



AI-Driven Development — 30-Day Challenge

Task 2

Name: Khawaja Naqeeb Uddin

Slot: Friday 6-9 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?

Answer:

In my opinion as a system architect , one have to perform many tasks most important from these is to make system flawless. If system architect keep busy in repetitive tasks , his most of the time will waste unnecessarily. For these type of task he must write code for looping or copying the previous one with necessary changes. Therefore it is better to use AI development Agents like Gemini CLI and save precious time , moreover it will give faster and better task performance because repetitive task make human bore and hit their productivity. Now using AI Development Agent more time for designing, optimizing, and building smarter systems instead of repeating the same setup again and again.

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

Answer:

Nine Pillars of AIDD help developer to remove nine different barriers:

- I. AI CLI & Coding Agents are needed to code routine patterns.
- II. Markdown as Programming cover the gap between documentation and implementation.
- III. MCP Standard handles the tool integration task.
- IV. AI-First IDEs give the perfect platform for human and AI to work in flowing way.
- V. Linux Universal Dev Env give the consistency of development across all platforms.

- VI. Test-Driven Development help to automate testing during development which enhance confidence.
- VII. Spec-Driven Development give the structured resumable workflows.
- VIII. Composable Vertical Skills resolves the same problem repeatedly
- IX. Universal Cloud Deployment solve deployment problem with standard tools now deploy anywhere

All these nine pillars not just make us faster but they change us to M-shaped developer , mean one can now go deep in multiple domains in development simultaneously . We don't need to choose between frontend and backend, between infrastructure and application code, between data science and web development. With AI assistance and the right structure, we can develop genuine depth in multiple vertical domains.

2. Vibe Coding vs Specification-Driven Development

- Why does Vibe Coding usually create problems after one week?

Answer:

In Vibe coding one usually just give the prompt by using his or her intuition without planning just see the problem and give the instructions to the AI Agent (LLM). It seems easy and fast the beginning but after sometime when the bug is noticed and program stuck nothing is understandable because previously no planning was done.

- How would Specification-Driven Development prevent those problems?

Answer:

In specification Driven Development prevent all the above said problem , because we first fully understand , plan and give proper instructions to the LLM. Proper context also provided to the agent which eliminate bugs. Proper testing is also done before implementing the actual run of the project. Moreover, if any new bug or stuck also tackle with confidence because implementation done step by step and full record of instruction also saved in the form of Markdown document.

3. Architecture Thinking

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

Answer:

Architecture-first thinking changes the role of a developer in AIDD from simple coder to the controller and orchestrator, now designing systems, orchestrating AI agents, and acting as quality controllers who create and shape the structure instead of typing the implementation.

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

Answer:

Now in AIDD developers must think in layers because sophisticated software requires boundaries , well-defined structure which is also scalable and must be well-organized which the raw code can never provide. Now developer's role a controller of AI, an architect, an orchestrator who have to manage work in separate layers, fully test them and make changes in different stages without break of the flow or mess-up the entire project and all these can only be possible when one is working in layers.

Part B — Practical Task (Screenshot Required)

Task:

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

Requirements:

Must contain "@" Must contain a valid domain (e.g., .com, .org) Should return clear error messages

```
> GEMINI

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

Gemini CLI update available! 0.14.0 → 0.16.0
Installed with npm. Attempting to automatically update now...

Update successful! The new version will be used on your next run.

> Write a paragraph that specifies an email validation function. Requirements: purpose, input type, output, validation(@,domain labels etc.) edge cases and one line example of valid and invalid address. Out put only in paragraph without headings.

This email validation function serves to verify the correctness of email addresses according to standard formatting rules. It accepts a single string as input, representing the email address to be validated. The function returns a boolean value: true if the email is valid, and false otherwise. Validation criteria include the presence of a single '@' symbol separating the local part from the domain, at least one domain label (e.g., '.com', '.org'), and no leading or trailing spaces. Edge cases such as empty strings, multiple '@' symbols, or invalid characters within the local or domain parts are handled, ensuring robust checking. For example, "test@example.com" is a valid address, while "invalid-email" is not.
```

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

Answer: option : 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

Answer: option : 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

Answer: option : 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

Answer: option : 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

Answer: option : C

