

AIDD 30 Days Challenge Task: 2

Part A - Theory (Short Questions)

- **Nine Pillars Understanding:**

Question 1:

Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better for your growth as a system architect?

Answer:

Using any AI Agent like Gemini CLI for repetitive tasks can save the brain getting tired on boring tasks. Without using AI we are like an alone dev who just have a single task or a goal, he just works on a specific product spending his all time, in simple words using AI with us it can make us think at very high level to build a strong architecture.

But how?

For example when any Agent singlehandedly work on the project setup, code, the folder structure so we dont have to have to waste 20 minutes, the environment etc. Meanwhile we “Devs” can spend our time on thinking about adding more features or scalability and many other things and I tested this my self too it really gives peace of mind to think about. So honestly, this helps us building a strong architecture along with our partner (AI Agent).

Question 2:

Explain how the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer.

Answer:

Nine-Pillars “I can say it is the blue print of building something” are the key concepts on which a developer can work to build something very cool using its AI Agent.

Nine Pillars includes:

- 1) SDD: In Spec-Driven Development we create complete specification our product through which the AI Agent gets the idea to work on and the AI after reading this has a complete work plan.
- 2) MCP Standard: Through this AI understand the specification given to it.
- 3) AI CLI Coding Agent: After understanding the specs the AI then starts coding and creating the docs into reality.
- 4) AI First IDEs: The AI then codes in the AI First IDE. It automatically creates files and manage the folder structure etc.

This approach helps us “Devs” to work on multiple items in the same project or even in the other one too. The M-Shaped means that the Developer have command on multiple domains like frontend, backend or design etc. The M shaped dev is not just limited to any single skill or domain but can work on many with an AI partner.

- **Vibe Coding vs Specification-Driven Development:**

Question 1:

Why does Vibe Coding usually create problems after one week?

Answer:

Vibe coding is basically a code by a beginner or any unaware dev. For example, someone codes through AI and implements it, in the very start the app will work so smoothly but after some time the dev will be unaware of what AI wrote where. He will not know the task of the functions.

In the start it feels like woww but after sometime when the coder faces any bug he is then clueless.

Question 2:

Vibe Coding vs Specification-Driven Development?

Answer:

Likewise when we are using SDD we are just saying AI to write the code for the HeroSection for my website. But, we will be giving the AI a full fledged guide about our final product. Specifications is like the syllabus the institute gives us and they demand us the best result we can get in our final exams.

For example, Sir Zia is the institute and we are the AI. Here Sir Zia designs a spec for us and we then have to read and study it carefully to produce best results.

I think this will clear what mean by vibe coding and SDD.

- **Architecture Thinking:**

Question 1:

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

Answer:

When we are working on AIDD we are not just the coders who write complete code manually. Instead, we assign this job to AI partner and we evolve our role as dev to a system designer if I can deliver what I think. Using our high level architecture the AI will respond in a way no one can think.

Example of role change of a dev in AIDD:

- 1) Our role changes to design a complete blueprint for AI instead of typing code.
- 2) We don't have to do everything by our own, instead we can direct our partner AI Agent to complete the task which we have assigned.
- 3) Bugs are the thing the dev hates the most and using AIDD the AI will solve the bugs instantly when they will occur.

Question 2:

Explain why developers must think in layers and systems instead of raw code?

Answer:

When a developer thinks in layers the product automatically becomes the best.

For example: We are using AIDD “AI Driven Development” and it is the set of some layers like specifications, Logics, AI Coding, Testing and Implementing. So it helps to define clear responsibilities.

So I think that layered thinking reduces risks and it can speed up the development.

Part B - Practical Task

Task:

Using any AI CLI tool, generate a 1-paragraph specification for an email validation function.

My Prompt Screenshot:

```
> Command: Generate a one paragraph specifications for an "email validation function." Requirements: The function must check that "@" is present. It must check that email end with a valid domain extention like (.net, .com, .pk, .org etc.). It must return a clear, human understandable error messages for each type of validation failures. Keep the specification technical but readable for everyone. Output format: Produce a well structured paragraph describing the functional specifications.
```

My Prompt “Text”:

Command:

Generate a one paragraph specifications for an "email validation function."

Requirements:

The function must check that "@" is present.

It must check that email end with a valid domain extention like (.net, .com, .pk, .org etc.).

It must return a clear, human understandable error messages for each type of validation failures.

Keep the specification technical but readable for everyone.

Output format:

Produce a well structured paragraph describing the functional specifications.

Gemini CLI Response Screenshot:

```
■ The email validation function shall accept a string input representing an email address and perform a series of checks to ensure its syntactic correctness. Specifically, it must verify the presence of a single "@" symbol separating the local part from the domain. Furthermore, the function will validate the domain against a predefined list of top-level domains (TLDs) such as .com, .net, .org, .pk, ensuring the email address terminates with a recognized and valid extension. Upon encountering any validation failure, the function shall return a distinct, human-readable error message precisely indicating the nature of the invalidity, such as "Missing '@' symbol," "Invalid domain format," or "Unsupported top-level domain," to facilitate clear user feedback and debugging.
```

Gemini CLI Response Text:

The email validation function shall accept a string input representing an email address and perform a series of checks to ensure its syntactic correctness. Specifically, it must verify the presence of a single "@" symbol separating the local part from the domain. Furthermore, the function will validate the domain against a predefined list of top-level domains (TLDs) such as .com, .net, .org, .pk, ensuring the email address terminates with a recognized and valid extension. Upon encountering any validation failure, the function shall return a distinct, human-readable error message precisely indicating the nature of the invalidity, such as

"Missing '@' symbol," "Invalid domain format," or "Unsupported top-level domain," to facilitate clear user feedback and debugging.

Part C - MCQs

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

My Answer: B

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

My Answer: B

3. Biggest failure of Vibe Coding?

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

My Answer: B

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

My Answer: B

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

My Answer: C

By the end of this Assignment:

After this assignment I'm very much aware of some new terms and some new techniques. I learned about what are NINE PILLARS of AIDD. What are the qualities of a M-Shaped Developer. How vibe coding is not effective without SDD (Spec-Driven Development).