

AI-Driven Development — 30-Day Challenge

Task 2

Rida Azam

Part A — Theory (Short Questions)

Question 1: Nine Pillars Understanding

1. Nine Pillars Understanding

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

AI agents handle boring and repetitive work like setting up projects or writing standard code. This lets you focus on **designing the system**, planning architecture, and making decisions instead of just coding. Tools like Markdown give clear instructions to AI. MCP connects AI to other tools. AI-first IDEs like VS Code put AI at the center of your workflow. Testing (TDD) ensures AI-generated code works correctly. Cloud tools like Docker and Kubernetes create deployment-ready systems. Using these pillars together **saves time, reduces errors, and helps you learn faster** as a system architect.

b. How do the Nine Pillars help a developer grow into an M-Shaped Developer?

The Nine Pillars let you build **deep skills in many areas at once**. AI agents and AI-first IDEs remove routine tasks. Standard tools like Linux, MCP, and cloud deployment make work consistent. Testing gives confidence that code works. Composable skills let AI guide you in any area. This helps developers combine knowledge from multiple domains and become **M-shaped developers** — skilled in several related fields.

Question 2: Vibe Coding vs Specification-Driven Development

Vibe Coding

Starts coding immediately, no plan.

Specification-Driven Development (SDD)

Makes a clear plan (spec) before coding.

Vibe Coding

Confusing requirements for team members.

Skips tests or relies on manual checking.

Code grows messy and architecture drifts.

Adding new features can break old code.

Problems appear after a few days.

Teamwork can be messy.

Hard to maintain or teach new developers.

Specification-Driven Development (SDD)

Everyone follows the same plan.

Tests are written first (TDD).

Code follows the plan and stays organized.

New features add safely without breaking existing code.

Predictable, maintainable, and easy to grow.

Clear specs improve collaboration.

Documentation + tests make it easy.

Summary: Vibe coding is fast but messy. SDD is slower at first but keeps the code **reliable, organized, and easy to maintain**.

Question 3: Architecture Thinking

3. Architecture Thinking

a. How does architecture-first thinking change a developer's role?

Developers become **system designers**. They plan the structure, data flow, and integration points first. Then AI agents implement the code based on that plan. Developers guide AI to make **reliable, maintainable, and scalable systems**.

b. Why must developers think in layers and systems instead of raw code?

Layers separate the system into parts like **data, backend, and frontend**, each with a clear job. This makes AI-generated code fit naturally, helps add features easily, and improves teamwork. Thinking in systems keeps the project **organized, flexible, and scalable**.

Part B — Practical Task (Screenshot Required)

Task: Generate a 1-paragraph specification for an email validation function.

Requirements:

- Must contain “@”
- Must have a valid domain (e.g., .com, .org)
- Should return clear error messages

Submission:

1. CLI prompt text or screenshot
2. 1-paragraph specification generated by AI CLI

```

File Edit Selection View Go Run ... task2
EXPLORER PROBLEMS TERMINAL
TASK2
i Update successful! The new version will be used on your next run.
> 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.

> Write one single paragraph that describes a step-by-step specification for an email validation function. The paragraph must include the '
symbol and a valid domain example like user@example.com or admin@mysite.org (.com or .org). Explain step-by-step validation inside the
paragraph (Step 1: Step 2: etc.). Also mention that the function should return clear error messages such as 'Missing @', 'Invalid domain
must end with .com or .org', or 'Local part empty'. Output only one paragraph and nothing else."

i Gemini CLI update available! 0.13.0 -> 0.16.0
Installed with npm. Attempting to automatically update now...
* The email validation function should meticulously check the input string step-by-step to ensure it conforms to standard email formats,
returning specific error messages for clarity: Step 1: The function must first verify the presence of a single '@' symbol; if absent, it
should return 'Missing @'. Step 2: Upon finding the '@' symbol, the string is split into a local part and a domain part. Step 3: The
local part is then validated to ensure it is not empty, returning 'Local part empty' if it is. Step 4: Finally, the domain part is
checked to confirm it ends with either '.com' or '.org', as seen in examples like user@example.com or admin@mysite.org; otherwise, it
should return 'Invalid domain: must end with .com or .org'.

i Update successful! The new version will be used on your next run.
Using: 1 MCP server

```

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

Reflection

This task shows how **AI is changing the developer role**. Concepts like the Nine Pillars, SDD, and AI Development Agents are more than tools—they change how software is planned, built, and maintained. Architecture-first thinking and collaborating with AI agents help developers grow from simple coders into **system-level thinkers**, eventually becoming **M-shaped professionals** who combine deep skills across domains with intelligent tools.