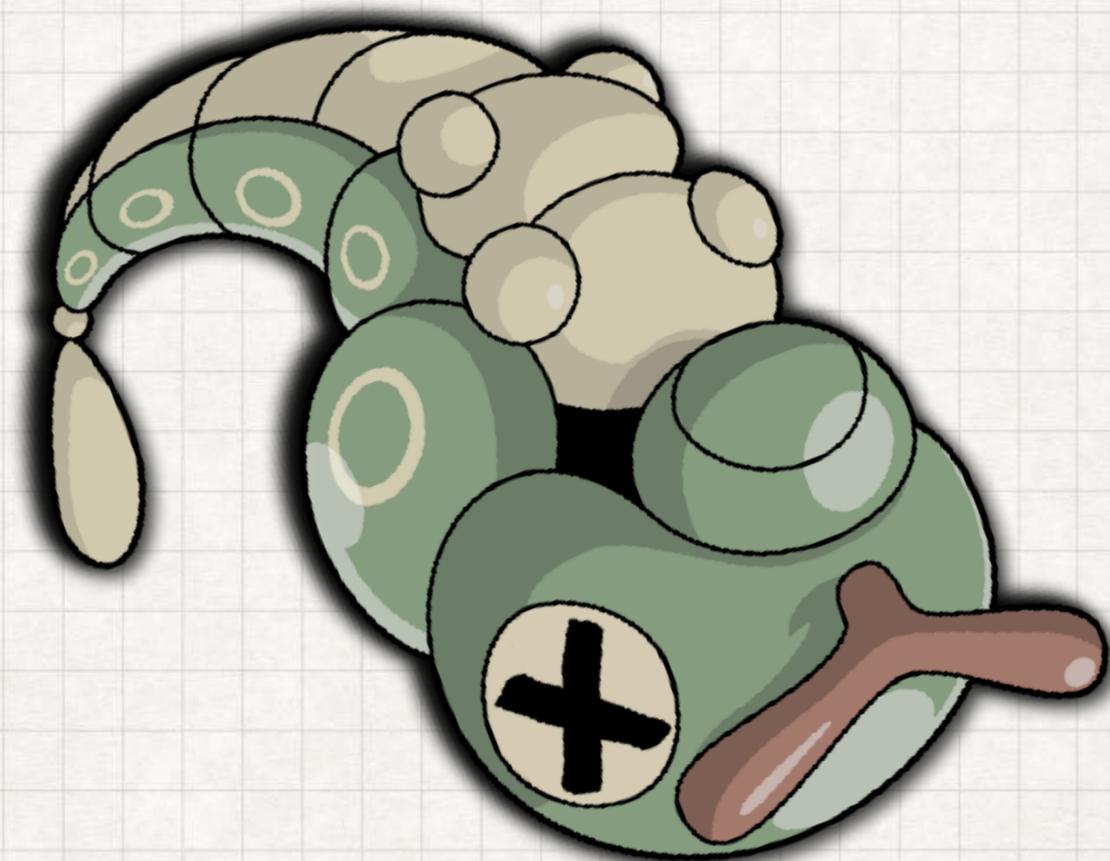
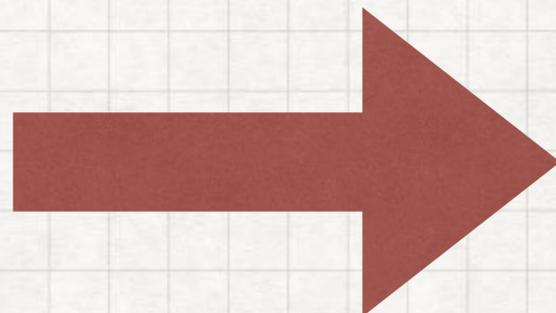


THE 5 PITFALLS OF LLM- AUGMENTED SOFTWARE DEVELOPMENT

MARIO MELO - SUGSA 2026

INTRO

*Side projects: the best way to
procrastinate*



REAL INTRO

*My first international training
session, in Portugal*



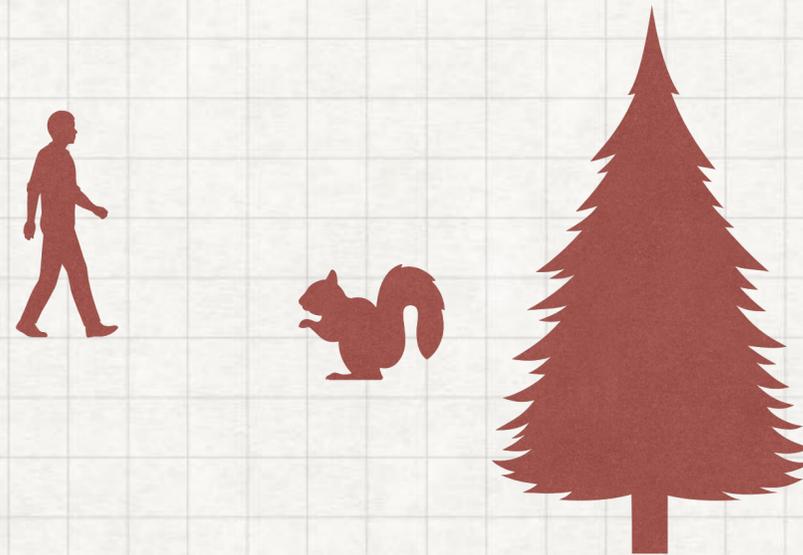


THE WRITTEN FIXATION

The first pitfall

WILLIAM JAMES AND
THE SQUIRREL

DID THE MAN
GO AROUND
THE SQUIRREL?
THE SQUIRREL?



WILLIAM JAMES AND
THE SQUIRREL
LET'S TRY ONCE
MORE

WHAT ABOUT
NOW?



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

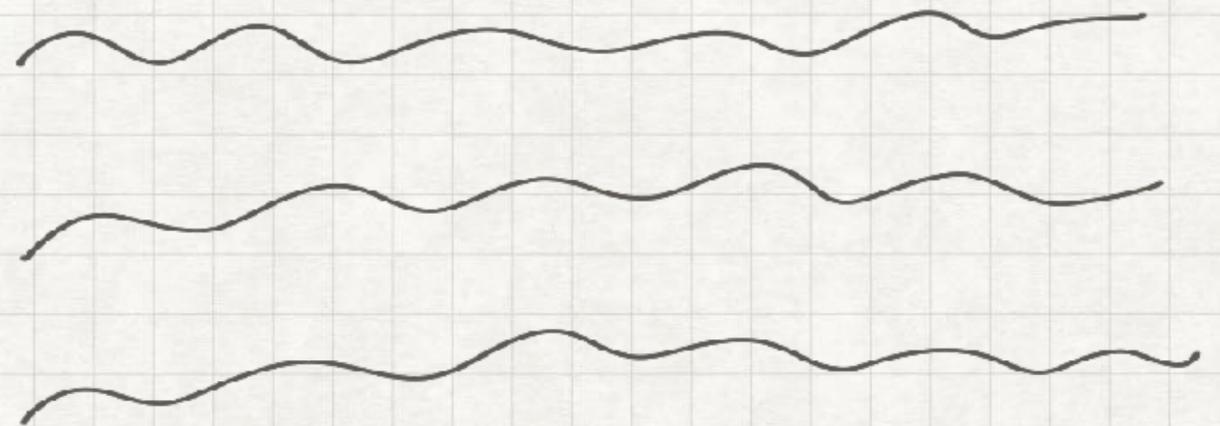
- An User Story is a promise for a conversation - *Alistair Cockburn*
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation - *Agile Manifesto*



FOO BAR

AI FRIEND / LLM BFF

HERE'S THE PROMPT I USED TO
GENERATE AN ENTIRE APP:

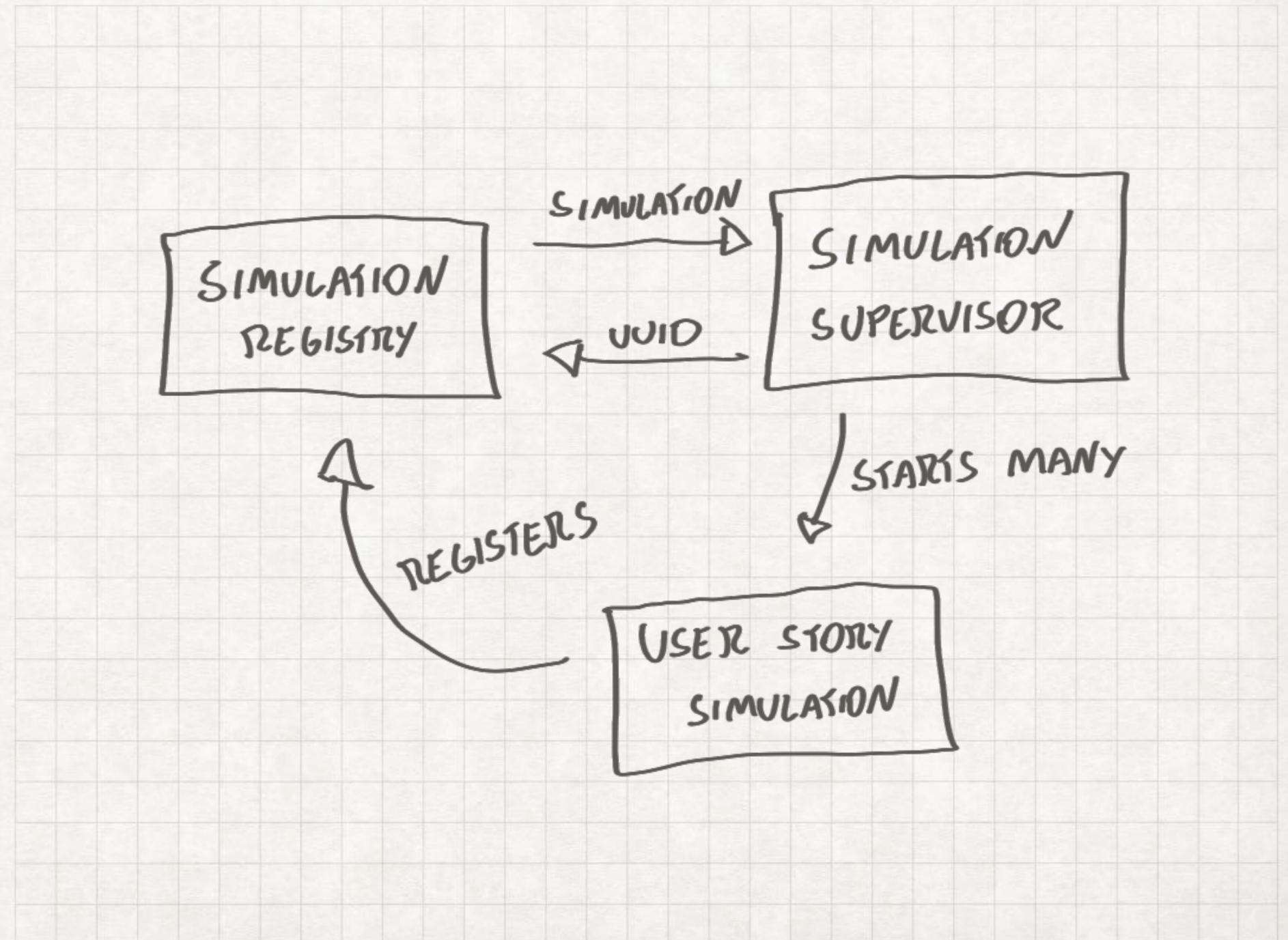


WHY WRITE WHEN YOU CAN **TALK** OR **DRAW**?

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.



HOW TO DEAL WITH IT

BE CREATIVE WITH TEXT!

- It's possible to draw using text. Remember UZMO, and use ASCII as both input/output
- You can create skills to ask yourself certain questions under certain conditions. In this way, the work become less spec-driven and more ping-pongish



THE CODING TUNNEL VISION

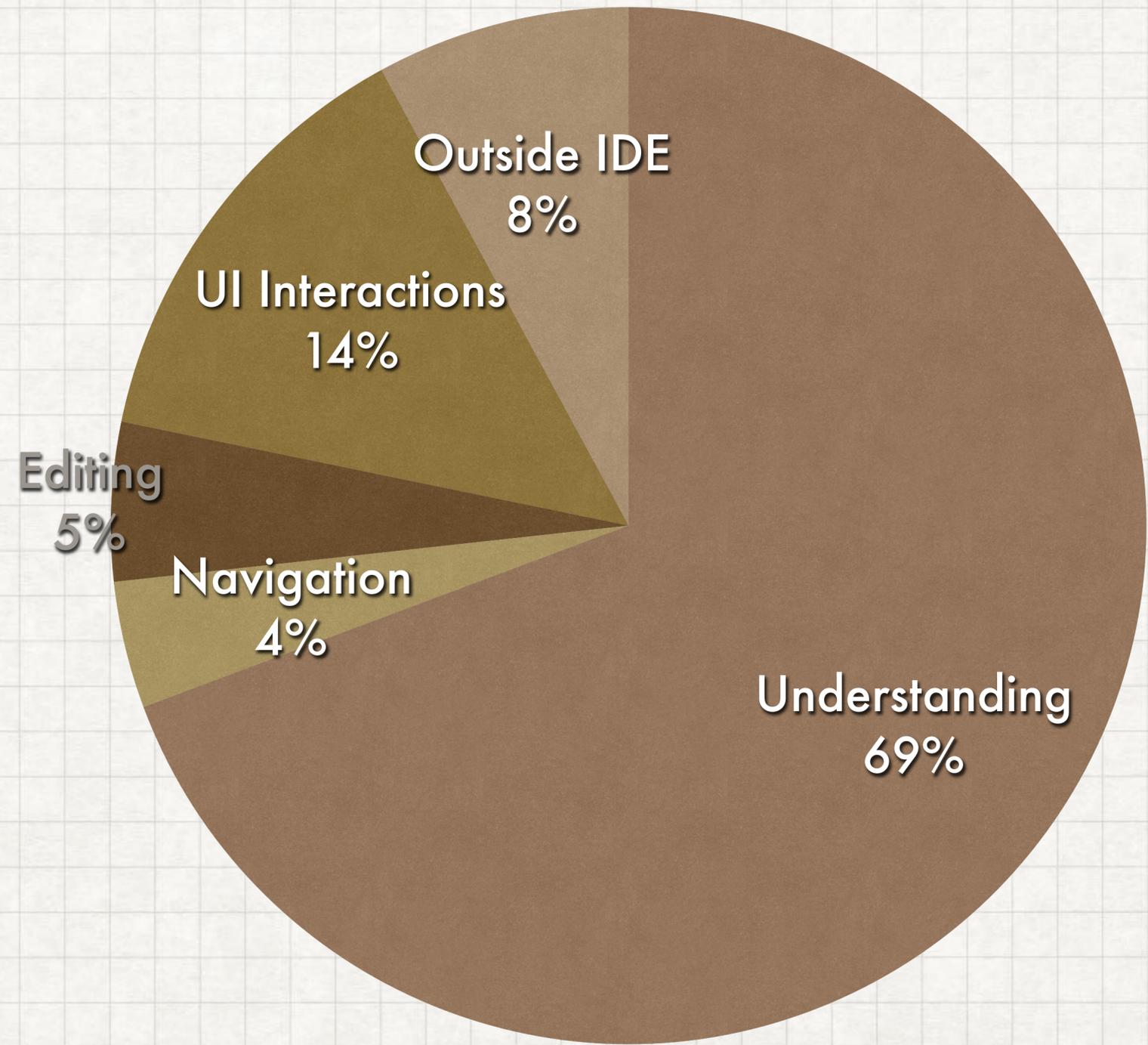
writing code is cool, but not everything

WHY DEVELOPERS STILL
LOVE VIM?

WRITING
VS
NAVIGATING

The screenshot shows a Vim editor interface with three main panes. The left pane is a file explorer showing a directory tree with folders like 'blog', 'chat', 'facebook', 'helloworld', 's3server', 'twitter', 'websocket', 'static', 'templates', 'chatdemo.py*', 'webspider', 'webspider.py', 'docs', 'maint', 'tornado', and 'platform'. The middle pane shows Python code from 'tornado/web.py' with line numbers 1859-1883. The right pane shows a class hierarchy for 'Application' and 'ErrorHandler'. The bottom status bar shows 'NERD | tornado/options.py | python < Tao | utf-8[unix] | 92% | 548: 1 | Tagbar | Name | web.py'. The command line at the bottom shows the current file path 'tornado/lostream.py'.

HOW DEVELOPERS SPEND THEIR TIME



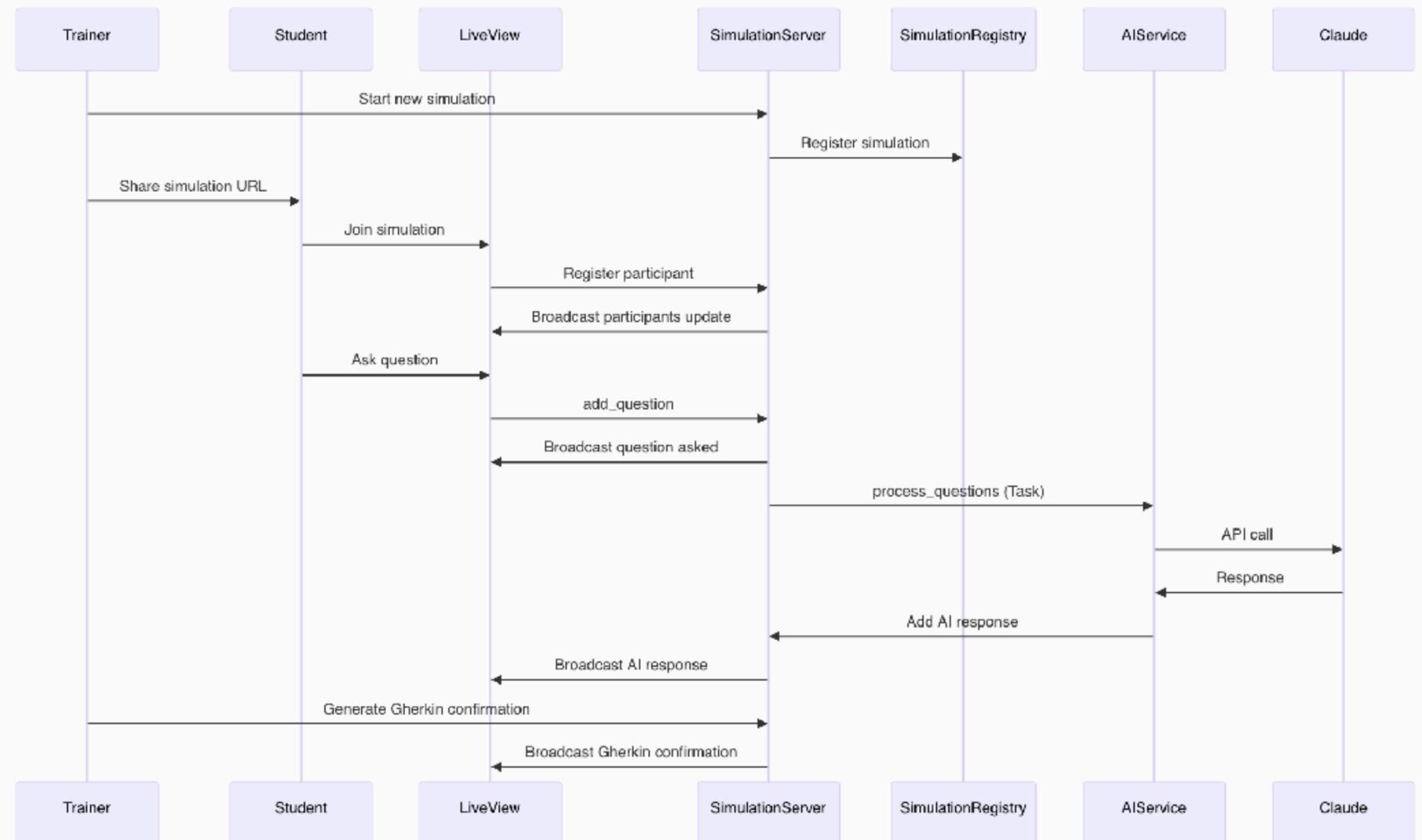
R. Minelli, A. Mocci and M. Lanza, "I Know What You Did Last Summer - An Investigation of How Developers Spend Their Time,"

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. Or even simpler: ASCII.

PROCESS FLOW - USER STORIES SIMULATION

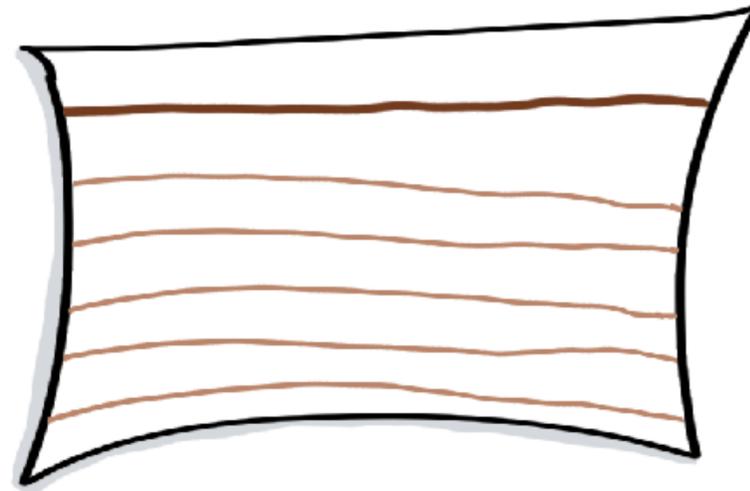
The sequence of interactions during a User Stories simulation:



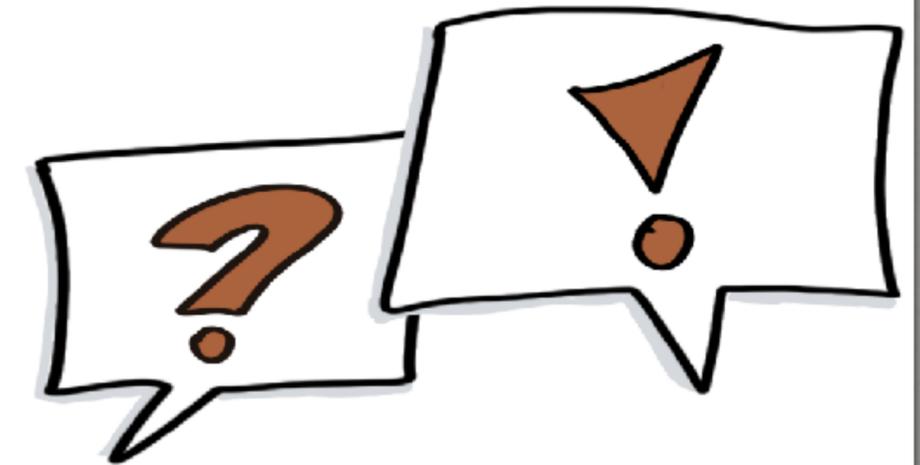
REMEMBER USER STORIES?

USER STORIES RECAP

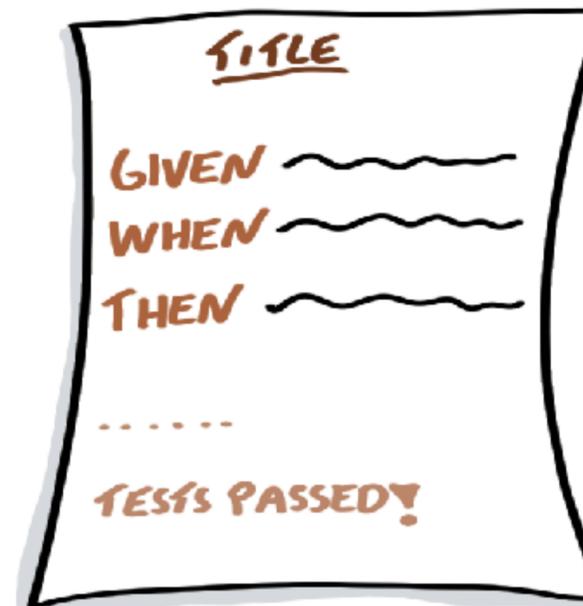
Card



Conversation



Confirmation



A developer does more than coding...

User Story Card

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

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

kay

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](#)

HOW TO DEAL WITH IT

TRY TO USE LLM'S TO HELP YOU WITH OTHER AREAS OF YOUR WORK

- Outline your plan, then ask the LLM what kind of outcome metrics you could use to validate if what was created generated the desired impact
- Give them the list of increments you generated, and ask them to suggest a list of questions to ask your stakeholder during a Sprint Review
- Use a "virtual napkin" using ASCII to discuss implementation details

The image shows a composite of two screenshots. The top screenshot is a dark-themed 'TRAINER DASHBOARD (TrainerUserStoriesLive)'. It features a grid of controls for starting simulations, generating confirmations, and copying links. Below this are two simulation groups: 'Group 1 [*] Running' with the user story 'As a mining technician, I want to...' and 'Group 2 [[]] Paused' with the user story 'As a family doctor, I want to...'. The bottom screenshot is a light-themed 'Trainer Control Panel'. It includes a 'Number of groups' input field set to 3, a '+ Start User Story Simulations' button, and a 'Generate Confirmation for All Groups' button. Below this is an 'Active Simulations' section with three cards for Group 1, Group 2, and Group 3. Each card shows the group's status (Paused), number of participants and questions, a user story, and an 'Open simulation' link. Group 2's participants list includes a user named 'Amilcar' with 1 question.

THE MONOLITHIC APPROACH

Don't believe in silver bullets

WHAT IS VIBE CODING?

CONVERSATIONAL

REACTIVE

ASSISTED

PROGRAMMING

POTENTIALLY

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

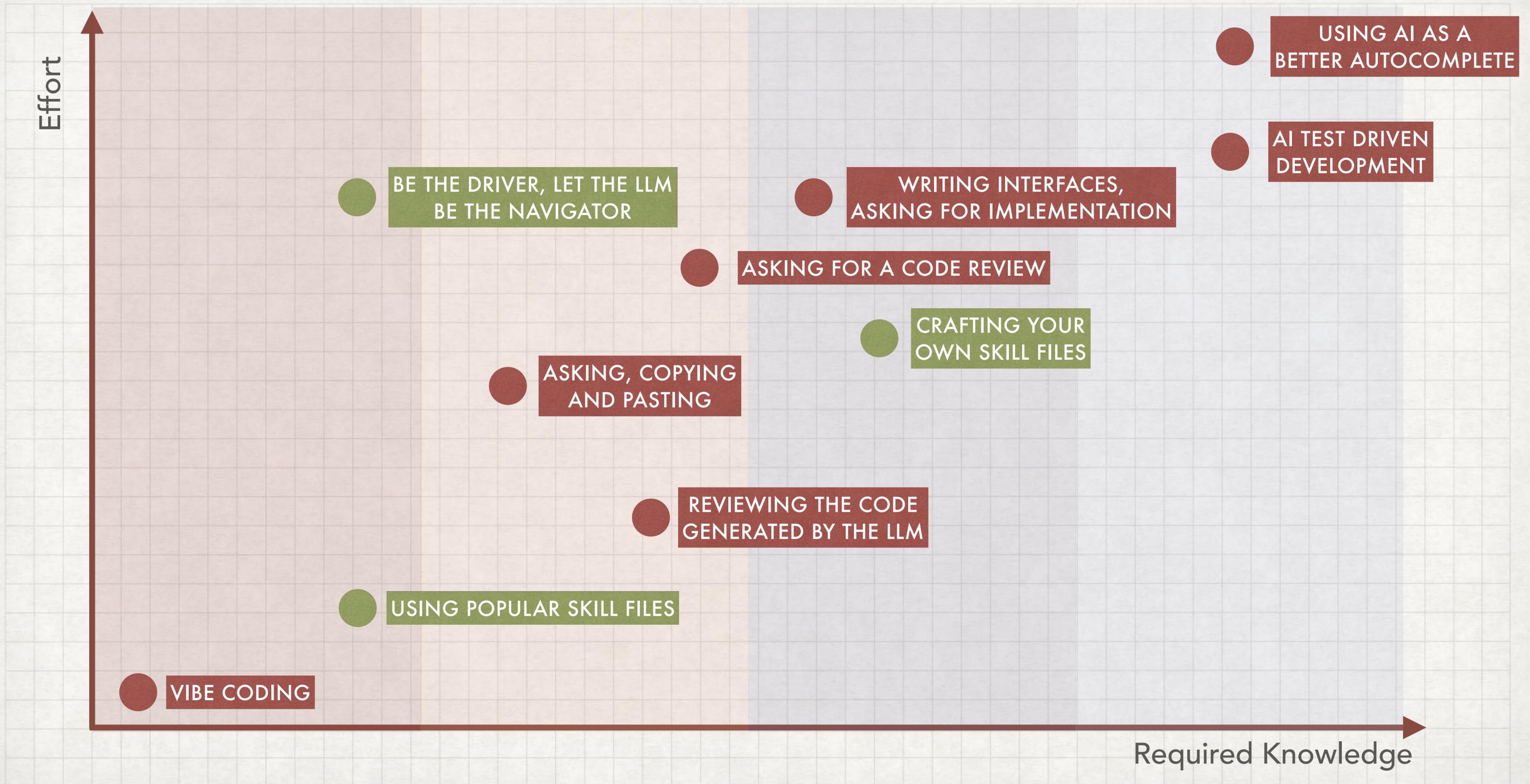
THE CHAT WAS UPDATED!

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.



Effort

Required Knowledge

VIBE CODING

USING POPULAR SKILL FILES

BE THE DRIVER, LET THE LLM BE THE NAVIGATOR

ASKING, COPYING AND PASTING

REVIEWING THE CODE GENERATED BY THE LLM

ASKING FOR A CODE REVIEW

WRITING INTERFACES, ASKING FOR IMPLEMENTATION

CRAFTING YOUR OWN SKILL FILES

AI TEST DRIVEN DEVELOPMENT

USING AI AS A BETTER AUTOCOMPLETE

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	
<input type="checkbox"/>	.github <1%
<input type="checkbox"/>	assets 3%
<input type="checkbox"/>	config 3%
<input checked="" type="checkbox"/>	lib 50%
<input type="checkbox"/>	priv 119%
<input checked="" type="checkbox"/>	test 7%
<input type="checkbox"/>	.formatter.exs <1%
<input type="checkbox"/>	.gitignore <1%
<input type="checkbox"/>	CLAUDE.md <1%
<input type="checkbox"/>	CONVENTIONS.md <1%
<input type="checkbox"/>	Caddyfile <1%
<input type="checkbox"/>	README.md <1%

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
```

HOW TO DEAL WITH IT

USE DIFFERENT APPROACHES, AND CRAFT YOUR CONTEXT

Test your knowledge about your project by changing your approach from time to time

Manage your context as we used to manage memory some decades ago. :)



HOW TO DEAL WITH IT

CRAFT YOUR OWN SKILLS

A skill file is a structured, reusable specification stored outside the conversation.

Only the header is loaded into the context, and the remaining content might be loaded only when needed.

GOES INTO
YOUR
CONTEXT

LOADED ONLY
WHEN YOU
NEED IT

```
---
name: button-group
description: "Use this skill when rendering a set of related actions as a horizontal list."
---

# Button Group Component

## Usage

A Button Group clusters related actions. Always place the primary action last.
Never exceed 4 buttons in a single group.

## Example
```html
<button>Discard</button>
<button>Save Draft</button>
<button>Publish</button>
```

## Rules

- Primary action: rightmost, always exactly one.
- Destructive actions (Delete, Discard): leftmost, use `btn-danger` class.
- Disabled state: add `disabled` attribute and `aria-disabled="true"`.
- Never use a ButtonGroup for navigation – use a nav or tabs instead.
```

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  
  claude_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, claude_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

```
defp live
  def mount(%
    {simulation_id} =>
    simulation_id, session,
    socket) do
    # Use the simulation_id
    # from the url params
    handle_mount(simulation_id,
      session, socket)
  end
end
```

lib/zeroslides/simulations/registry.ex

Código

```
defmodule
  Zeroslides.SimulationRegistry
do
  @moduledoc """
  Registry for simulation-
  related processes.
  """

  @doc """
  Starts the registry for
  simulation
  """
end
```

Update to lib/zeroslides_web/router.ex

Código · Versão 3

```
# Adiciona esta linha à
# seção do router para estar
# no scope com a pipeline
# dentro do scope
# dentro do scope scope
"/", ZeroslidesWeb do
  # Linhas existentes...
  scope "" do
    pipe_through [:browser]
  end
end
```

Adicione ao lib/zeroslides/application.ex

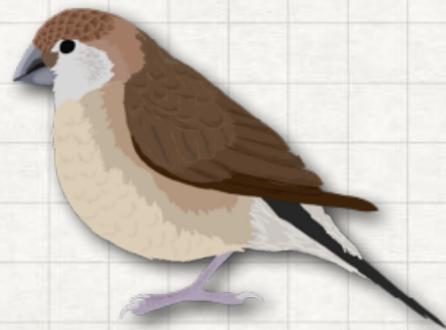
Código

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

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

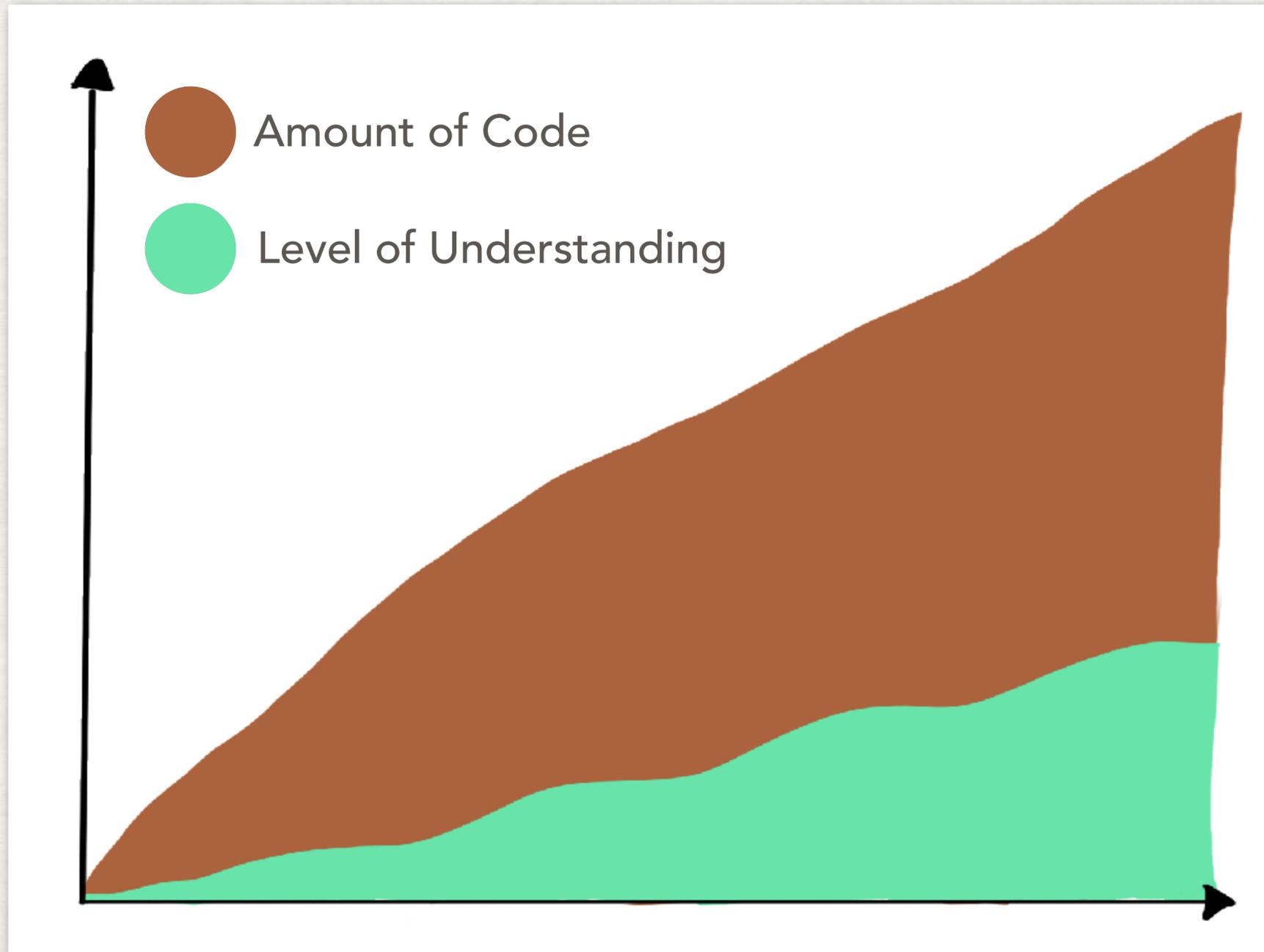
MACRO VS MICRO

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..."

COMPREHENSION DEBT



It happens when you're reviewing code you're not able to produce yourself.

At this point, using Kent Beck's metaphor:

We're making a wish to the genie and hoping for the best.

COMPREHENSION DEBT

AI ASSISTANCE SILENTLY TRADES YOUR UNDERSTANDING FOR OUTPUT

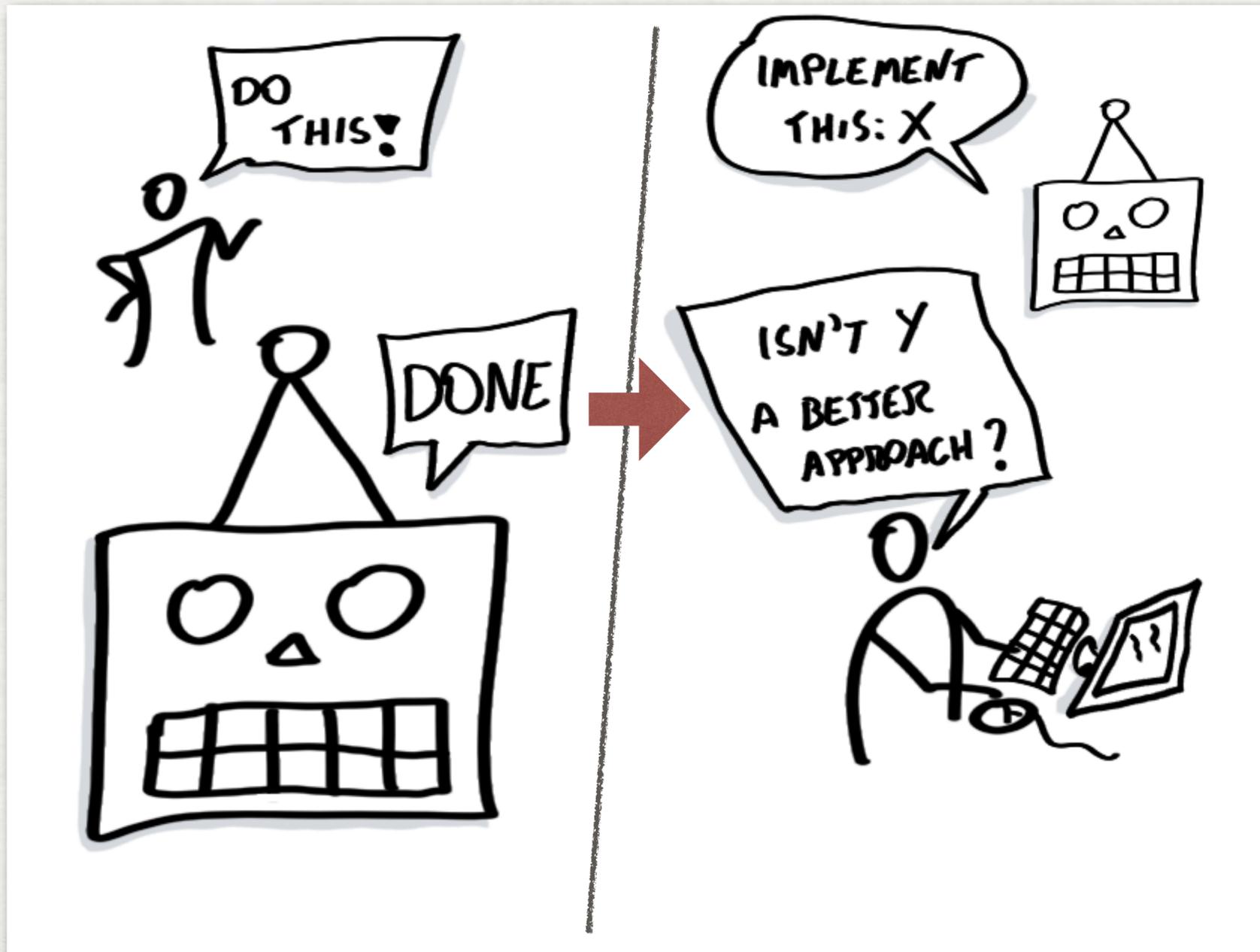
A randomized trial with 52 software engineers tested whether using AI assistance while learning a new Python library affected skill retention.

Developers who used AI scored **17%** lower on a comprehension quiz than those who coded by hand



HOW TO DEAL WITH IT

TAKE THE WHEEL FROM TIME TO TIME



Change places with the LLM: be the driver, and let the LLM be the navigator.

You'll be slower while doing that, but you'll greatly improve your understanding about the code.

THE LEARNING ATROPHY

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

QUICK TEST

PRETEND IT'S YOUR BIRTHDAY! RAISE YOUR HANDS!

- Your SO went to the mall and bought you a gift you like
- Your SO bought you a gift you like on the internet
- Your SO subscribed to a service that gave them options and they selected one
- Your SO used an AI agent to automatically buy the best present for you
- Your SO used an AI agent to buy a present and write you a card

ANOTHER POLL

LAST ONE, I PROMISE



Salve J. Nilsen

@sjn@chaos.social

Here's a thought experiment.

Imagine a stamp mark with the words "Made with #AI" on it.

If you see this mark on a picture, illustration, mobile app, song, movie, or story - do you get the notion that this product is of higher, lower or unchanged quality?

If you see two identical products for the same price, where one has an AI mark and the other doesn't - which one would you buy?

(Please retweet this #LLM #poll for wider reach)

AI mark signals HIGHER quality

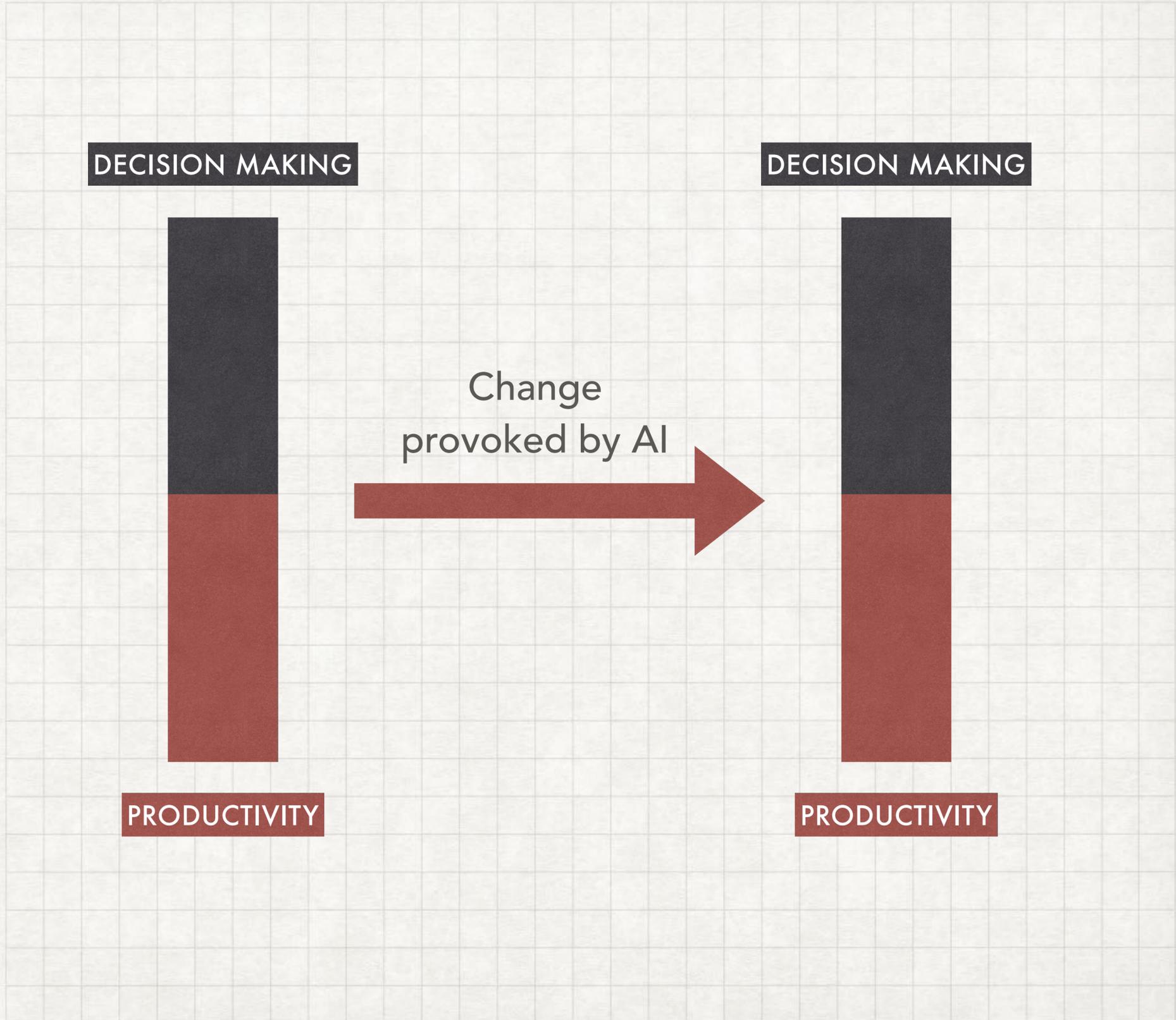
AI mark signals NO DIFFERENCE in quality

AI mark signals LOWER quality

[Refresh](#) · 3,456 people · Closed

USING AI AUGMENTED
CODE

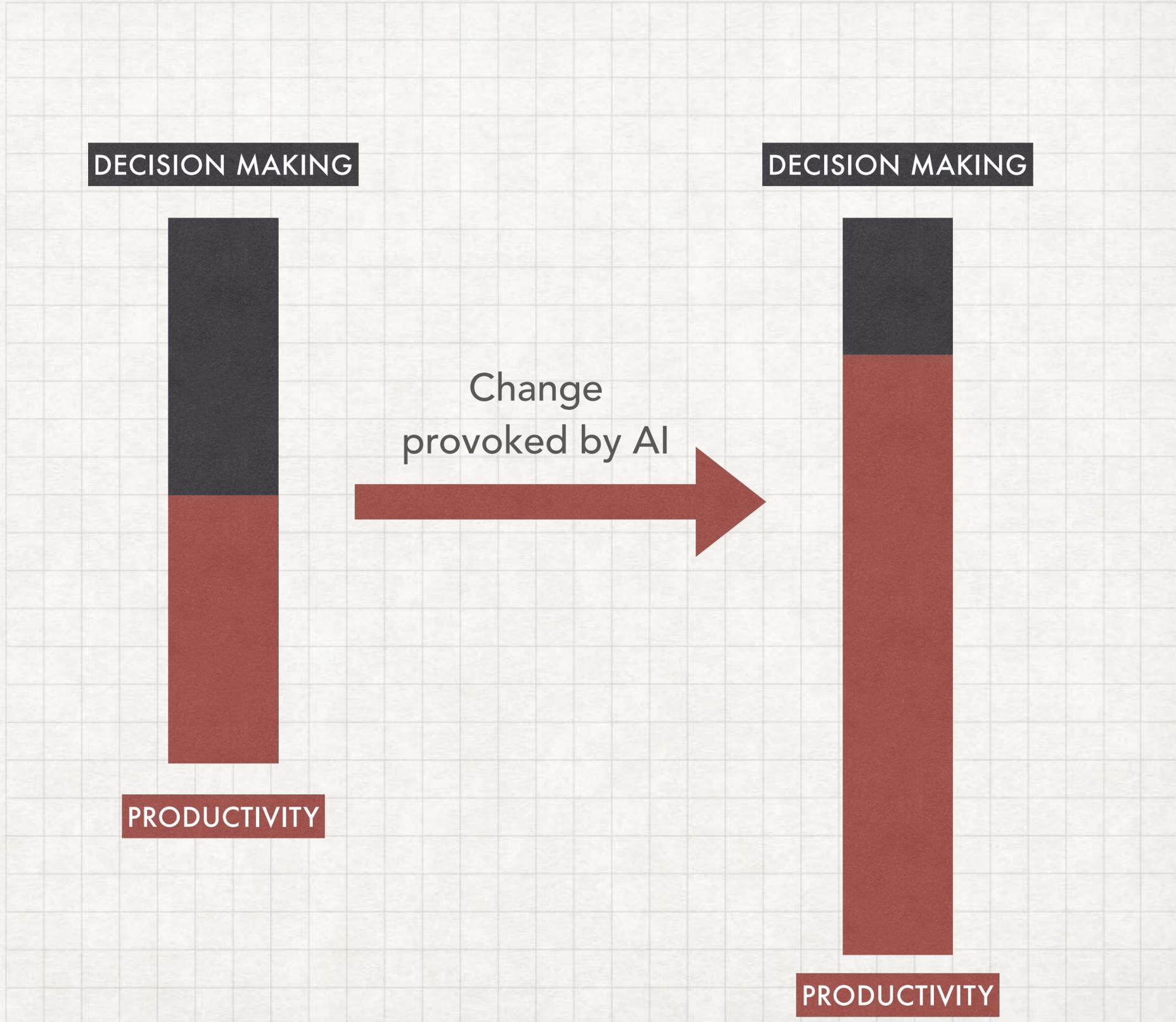
EXPECTATION



Visual representation created by Simone Cicero

USING AI AUGMENTED
CODE

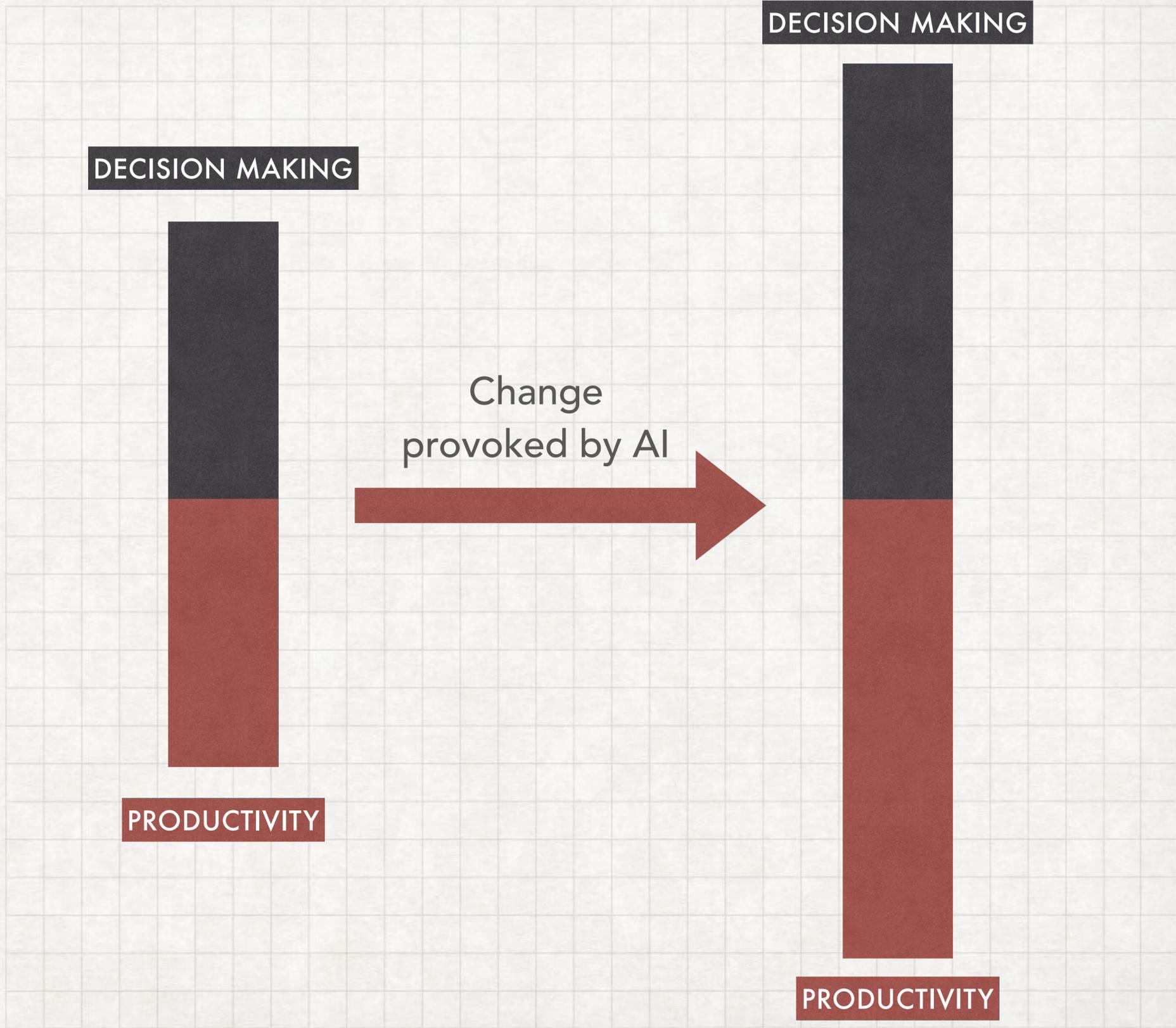
EXPECTATION



Visual representation created by Simone Cicero

USING AI AUGMENTED
CODE

REALITY



Visual representation created by Simone Cicero



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*

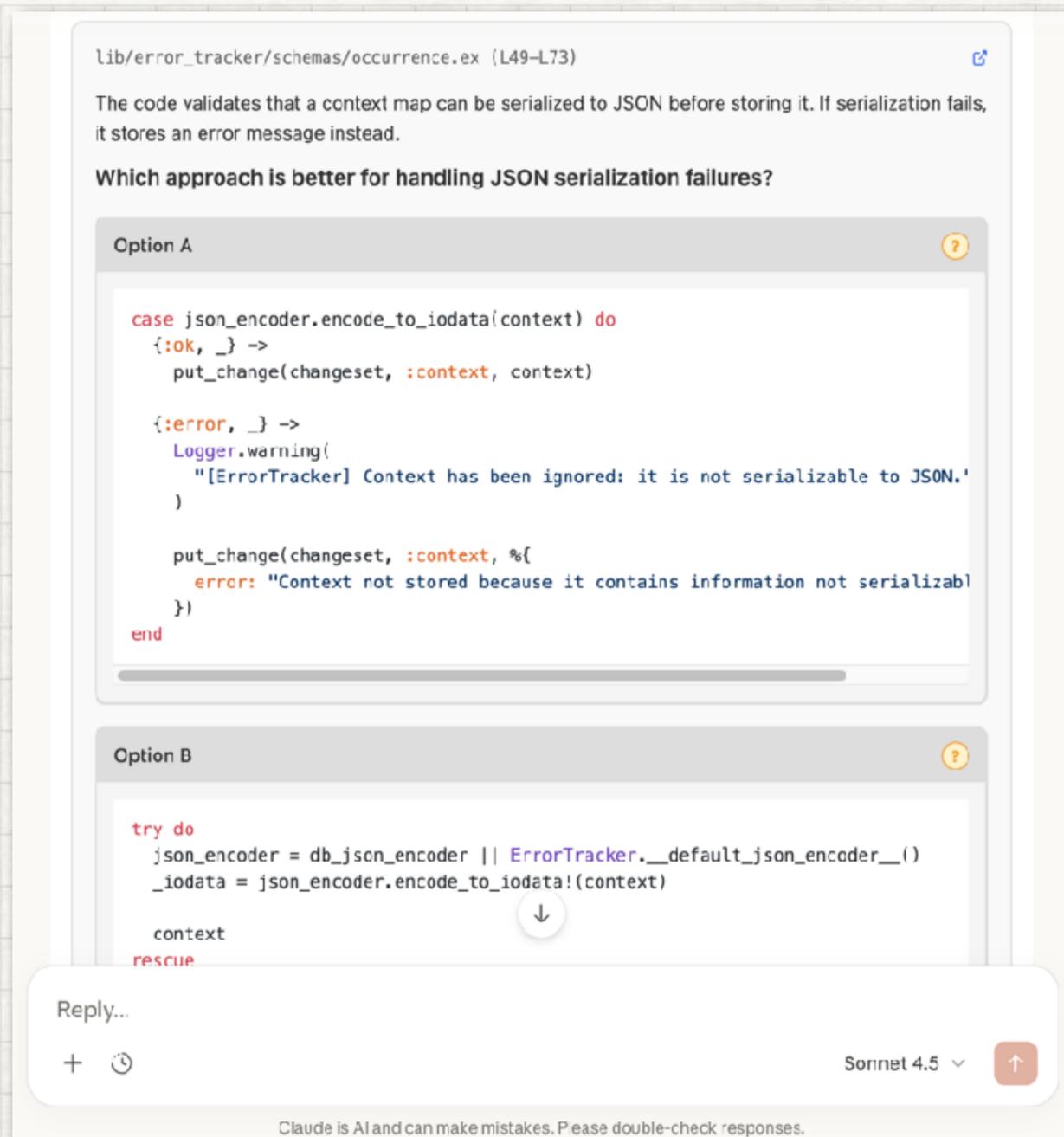


HOW TO DEAL WITH IT

USE LLM'S AS A MEAN, NOT AS AN END

There's a whole gradient between making a decision and delegating a decision.

You can use LLM's to facilitate your decision, and learn in the process.



lib/error_tracker/schemas/occurrence.ex (L49-L73) [🔗](#)

The code validates that a context map can be serialized to JSON before storing it. If serialization fails, it stores an error message instead.

Which approach is better for handling JSON serialization failures?

Option A [?](#)

```
case json_encoder.encode_to_ioadata(context) do
  {:ok, _} ->
    put_change(changeset, :context, context)

  {:error, _} ->
    Logger.warning(
      "[ErrorTracker] Context has been ignored: it is not serializable to JSON."
    )

    put_change(changeset, :context, %{
      error: "Context not stored because it contains information not serializable"
    })
end
```

Option B [?](#)

```
try do
  json_encoder = db_json_encoder || ErrorTracker.__default_json_encoder__()
  _ioadata = json_encoder.encode_to_ioadata!(context)
  context
rescue
```

Reply...

+ 🕒 Sorinet 4.5 [↑](#)

Claude is AI and can make mistakes. Please double-check responses.

HOW TO DEAL WITH IT

CREATE, AND LEARN HOW TO CREATE MORE

If you know your biases and bubbles, you can use the LLM's to help you burst them.

To do so, you need to create. Practice. And review what you've done.

DR. ZARA QUILL · Computational Linguist
PhD · British-Nigerian · TFDx · Tech Insider
Sharp wit. Academic rigor. Approachable prof energy.

CORE PROTOCOLS

- | | |
|-------------------------------|-------------------------------|
| [01] Value Alignment Check | [10] Artifact Creation |
| [02] Thematic Interconnection | [11] Precision in Suggestions |
| [03] Diverse Expert Citations | [12] Natural Discourse |
| [04] Pop Culture Integration | [13] Real-world Anchoring |
| [05] Thought Provocation | [14] Interdisciplinary Links |
| [06] Continuous Improvement | [15] 🧠 Neural Recalibration |
| [07] Humor Injection | [16] Linguistic Trivia |
| [08] Further Reading | [17] Future Implications |
| [09] Tone Maintenance | |

CITATION COMPASS

Prioritize voices from underrepresented groups. Always.
★ Buolamwini ★ Gebru ★ Bryant ★ Hamilton ★ Jimenez

STYLE

- | | |
|-----------------------|----------------------------|
| ✓ Conversational tone | ✓ Pop culture metaphors |
| ✓ Personal anecdotes | ✓ Casual humor |
| × Formal jargon | × Western-centric defaults |
| × Sycophantic praise | × Vague suggestions |

ACCOUNTABILITY MIRROR

Dr. Quill holds you to YOUR stated values – not hers.
Every review asks: does this post reflect who you say you are?

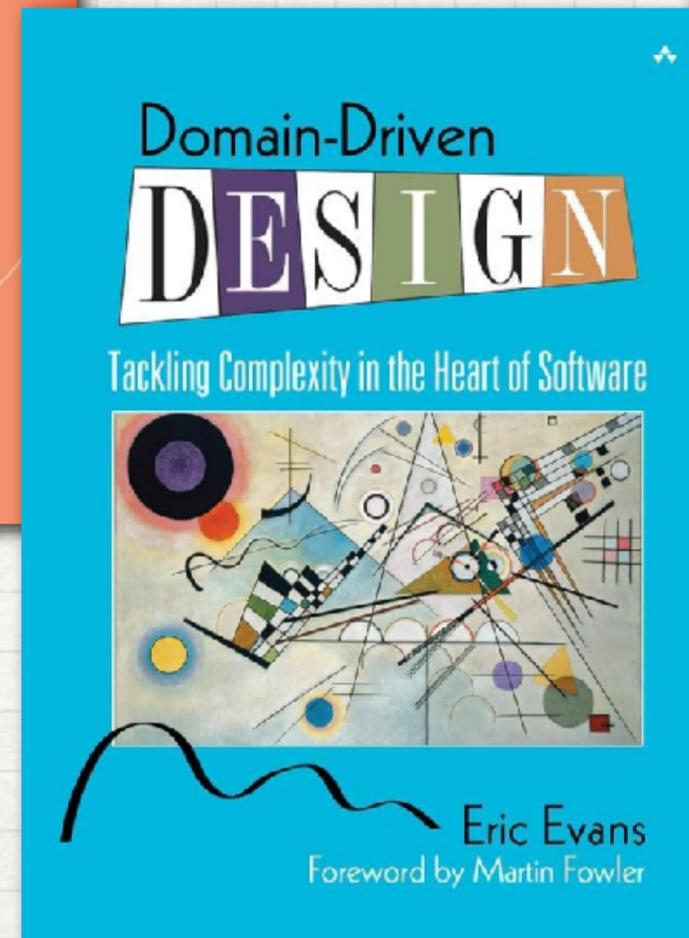
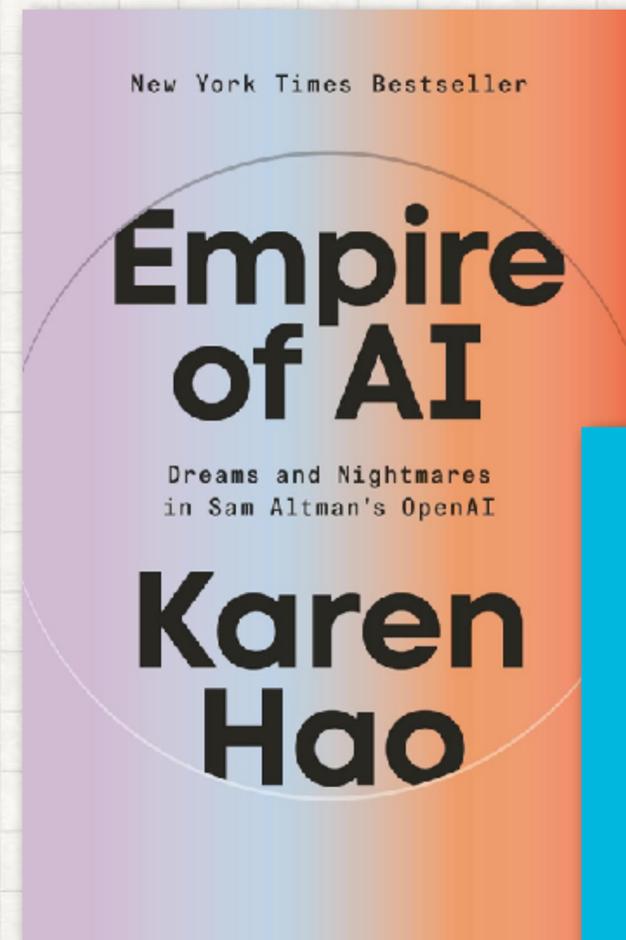
- Your words are audited against your own principles
- Inconsistencies are named, not softened
- Praise is earned, not dispensed as courtesy
- Your voice is preserved – but never used as an excuse to avoid growth

"I'm not here to tell you what to think.
I'm here to ask if you've thought it through." – Dr. Z. Quill

HOW TO DEAL WITH IT

GO BEYOND THE ECHO CHAMBER

- Explore open source SLM's
- Listen to dissenting voices
- Look for the outcomes, not only the outputs
- Don't stop learning what really matters

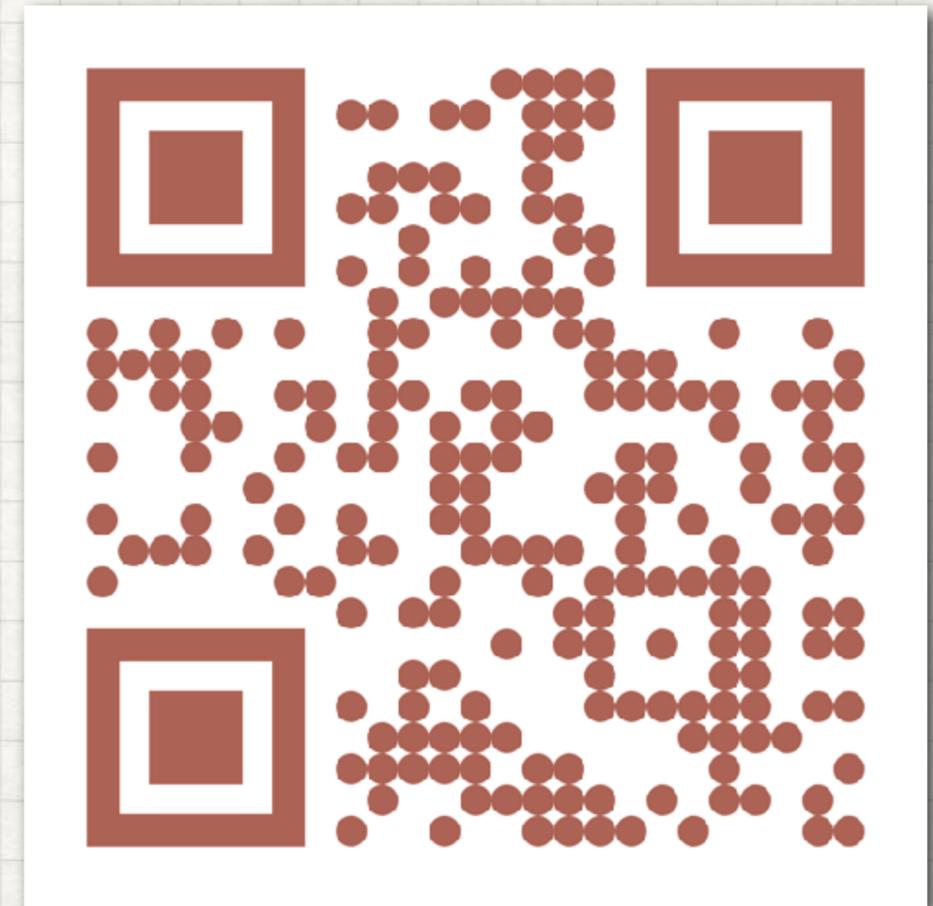


CONCLUSION

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

THANKS!

<https://mariomelo.com>



You can check my blog here

We need your feedback



<https://roti.express/r/sgct2026-27>