. Writing more code manually
- B. Thinking in systems and clear instructions
- C. Memorizing more syntax
- D. Working without any tools

3. Biggest failure of Vibe Coding?

- A. AI stops responding
- B. Architecture becomes hard to extend
- C. Code runs slow

D. Fewer comments written

4. Main advantage of using AI CLI agents (like Gemini CLI)?

A. They replace the developer completely

B. Handle repetitive tasks so dev focuses on design & problem-solving

C. Make coding faster but less reliable

D. Make coding optional

5. What defines an M-Shaped Developer?

A. Knows little about everything

B. Deep in only one field

C. Deep skills in multiple related domains

D. Works without AI tools

## End of Day Reflection

Today I learned that AI-Driven Development nine pillars namely AI CLI & Coding Agents, Markdown as a Programming Language, MCP Standard, AI-First IDEs, Linux Universal Development Environment, Test-Driven Development (TDD), Specification-Driven Development (SDD), Composable Vertical Skills, Universal Cloud Deployment Platforms and how it helps TO M-Shaped developer. Additionally, how Vibe coding is harmful for us and how SDD is useful.