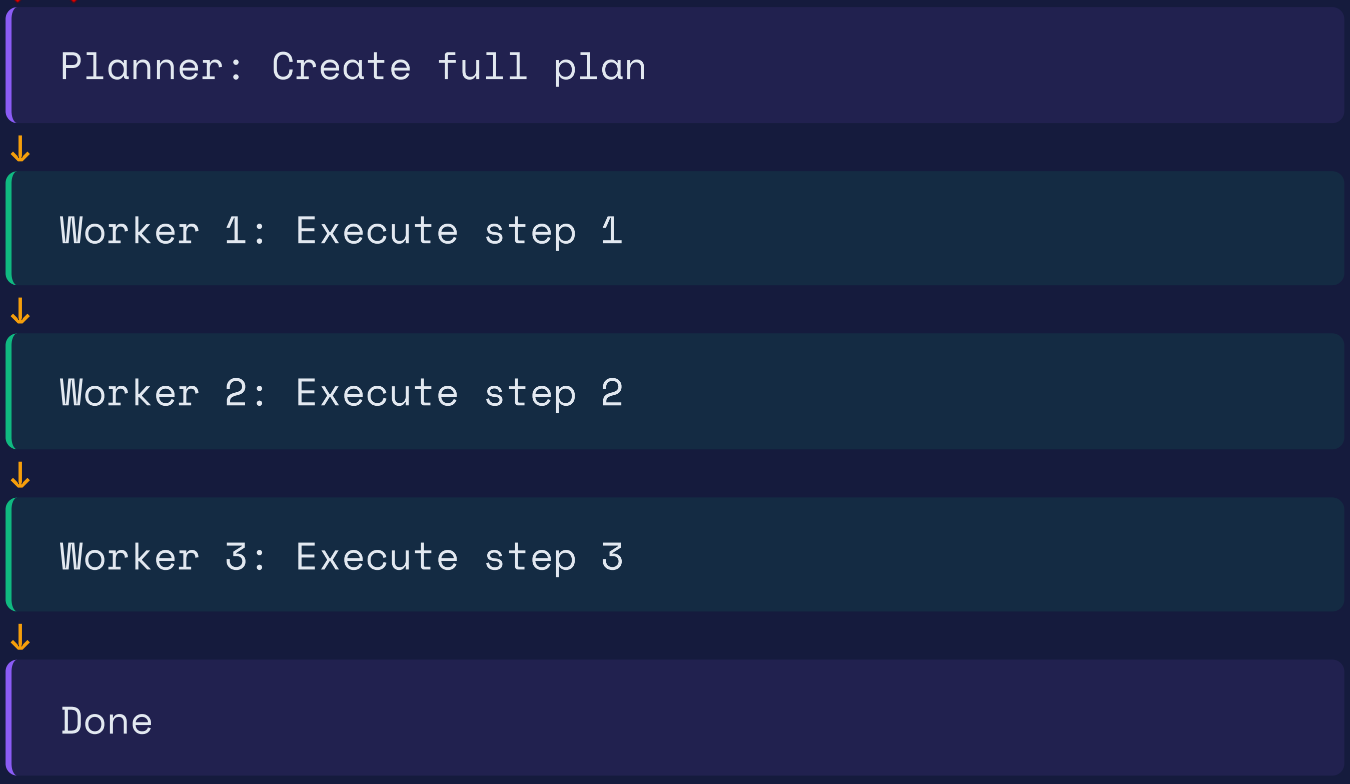
The Planner-Worker Pattern Everyone Gets Wrong

Common Mistakes & How to Fix Them

> Most implementations fail because they treat it as sequential, not iterative.

Wrong vs Right Approach

❌ What Most Do



Planner: Create full plan

↓

Worker 1: Execute step 1

↓

Worker 2: Execute step 2

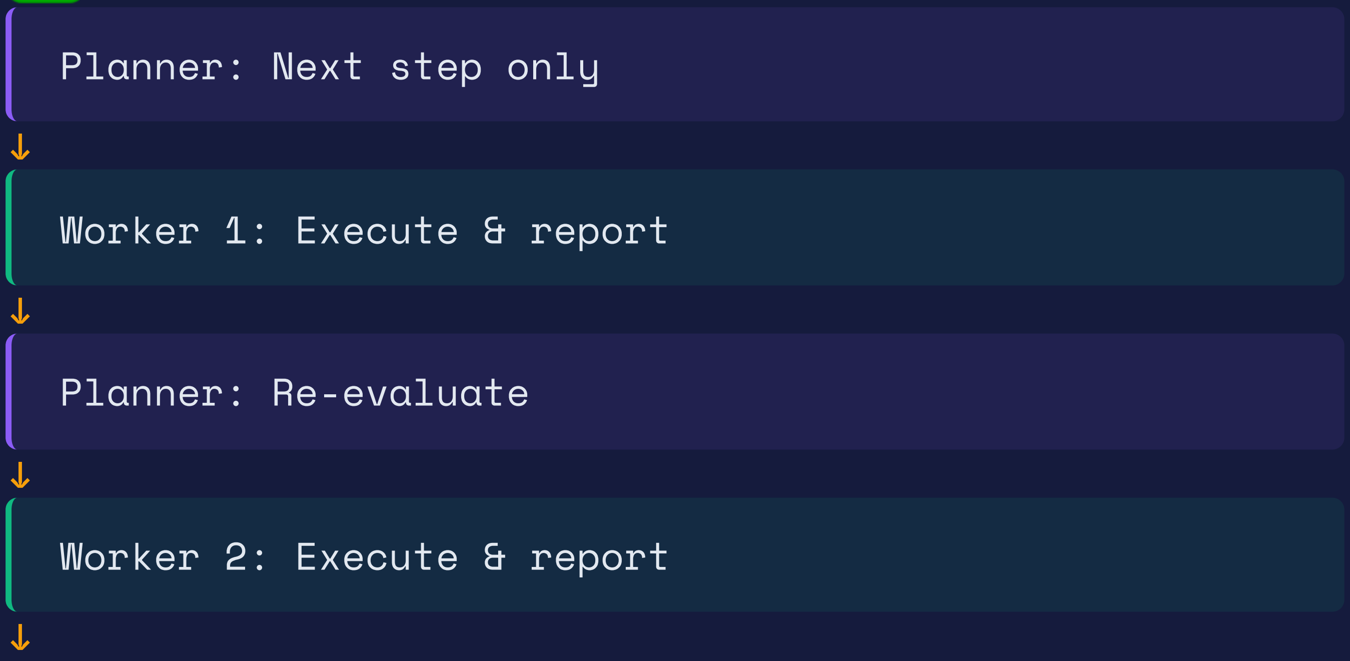
↓

Worker 3: Execute step 3

↓

Done

✅ What You Should Do



Planner: Next step only

↓

Worker 1: Execute & report

↓

Planner: Re-evaluate

↓

Worker 2: Execute & report

↓

```
Planner: Adjust
```

Common Mistakes

Mistake #1

Planning Everything Upfront

Creating a complete plan before execution. Reality changes, plan becomes obsolete.

Mistake #2

No Feedback Loop

Workers execute blindly without reporting back. Planner can't adjust.

Mistake #3

Treating Failures as Fatal

One worker fails → entire system fails. No retry, no alternative paths.

State Management

What to Track

- Current goal
- Steps completed
- Results so far
- Failed attempts

How to Track

- Shared state object
- Update after each step
- Pass to next worker
- Planner reads state

What Workers Return

- Success/failure status
- Result data
- Error details (if failed)
- Recommendations

What Planner Does

Read worker results

Update understanding

Decide next action

Adjust if needed

When Workers Fail

Strategy 1: Retry with Backoff

Worker fails → Wait → Retry with same input

Best for: Transient errors (network, rate limits)

Strategy 2: Alternative Approach

Worker fails → Planner tries different worker or method

Best for: Tool-specific failures (API down, access denied)

Strategy 3: Simplify Task

Worker fails → Planner breaks task into smaller pieces

Best for: Task too complex (context overflow, ambiguity)

Strategy 4: Human Escalation

Multiple failures → Ask human for help or clarification

Best for: Unrecoverable errors, ambiguous requirements

Weekend Project

Build It This Weekend

Goal: Planner orchestrates 3 workers to complete a research task

Planner

Decomposes task, decides which worker to call, re-evaluates after each step

Worker 1: Data Retrieval

Makes API calls, returns raw data or error

Worker 2: Data Processing

Transforms data, performs computation, returns results

Worker 3: Summary

Takes processed data, generates natural language summary

Key Principles

1. Plan one step at a time, not the entire journey

2. Workers report back, planner re-evaluates

3. Failures are opportunities to adjust, not endpoints

4. State flows through the system, not just data

Iterative, not sequential.

That's the difference between brittle and robust.

💡

Implementation Examples

community.nachiketh.in

🎓

Production Patterns

bootcamp.nachiketh.in

Speaker notes