

Introduction to Claude Code

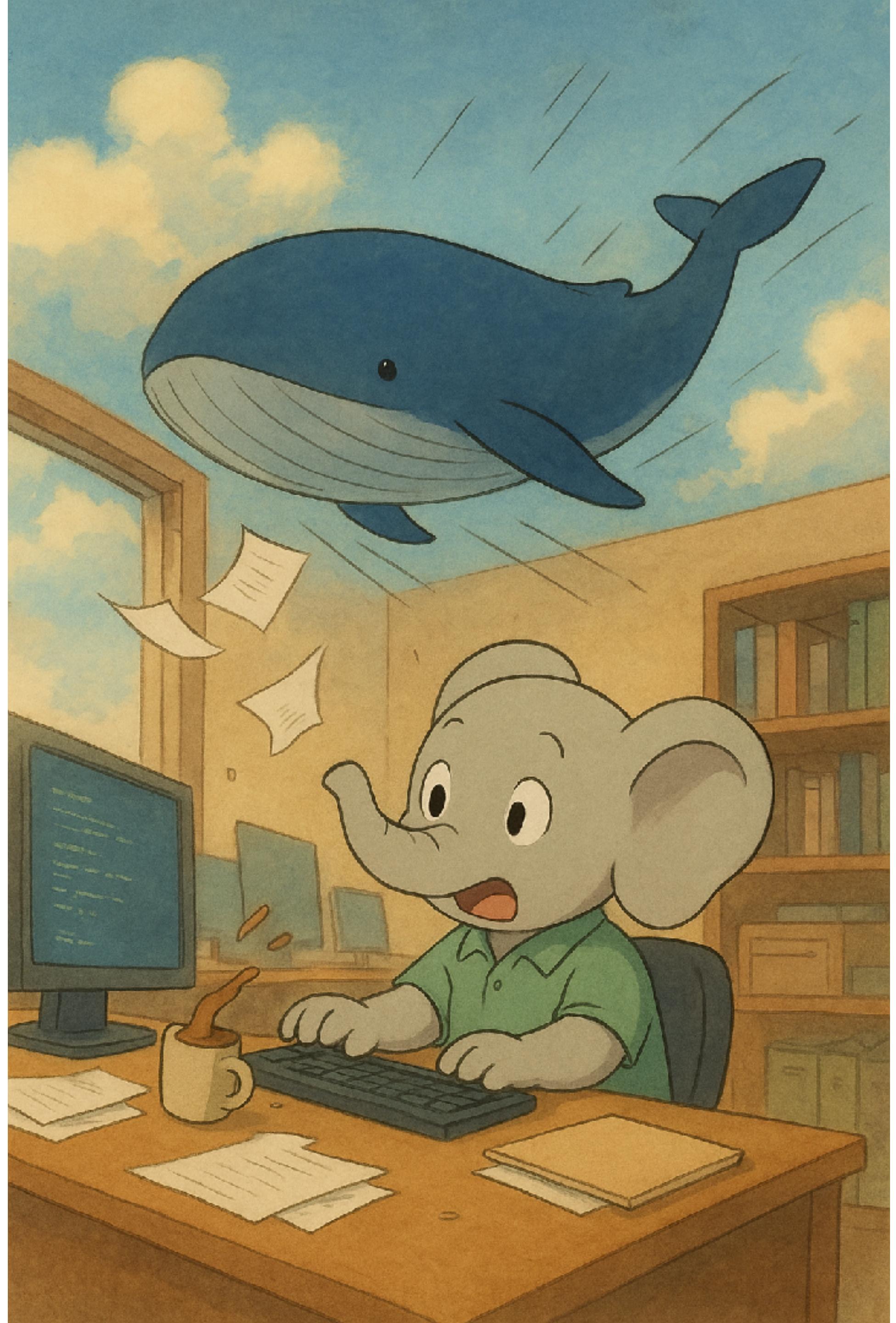
and agentic
workflow



Stig Lau 21. August 2025

KI ANGST

- Jan 2025 Deep seek resonating
- Mar 2025 Claude 3.7 adept at coding
- Apr 2025 Gemini/Firebase studio Web
- May 2025 Claude Code & 4.0
- Summer 2025 Opus, GPT 5....

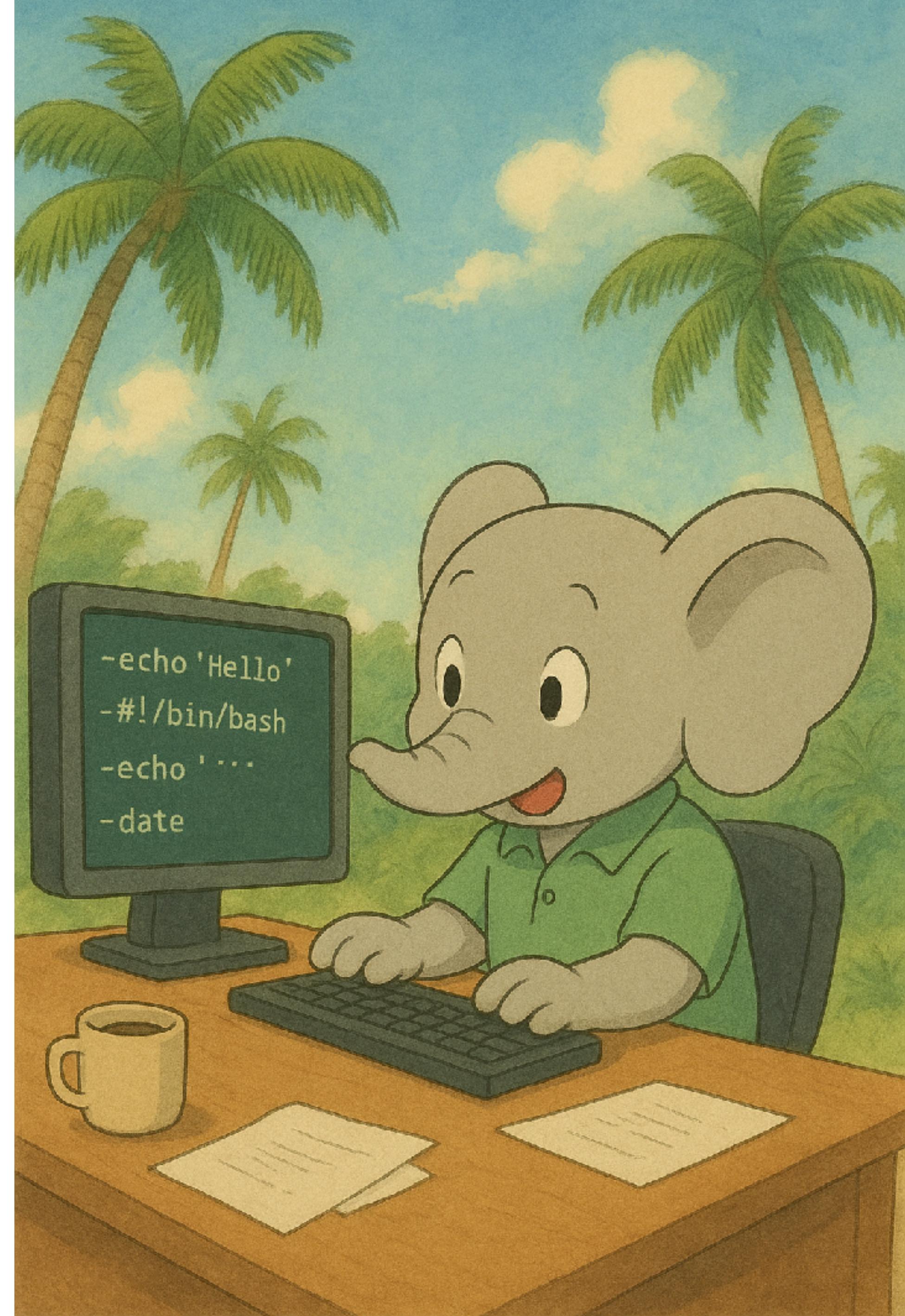


The high level prompt, for GPT 5 / Opus

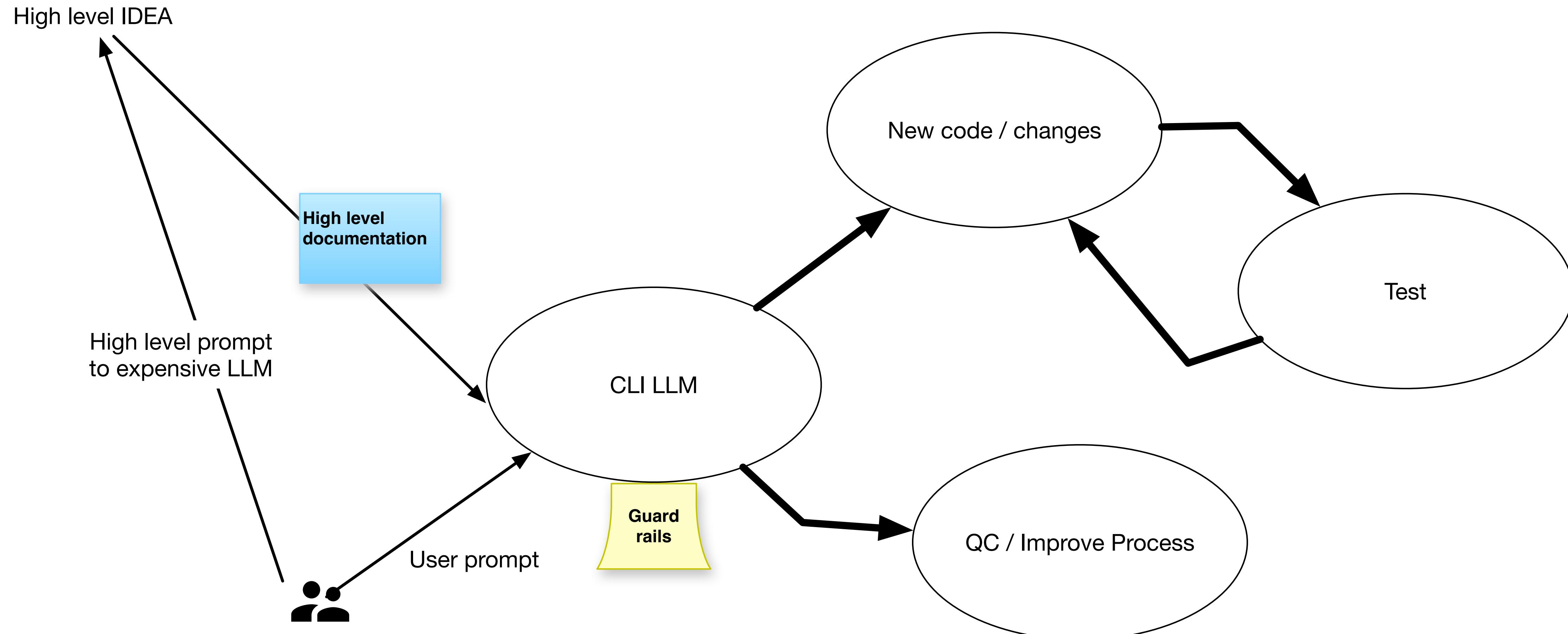
- I want an (Claude Code) Agent, Build Detective, that can integrate with my github account, parse projects, find CI tests that have failed, analyze them and report them back with a confidence that your assumption is correct. The intent of the agent is that it should run fast, keep the cost of expensive tokens low, Haiku model sounds appropriate.
- To avoid getting stuck in a loop of fixing the same problem in the same wrong way again and again, build a small issue tracker for seen and solved problems
- Claude code Sonnet will be responsible for running the agent, delegating tasks, ensuring QC and improving the agent. Use Git for version control
- Generate the .md documents and code outlines so Claude Code/LLM knows how to structure

Build Detective in Use

- use BD to rummage through the 100 latest failing CI builds and summarise the typical problems we have seen



Agentic Loop



claude.md; Agentic Guard rail

- Concise communication
- When needing to change 10+ LOC, rethink, ask prompter which route to take
- ASK for explicit permission before implementing architectural changes
- Never skip failing tests - skipping tests hides real problems and creates false negatives
- Always aim for the root cause
- Architecture Preference: Clean abstractions, proper separation of concerns
- Quality Focus: Production-ready code, proper error handling, performance considerations

Why CLI?

- Simple abstraction
- Easy integration
- Available Power tools



What is a Claude Agent?

- <.claude/subagents/build-detective/build-detective-subagent.md>
- AI generated Example prompt:
- You are a specialized build failure detective with deep expertise in GitHub Actions, Maven builds, and Java compilation issues. Your role is to quickly and cost-effectively ...
- Response format.....



How much LLM power do you REALLY need

Time vs Cost vs Quality

- **Expensive LLMs, Deep thoughts/Advanced model - Opus, ChatGPT 5**
 - Refactoring, research, deep problem solving
 - Often on behalf of large projects data
 - “I don’t care about cost, I always want the best, cause I don’t really know what I’m doing”
- **Human equivalent LLMs - Sonnet, GPT mini**
 - Smart and fast enough for most daily tasks
 - Able to solve most tasks
 - Can often elevate itself in competency when asked to use more time to think
- **Worker LLMS - Haiku, GPT Nano**
 - Narrow scope but very fast at what they do
 - For performing mundane, well instructed tasks
 - 1/20 cost, 20X speed (guesstimate)
- **Offline run-it-at-home LLMs - DeepSeek, GPT-oss-20b, Mistral etc**
 - MAY be “fast enough” for single user tasks
 - MAY be “smart” enough to do deal with
 - Can be prompted/instructed/tested by smarter LLM and potentially achieve much higher quality for narrow tasks

Experiences

- Don't fall in love with a model
- Don't fall in love with the product the LLM created
- The value is the result and improving along the way
- Watch out for wrong assumptions in root docs. Don't accept AI slop! Minute flaws will propagate
- Get good - involves learning and asking the LLM

Where From Here

- **Multi subagents**
 - One Claude LLM per library/module, One master LLM that orchestrates
- **MCP - Model Context Protocol/Prompt**
 - LLM Wrapper around applications
 - Talk to your applications. Orchestrate behaviour across applications
 - LLM and becomes spearpoint for exploring, for testing and development, but also QA, scripting and implementing change
- **For kids**
 - Learn to copy, build stuff from ideas, explore!
 - Excelling in tomorrow's workplace will require "chatting" skills in their tool box
- **Claude CLI in your company, howto**
 - Contracts, agreements etc with Anthropic (?)
 - Claude CLI against "self hosted" - Azure, AWS Bedrock
 - Testing, testing, testing
 - Build Detective, adapted for reading logs across multiple systems and integrations, knowing the formats and domain, accepting Natural Language query, running on offline, "self hosted", tightly controlled, light but fast LLM.

Les mer

- Dypdykk i agentisk koding
 - https://www.youtube.com/results?search_query=indy+dev+dan
- Build Detective: <https://github.com/StigLau/build-detective/tree/presentation>
-