

4-7 MAY 2025

ScrumAlliance®

Global Scrum Gathering®

M U N I C H

Common Pitfalls in AI-Augmented
Pair Programming



Learning outcomes

The attendees will be able to:

- Detect harmful approaches while developing with an LLM
- Coach developers on the different ways to use an LLM to code
- Map their current context to an AI-augmented strategy

4-7 MAY 2025 ScrumAlliance®
Global Scrum Gathering®
MUNICH



INTRO

*My first international training
session, in Portugal*

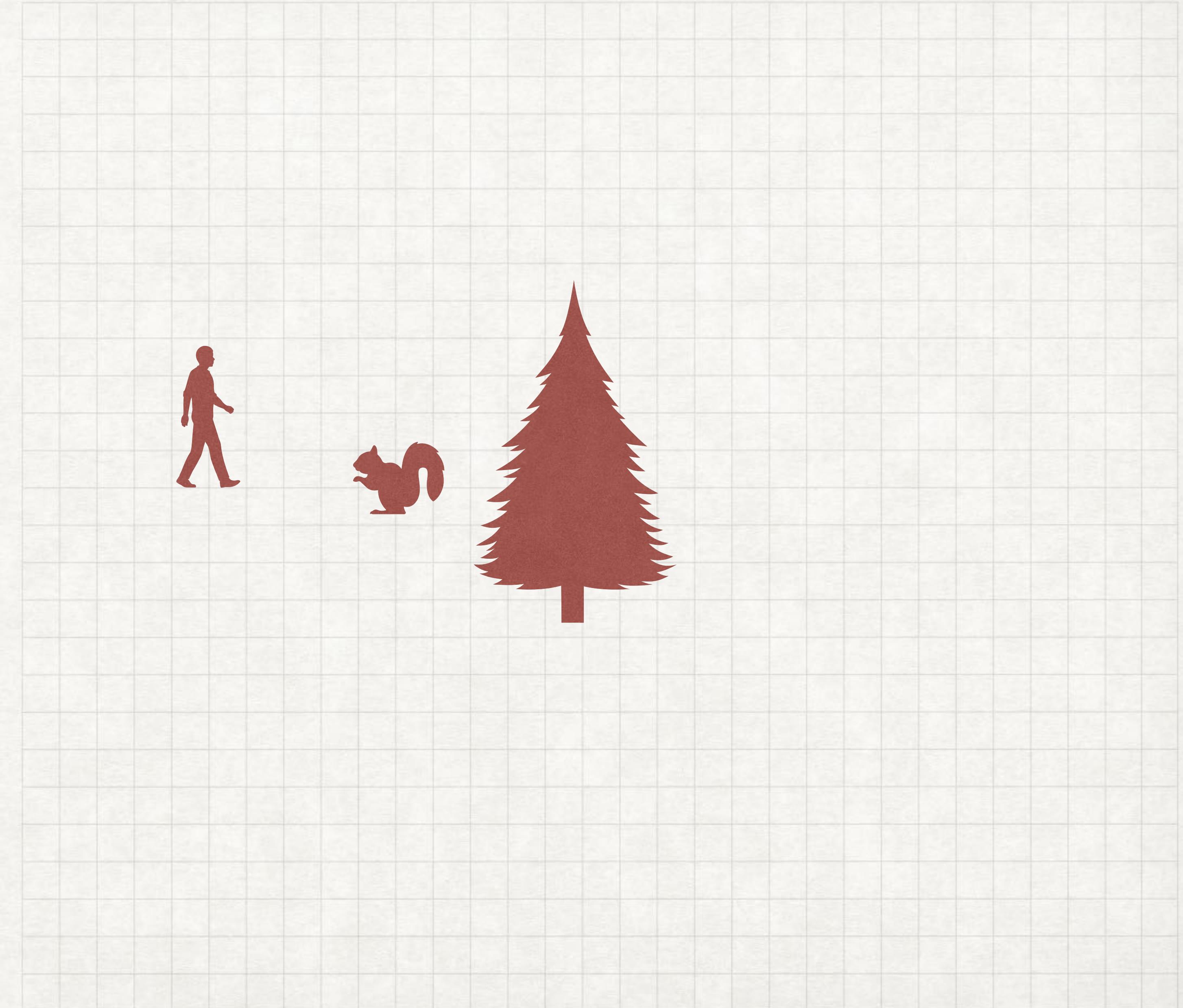




THE WRITTEN FIXATION

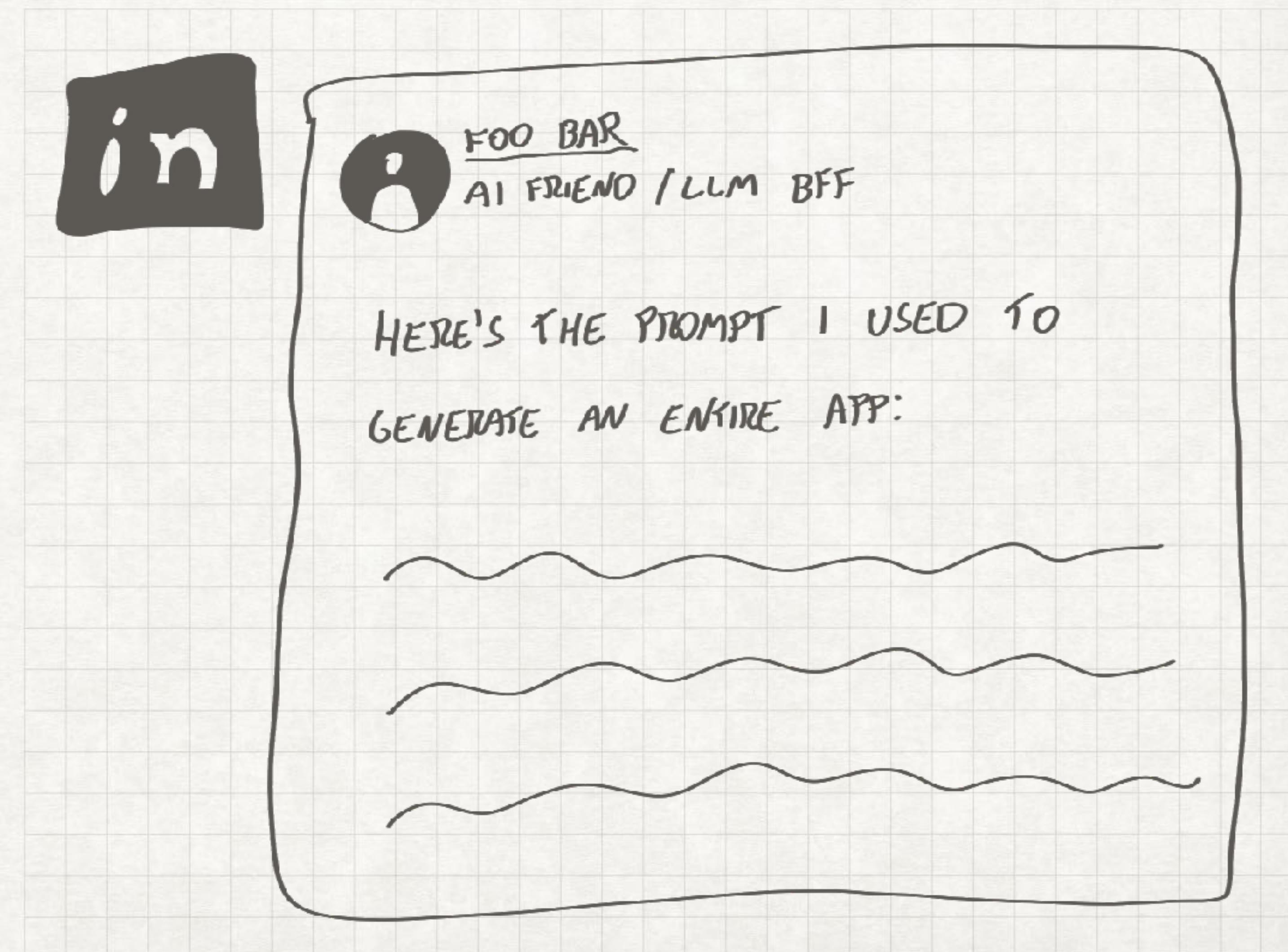
The first pitfall

WILLIAM JAMES AND
THE SQUIRREL
DID THE MAN
GO AROUND
THE SQUIRREL?



HOW DO WE COMMUNICATE OUR NEEDS? AND HOW DID OUR COMMUNICATION EVOLVE OVER TIME?

- An User Story is a promise for a conversation - *Alistair Cockburn*
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation - *Agile Manifesto*

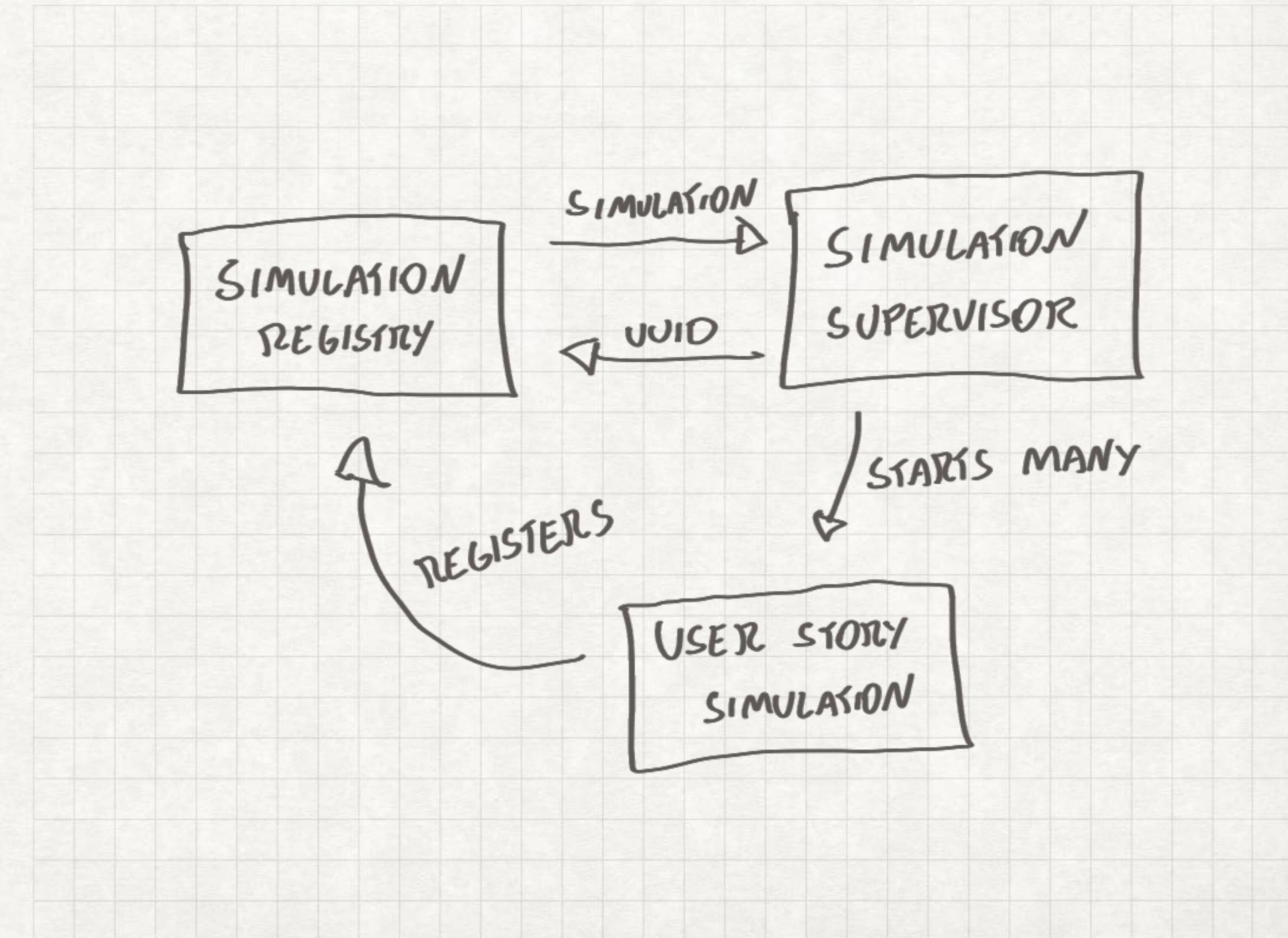


WHY WRITE WHEN YOU CAN TALK OR DRAW?

PLEASE, DON'T APPLY THIS LOGIC WHEN SENDING ME WHATSAPP MESSAGES

We've spent decades perfecting whiteboard communication with humans, only to abandon it completely when talking to AI.

It's the ultimate technological amnesia.



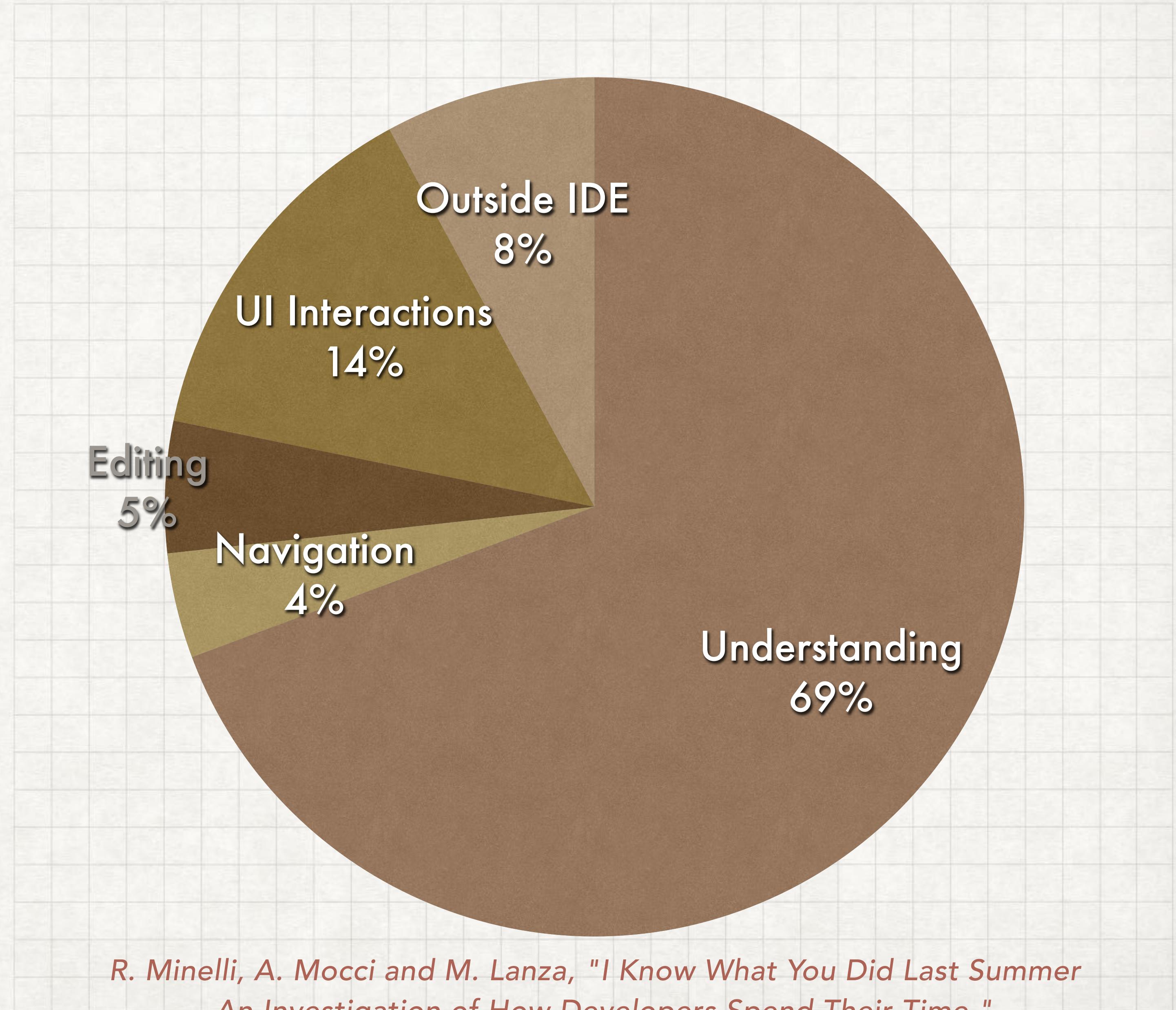
THE CODING TUNNEL VISION

writing code is cool, but not everything

WHY DEVELOPERS STILL LOVE VIM?

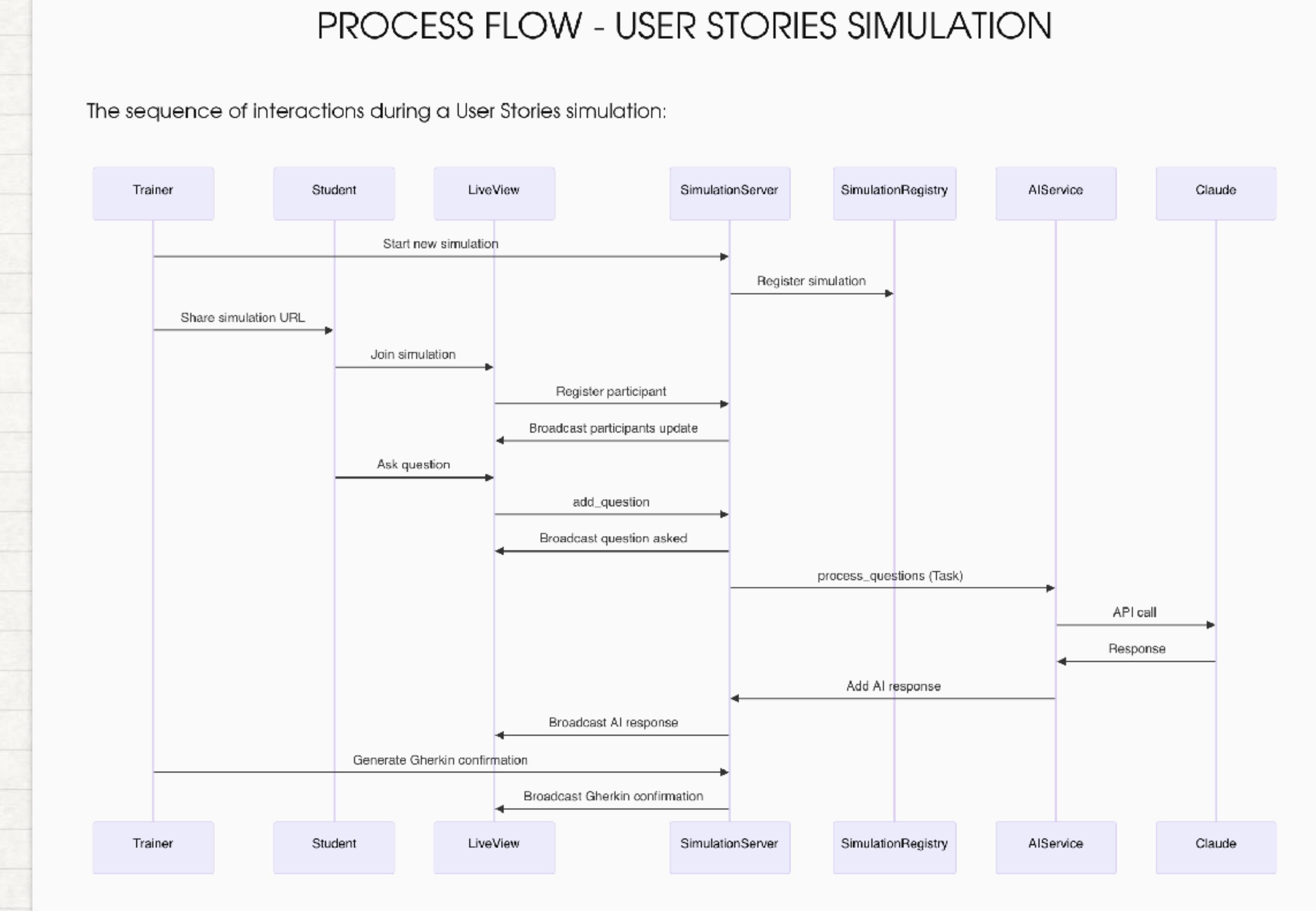
WRITING VS NAVIGATING

HOW DEVELOPERS SPEND THEIR TIME

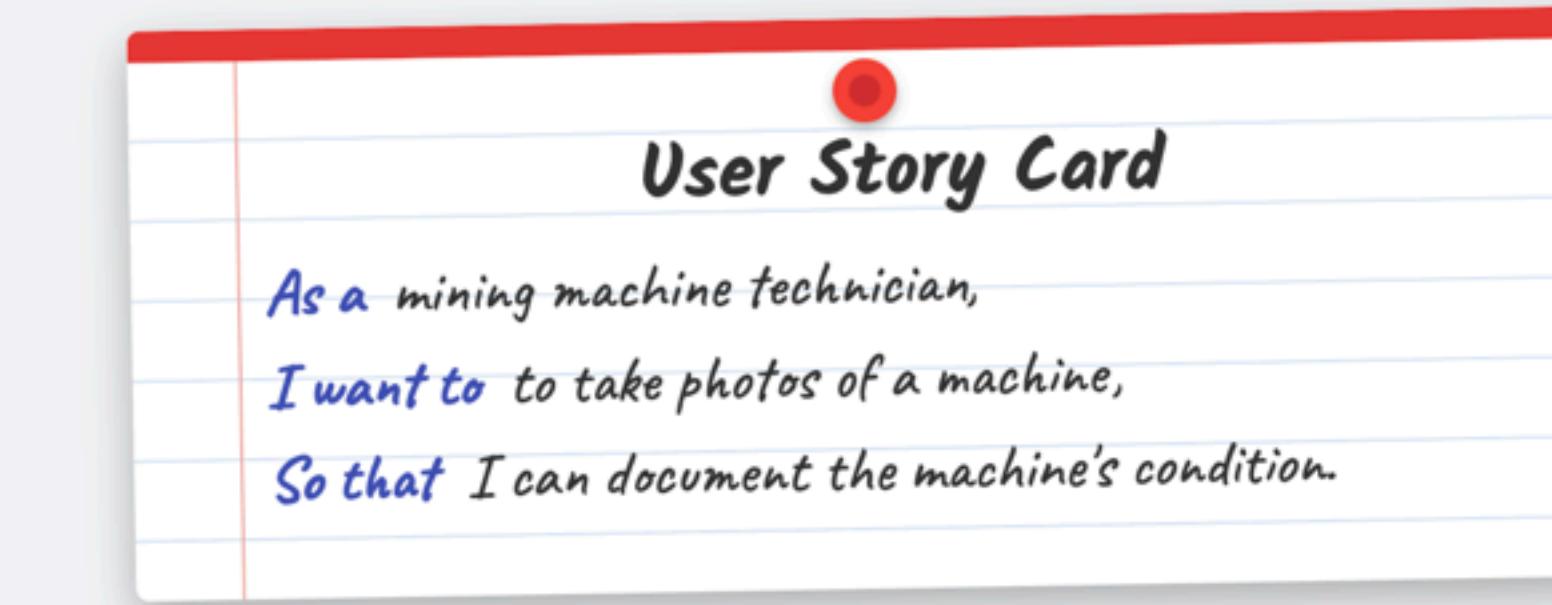


I asked the LLM to look at specific chunks of my code and generate a short document to help me onboard a new developer to the team.

Hint: Ask for a markdown document with mermaid diagrams.



A developer does more than coding...



What is an User Story?

User stories are told. They are not mere template, but a way to have developers and customers understanding each other. The whole process comprises 3 C's:

 **Card.** User stories are written on cards. The card does not contain all the information that makes up the requirement. Instead, the card has just enough text to identify the requirement, and to remind everyone what the story is.

 **Conversation.** The requirement itself is communicated from customer to programmers through conversation: an exchange of thoughts, opinions, and feelings. This conversation takes place over time, particularly when the story is estimated (usually during release planning), and again at the iteration planning meeting when the story is scheduled for implementation.

Conversation

26 messages

Conversation

26 messages

Conversation

key

is there a process for inferring the machine condition using photos?

Samantha Fey

That's a great question. Using photos to infer the condition of the equipment is definitely a key part of what I'm hoping to achieve here.

The way I see it, if we can build up a comprehensive visual record of each haul truck over time, it should give us the ability to start spotting trends and detecting potential issues before they become major problems.

Some of the things I'd want our technicians to focus on capturing in the photos include:

- Wear and tear on tires, suspension components, and other high-stress parts
- Fluid levels and any signs of leaks
- Damage or cracks in the body, frame, or attachments
- Engine and component cleanliness, which

⚠ Chat is disabled

This simulation is currently paused by the trainer. Please wait until it's resumed.

... so why people insist on
replacing devs with what fancy
code generators?

User Story Confirmation

Acceptance Criteria

feature.gherkin

Feature: Documenting Mining Equipment Condition

As a mining machine technician,
I want to take photos of a machine,
So that I can document the machine's condition.

Background:

Given a mining haul truck has been in service for 3-4 years

And the haul truck has experienced some maintenance issues in the past

Scenario:

Given the technician is inspecting a mining haul truck

When the technician takes photos of the haul truck

[Back to Conversation](#)

THE MONOLITHIC APPROACH

Don't believe in silver bullets

WHAT IS VIBE CODING?

CONVERSATIONAL
REACTIVE
ASSISTED
PROGRAMMING

A REAL EXAMPLE

I'm getting a duplicated message, could you give me a hand?

I'll fix that!

It didn't work, let me try another strategy.

I think I got it now, let me try again.

It seems to be a problem with another file, let me fix it first.

Hum... we're still exhibiting the duplicated message, I'll try another approach!

I did it!

+ 143 LINES OF CODE

- 41 LINES OF CODE

THE CHAT WAS UPDATED!

THE CHAT WAS UPDATED!

THE SOLUTION

Whenever we receive a message, we'll wait half a second to see if we receive another message with the same content.

If we do, we ignore the second one.

Having control over your context can take some of the fun away, but it makes you more conscious about what you're doing

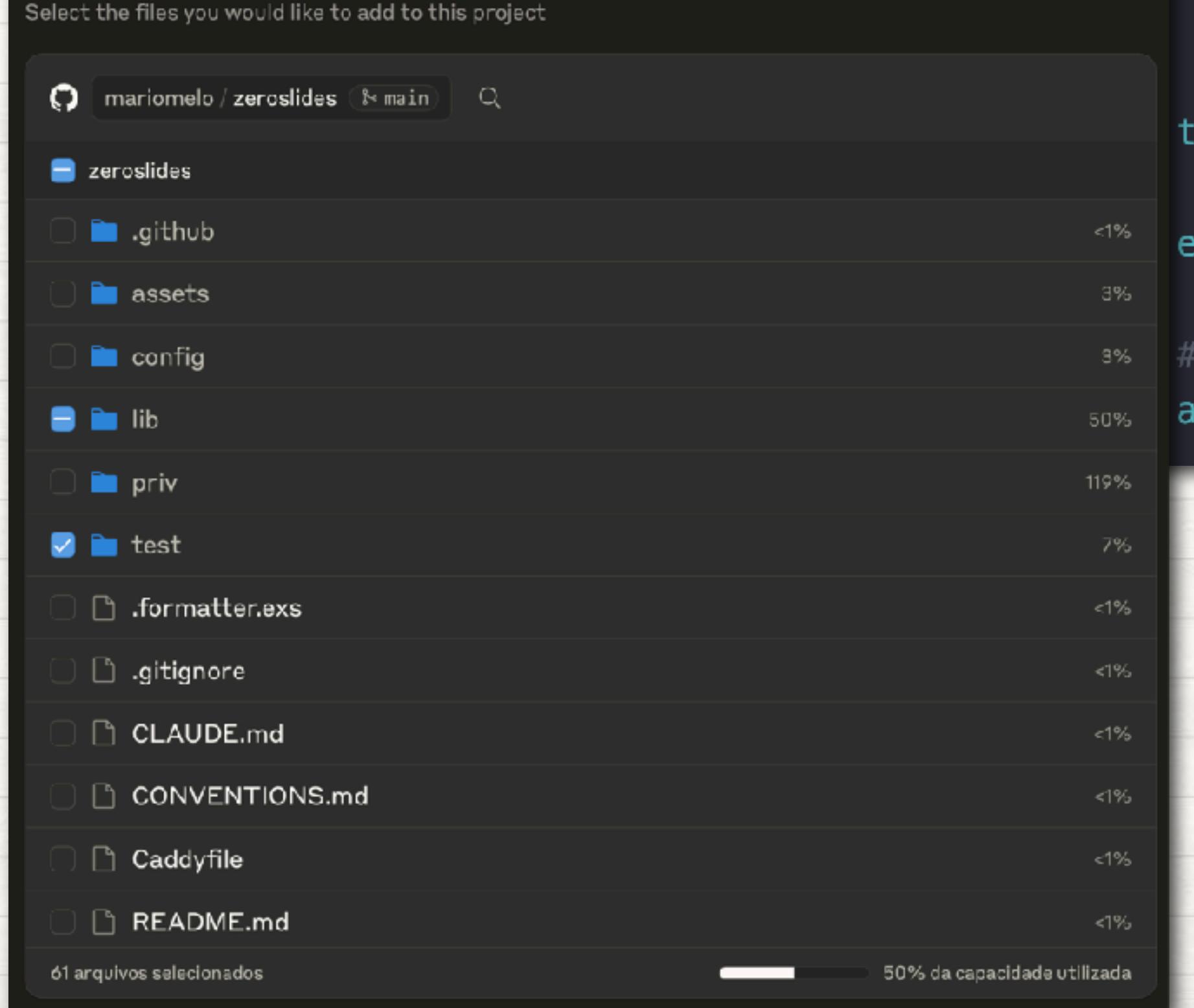
```
melomario@MacBook-Pro ~/p/zeroslides (main)> aider

Aider v0.82.2
Main model: deepseek/deepseek-reasoner with diff edit format, prompt cache, infinite output
Weak model: deepseek/deepseek-chat
Git repo: .git with 142 files
Repo-map: using 4096 tokens, auto refresh
Added README.md to the chat (read-only).
Added CONVENTIONS.md to the chat (read-only).

 Readonly: CONVENTIONS.md README.md
> 
```

Add content from GitHub

Select the files you would like to add to this project



mariomelo / zeroslides (main)

- zeroslides**
- .github**
- assets**
- config**
- lib**
- priv**
- test**
- .formatter.exs**
- .gitignore**
- CLAUDE.md**
- CONVENTIONS.md**
- Caddyfile**
- README.md**

61 arquivos selecionados 50% da capacidade utilizada

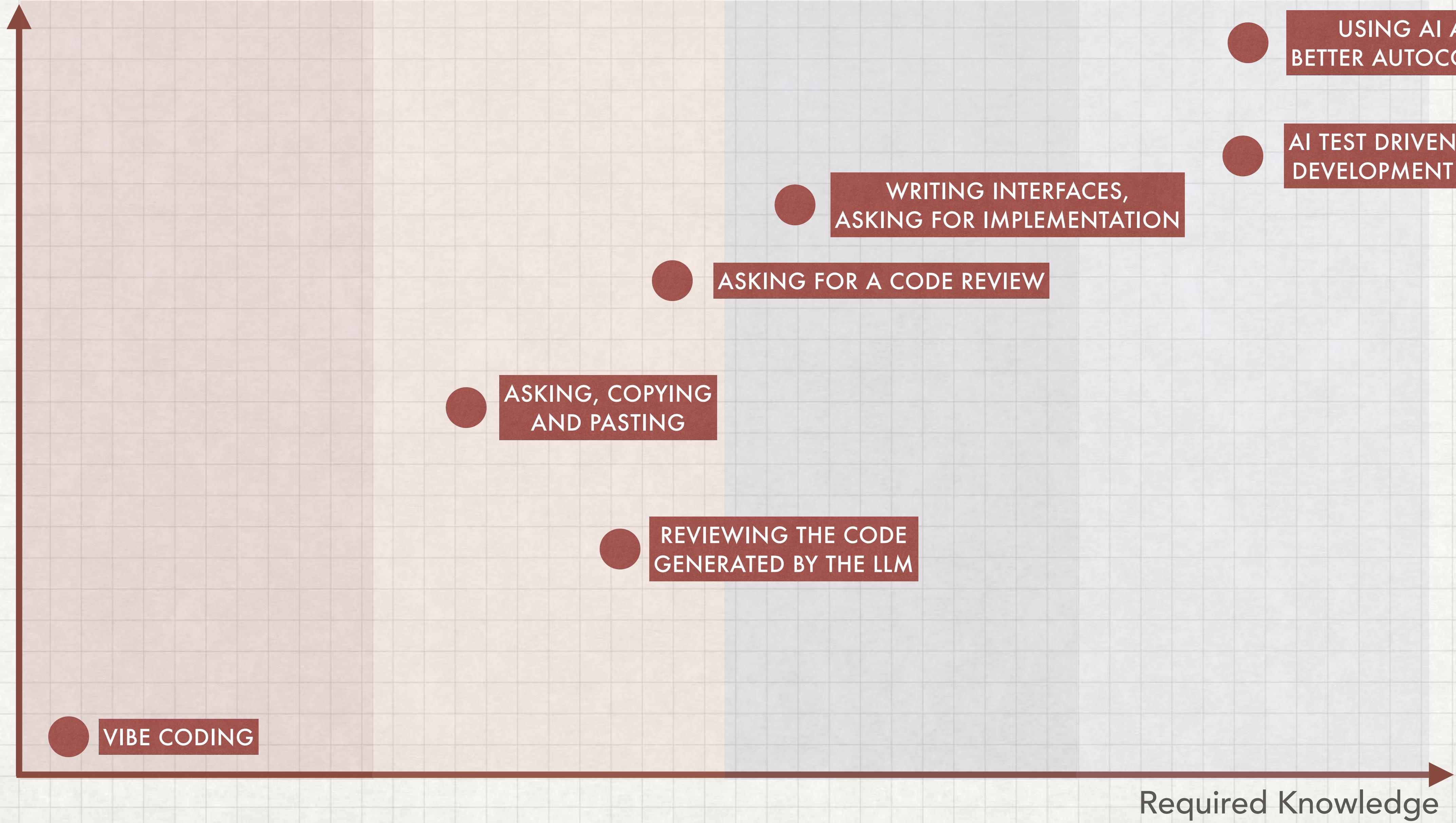
```
21 read:
20   - CONVENTIONS.md
19   - README.md
18
17 model: deepseek/deepseek-reasoner
16 editor-model: deepseek/deepseek-coder
15 weak-model: deepseek/deepseek-chat
14
13 architect: false
12
11 auto-lint: true
  × lint-cmd: mix compile

# auto-test: true
test-cmd: mix test

editor: nvim

# Git Configuration
auto-commits: false
```

Effort



THE BULK BIAS

Faster small steps are better than bigger steps

IT'S NOT ABOUT CODE QUALITY

```
@doc """
Adds Claude's response to the simulation and broadcasts it to all clients.

## Parameters

* `simulation_id` - The UUID of the simulation
* `response_text` - The text response from Claude
* `customer_name` - The name of the customer (from metadata)

## Returns

* `:ok` on success
* `{:error, reason}` on failure
"""

def add_ai_response_to_simulation(simulation_id, response_text, customer_name \\ "Claude") do
  # Create a message struct for Claude's response
  claudie_message = %{
    sender: customer_name,
    content: response_text,
    session_id: "claude-ai",
    timestamp: DateTime.utc_now(),
    # Optional avatar for UI
    avatar: get_avatar_path(customer_name)
  }

  # Add to UserStories.SimulationServer
  UserStories.SimulationServer.add_ai_response(simulation_id, claudie_message)
end
```

The code quality can be actually pretty good!

BIG STEPS, BIG PROBLEMS

IT'S ABOUT CODE QUANTITY

lib/zeroslides_web/live/user_stories_live.ex

Código · Versão 6

```
defimpl Live
  for _mounting
  do
    simulation_id = session_id
    session = session
    # Use the simulation_id
    # from the URL params
    handle_mount(simulation_id, session, socket)
```

lib/zeroslides/simulations/registry.ex

Código

```
defmodule Zeroslides.SimulationsRegistry
do
  @moduledoc """
  Registry for simulation
  processes.
  """
  @doc """
  Starts the registry for
  simulations.
  """
  # Agilizando esta linha é
  # só adicionar uma pipeline
  # com o session
  # dentro do escopo scope do
  # ZeroslidesWeb do
  # linhas existentes...
  # Scope "ZeroslidesWeb"
  # pipe_through :browser
```

Update to lib/zeroslides_web/router.ex

Código · Versão 3

```
# Agilizando esta linha é
# só adicionar uma pipeline
# com o session
# dentro do escopo scope do
# ZeroslidesWeb do
# linhas existentes...
# Scope "ZeroslidesWeb"
# pipe_through :browser
```

Adicione ao lib/zeroslides/application.ex

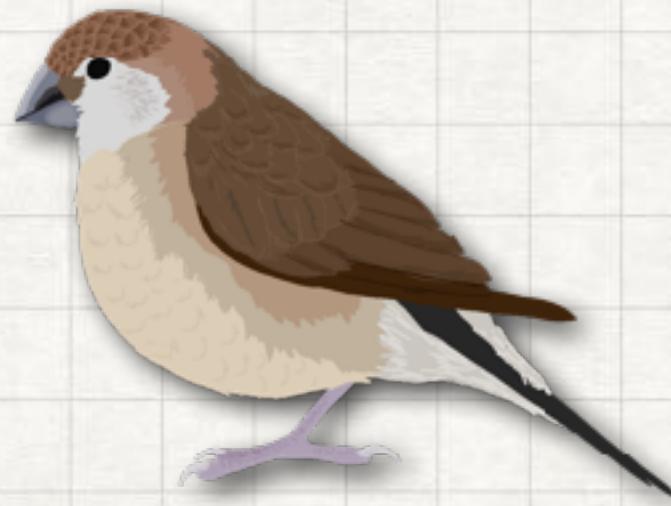
Código

```
# Adicione estas linhas nos
# children do seu Supervisor
# children = [
#   ... outros children
#   # Registry para processos de
#   # simulação, keys: :unique,
#   # ZeroslidesWeb]
```

What happened to baby steps, KISS, YAGNI, etc?

THE HUMMINGBIRD APPROACH

SPARROW



The standard way
to code

EAGLE



More distance
covered per
flap

HUMMINGBIRD



Quicker and
more precise
flaps

MACRO VS MICRO

Generate a screen where I can start multiple user story sessions. I want to be able to pause and see details such as the number of messages sent by each student.

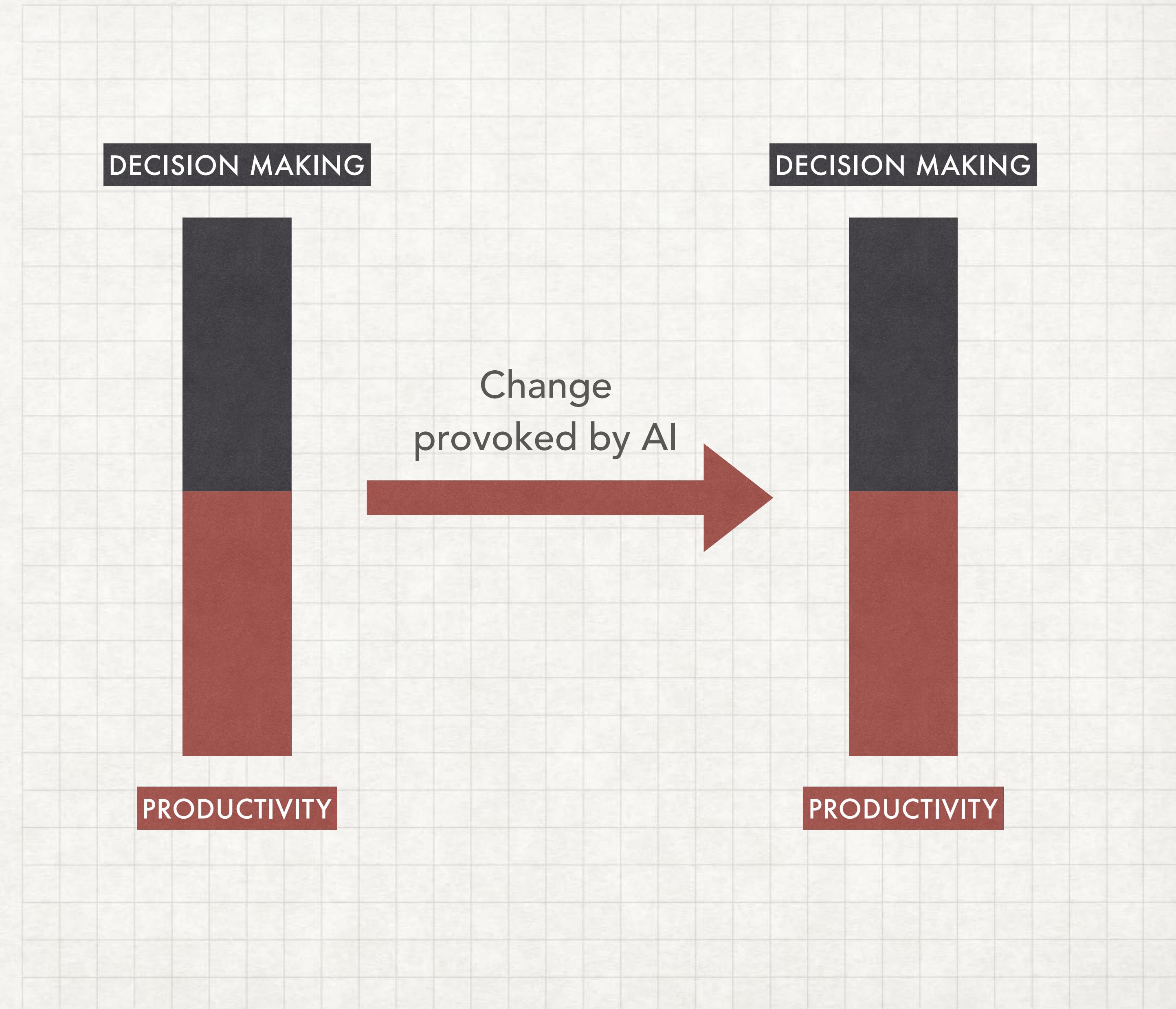
"Implement a GenServer that follows this interface specification and handles these edge cases..."

"Write a parser for this custom format using nimble_parsec with these requirements..."

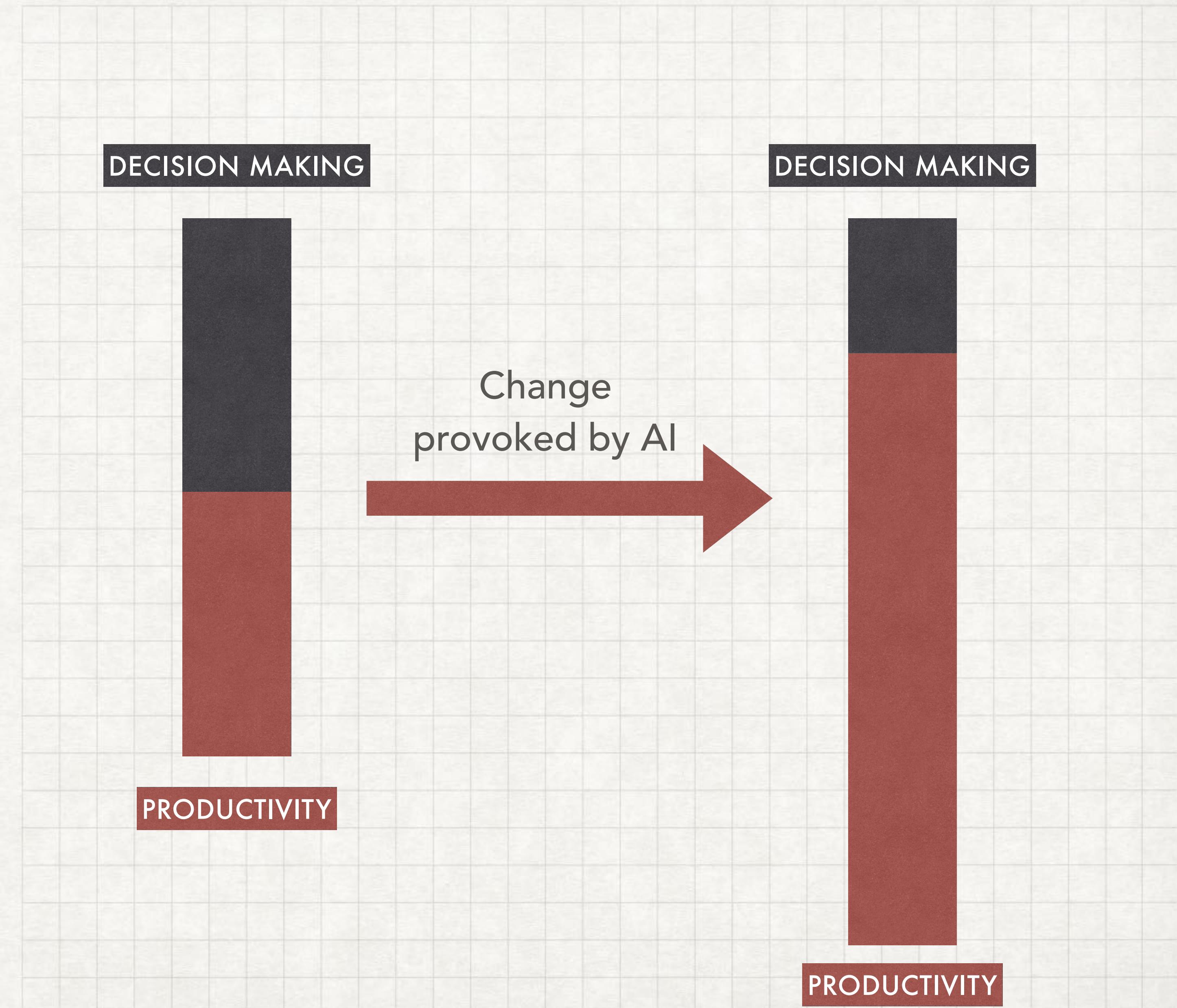
THE LEARNING ATROPHY

*You can only generate smaller changes if you
understand the lower levels of your code*

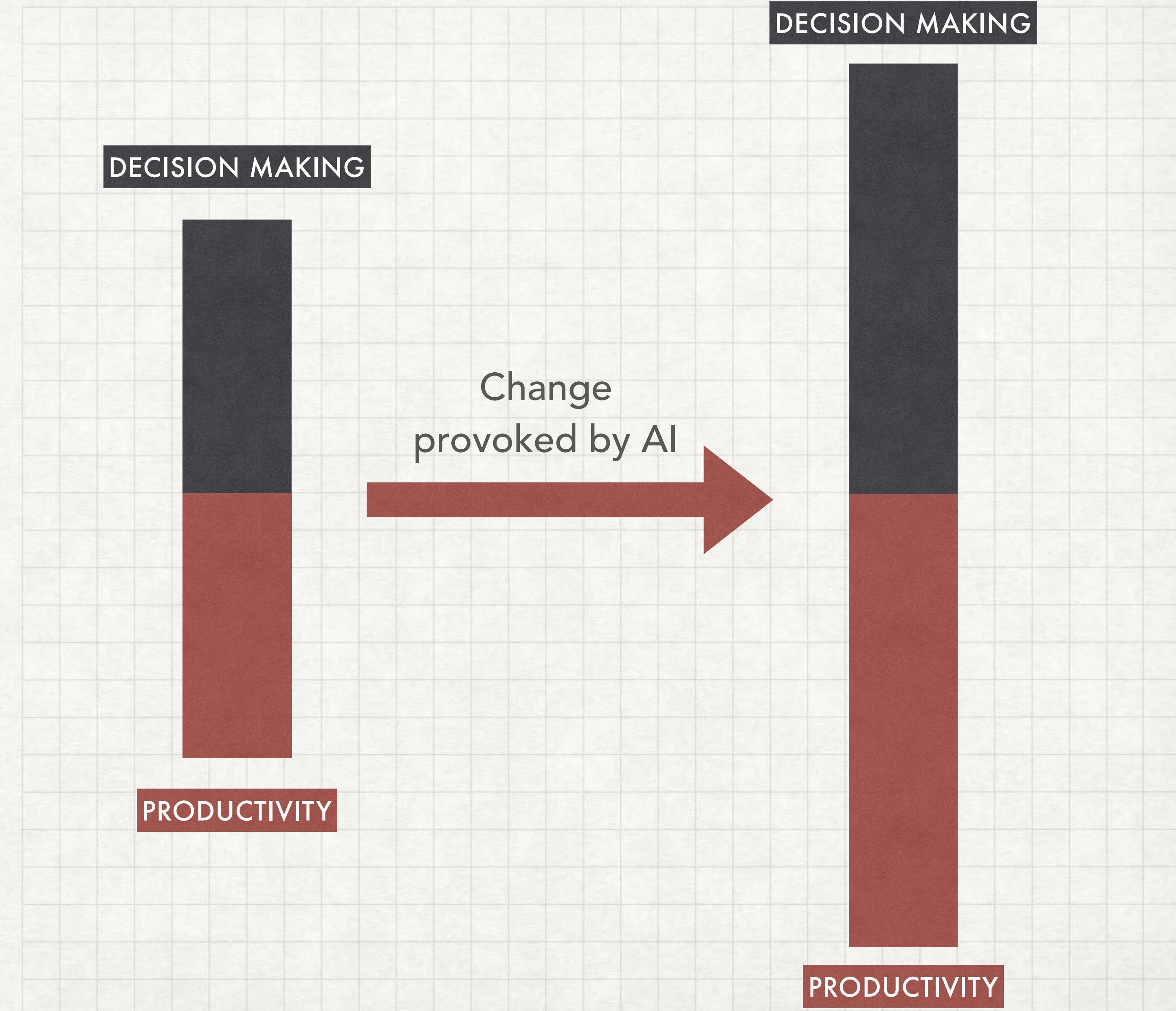
USING AI AUGMENTED CODE EXPECTATION



USING AI AUGMENTED CODE EXPECTATION



USING AI AUGMENTED CODE REALITY



“ ”

It's that they are both naive and enthusiastic. They are the toddlers of the industry, untouched by the world-weary cynicism of the rest of us. Unlike us, they don't know that **“you do it this way because we've always done it this way.”**

They don't know that an idea is ridiculous, or that something can't be done. Like toddlers, they're happy to try stuff and learn, and they almost expect to fail often—it's no big deal.

— *Dave Thomas, Agile Manifesto Author, about Junior Developers*

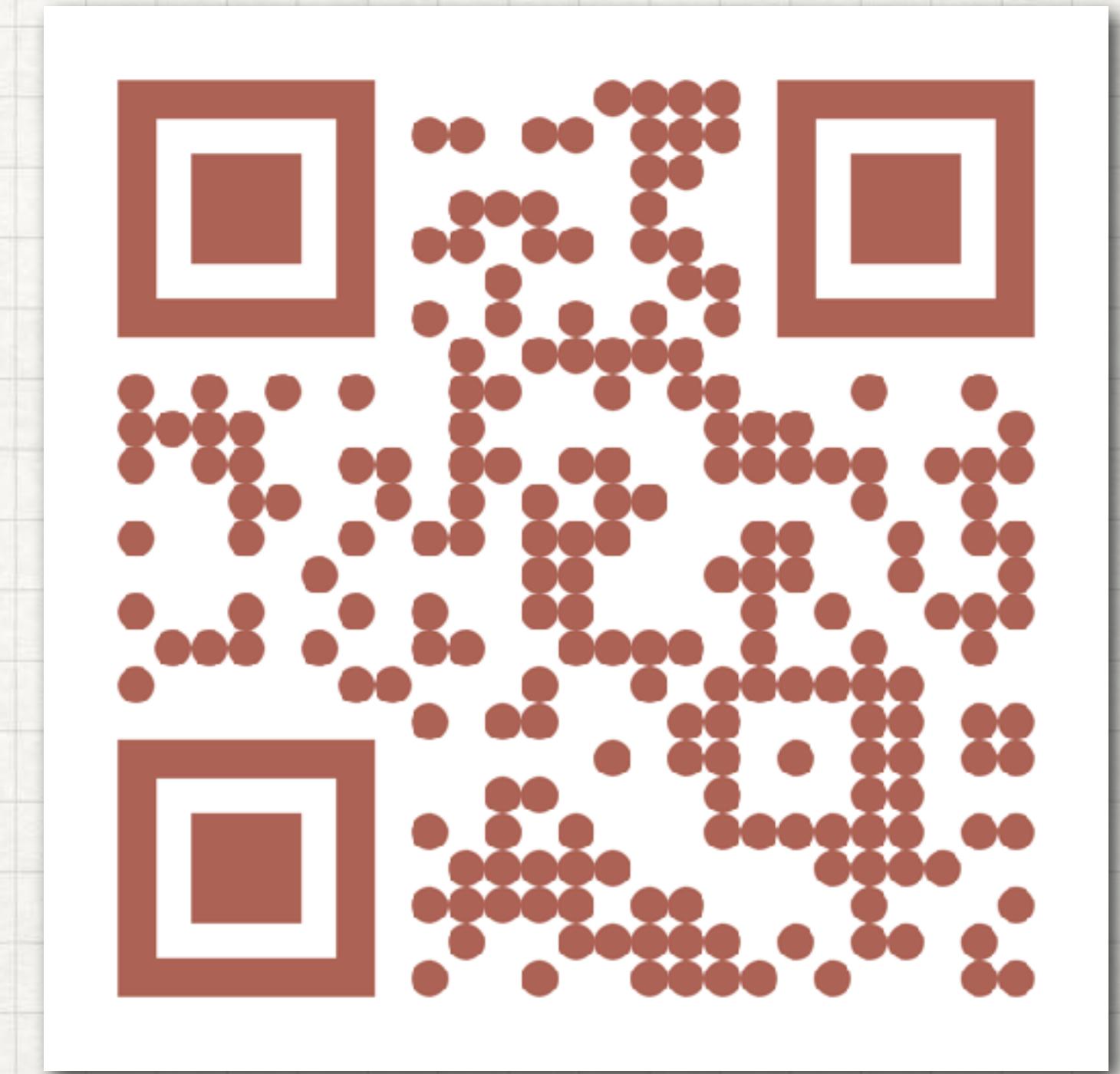
“ ”

CONCLUSION

- **Diversify communication:** draw, speak, and move beyond text-only interaction. Using an ubiquitous language is also important.
- **Cultivate curiosity:** ask questions to your AI and make it challenge your assumptions
- **Establish boundaries:** reduce context size, break down problems, and keep feedback loops short
- **Keep experimenting and learning:** use AI as an "amplifier of intention" rather than a "replacement of skill."

THANKS!

<https://mariomelo.com>



You can check my blog here