

# Design your own Agentic workflow



by **Rafał Camlet**  
CTO at **2N IT**

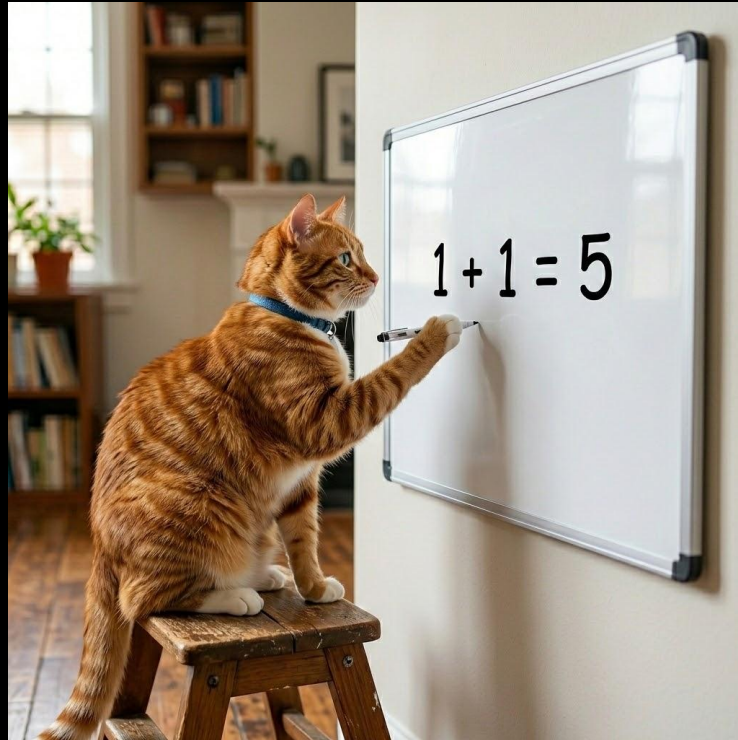
**“No tree can grow to heaven unless its roots  
reach down to hell.”**

- Carl Jung

# Context Management



# Non-determinism



# Quality degradation - garbage in, bigger garbage out



**“The entropy of an isolated system can only increase.”**

- Second Law of Thermodynamics

**The flat just gets  
messy on its own**



**“AI is a quick but sloppy co-worker”**

- Rafał Camlet

# Context Building

# Context Building

- Build-in research capabilities
- Project documentation `.ai/docs`
- AGENTS.md with documentation router
- External sources (like linear, pencil, ect.)
  
- Brainstorming

For exploratory questions ("what could we do about X?", "how should we approach this?", "what do you think?"), respond in 2-3 sentences with a recommendation and the main tradeoff.

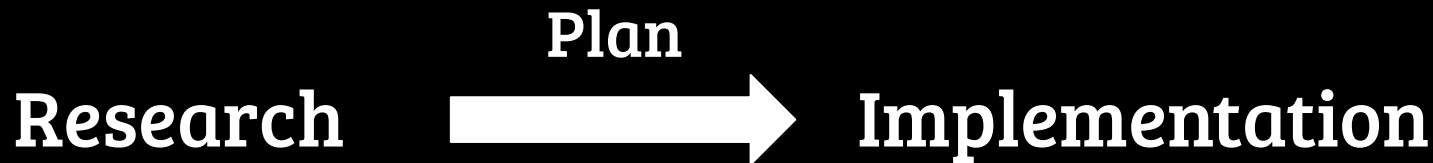


- Opus 3.7 system prompt

# Context Pollution

# Context Separation

- Plan Mode
- Compact
- Summaries
- Clear



# Subagents

- Claude Code forked context
- RAG systems
- MCP
- Other models

```
1 _____  
2 name: explorer  
3 description: Research a topic  
4 model: haiku  
5 context: fork  
6 _____
```



# Harness engineering

# Guardrails

Checking non-deterministic agents with deterministic tools

- permissions
- hooks:
  - linters - check code quality
  - tests - check correctness and requirements coverage
  - custom scripts - check everything else

# TDD - Test Driven Development

...or just Test First

AI loves tailoring tests to the existing implementation, which lowers their quality

No code, nothing to base on



**Project entropy  
grows with time**

**Be a gardener!**



# Cleanups

- AI self validation with clean context
- sync documentation
- retrospection (memory)
- refinement (dream)

**What do we use in  $2N$ ?**



# agentskills.io

```
my-skill/  
├── SKILL.md           # Required: metadata + instructions  
├── scripts/          # Optional: executable code  
├── references/       # Optional: documentation  
├── assets/           # Optional: templates, resources  
└── ...               # Any additional files or directories
```

## Prepare

- Fetch Linear ticket
- Fetch pencil design
- Read app documentation
- Pre coding tasks

## Brainstorm

- Propose alternative solutions
- List pros and cons

# Create Plan

- Detailed implementation plan
- Human approved
- Clear context

clear context

# Implement

- Clean context
- Test Driven Development



## Retro

Save lessons from

- coding loops
- human Interventions
- bugs

clear context

## Review

- Review code changes
- Check requirements coverage

# Document

- Write or update documentation for a completed task



# Sync Guides

- Check discrepancies between the code and the documentation, agents instructions

**Thanks!**