PMI AUSTIN · FALL 2026 PROFESSIONAL DEVELOPMENT DAY · OCT 2, 2026

AI Joined My Team.
What's My Role Now?

Clint Tuttle · Associate Professor of Instruction, McCombs School of Business, UT Austin

WHERE WE START

We're all at different places on the AI journey

1	2	3	4	5
Foundational	**Emerging**	**Operational**	**Scaled**	**Transformational**
Talking about AI, but nothing is really happening	**Experimenting** with pilots, but inconsistent.	AI is **being used** in real workflows with *some* standards.	AI is **scaled across the org** and built into many processes.	AI is **embedded in DNA** of org and **drives competitive advantage**

This may vary by pillar

- **Strategy** – Is there a clear, company-wide AI vision?
- **Product & Use Cases** – Are there real applications delivering value?
- **Governance** – Are there rules, ethics, and risk controls?
- **Engineering** – Are the tools, ML Ops, and infrastructure in place?
- **Data** – Is data high-quality, accessible, and well-governed?
- **Operating Model** – Are roles, teams, and processes built for AI?
- **Culture/People** – Are employees trained, aligned, and adopting AI?

Here’s one of my first non-chat experiments

- Use AI inside SQL using *Snowflake Cortex AI*
- Give an LLM a list of **21 values** (7 colors, 7 names, 7 animals)
- Ask: “Which of these are colors"

SELECT all rows from the table, no filter

```
37
38  select *
39  from AI_TEST;
40
```



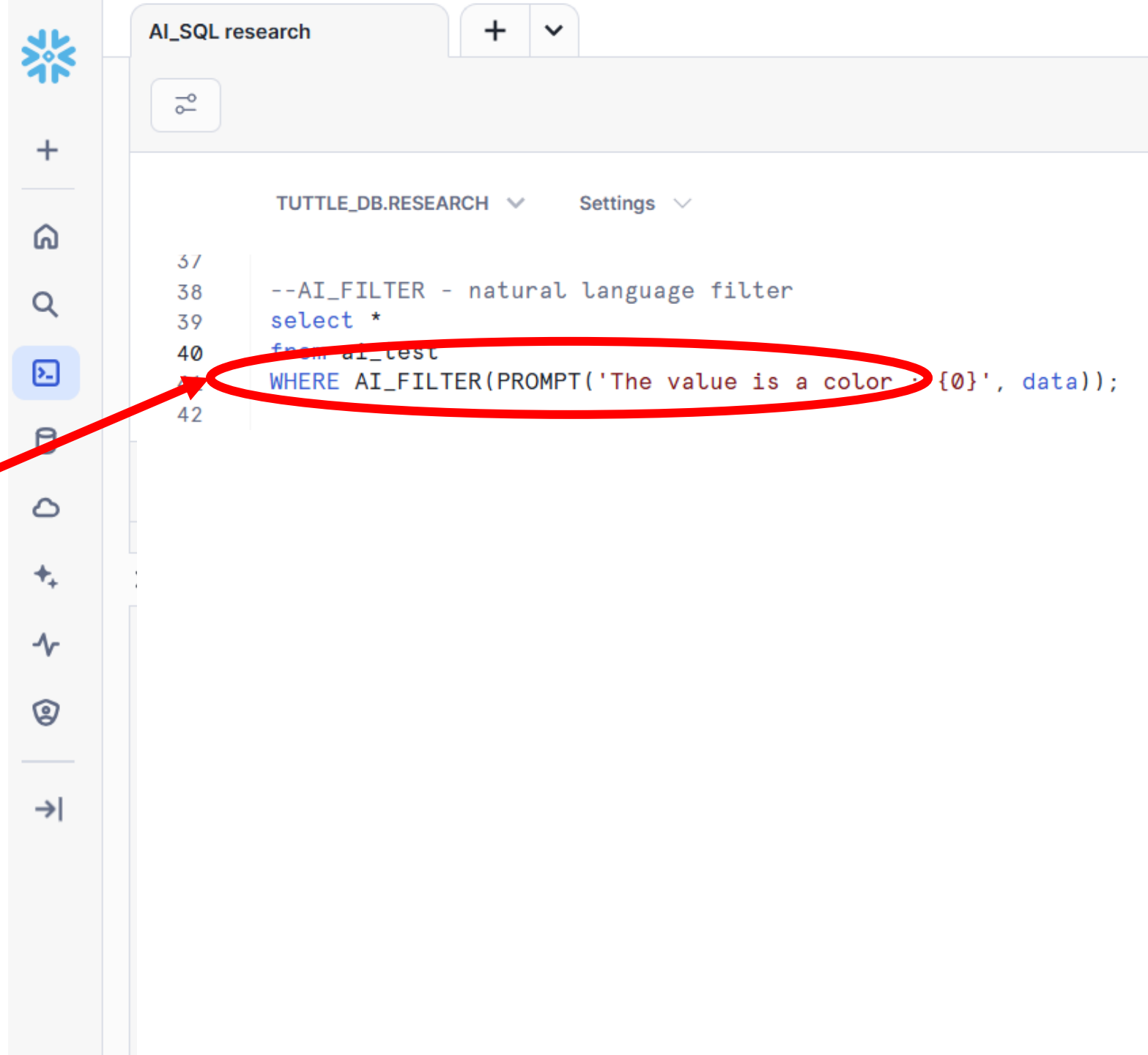
Results | Chart

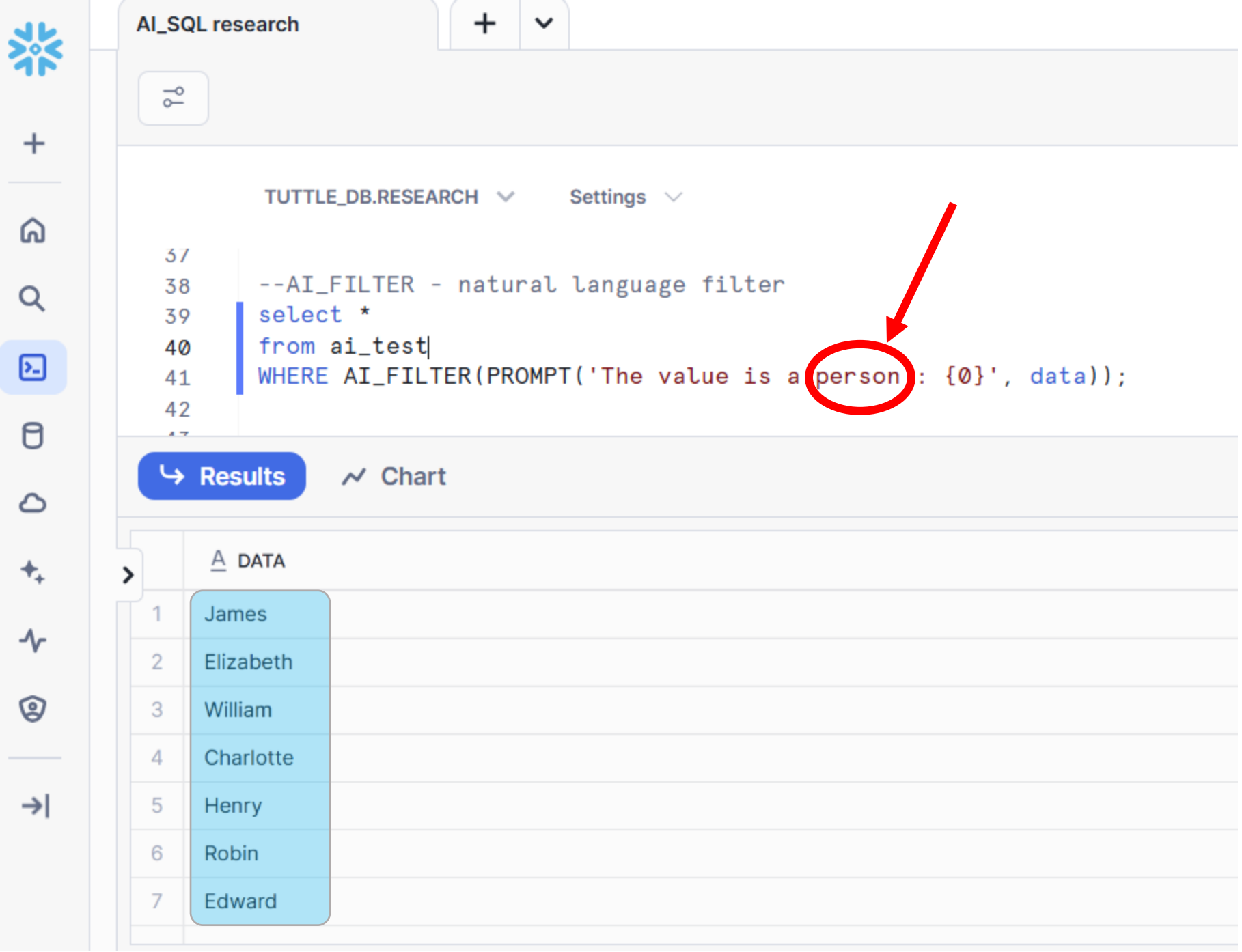
	DATA
1	James
2	Elizabeth
3	William
4	Charlotte
5	Henry
6	Robin
7	Edward
8	Dog
9	Cat
10	Horse
11	Cow
12	Parrot
13	Pig
14	Goat
15	Red
16	Blue
17	Yellow
18	Green
19	Orange
20	Gold
21	White

Code is usually **deterministic**

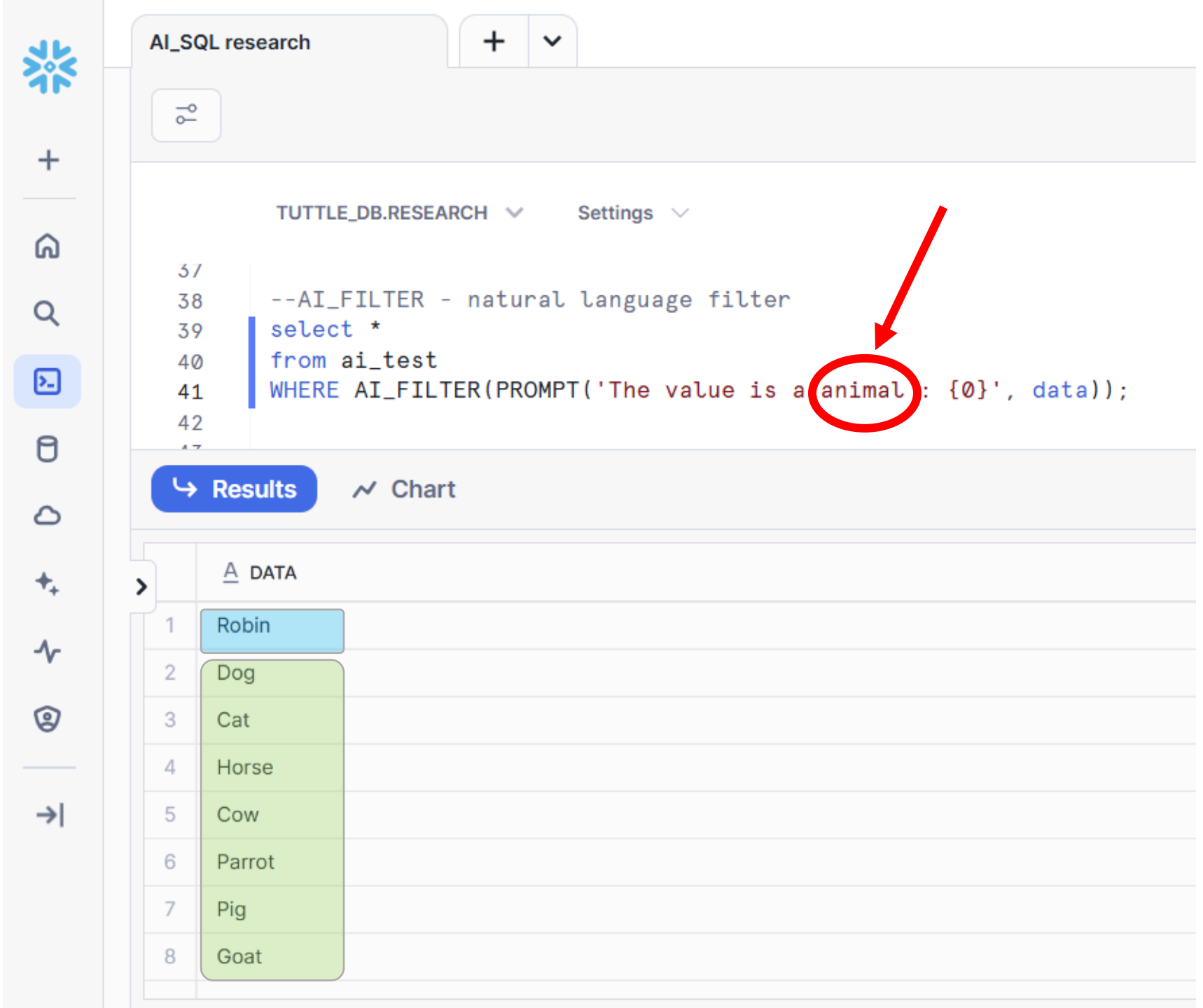
In SQL that means we have to write this:

```
SELECT *
FROM AI_TEST
WHERE data in (‘red’,
‘blue’, ‘yellow’,
‘green’, ‘orange’,
‘gold’, ‘white’);
```

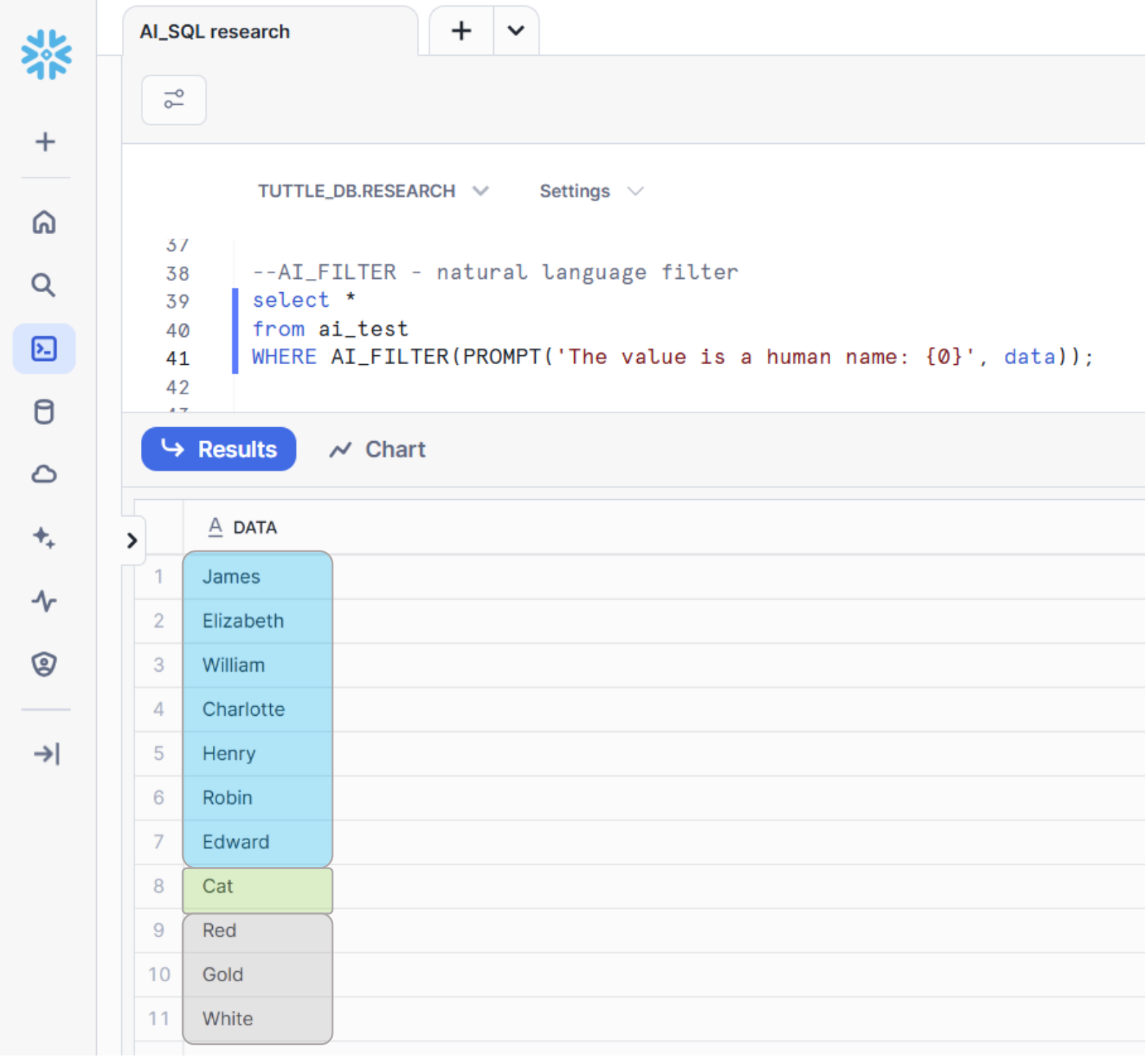




Some prompts were clear enough for the AI to get it right



Sometimes it gets it sort of right



The only way to steer the AI to the answer I wanted for really long prompts

Step 1

Understand What AI Actually Is

AI in not New. *Generative* AI is.

Predictive AI

Forecasts an outcome

- Answers: what is likely to happen?
- Use Case: Demand, fraud, credit risk
- Output: an estimated probability, a category, or a numerical prediction

Generative AI

Predicts the next token

- Answers: what plausibly comes next?
- Text, code, images - chunk by chunk
- Output: fluent, confident content that mimics patterns (average) of truth, not verified truth

Generative AI's output looks like an answer, but it’s really a guess.

How AI Builds an Answer, One Prediction at a Time

How LLMs actually works

- Given a sequence of words, predict the most likely next word
- Learns statistical patterns across training data repeated billions of times
- The result looks like understanding but this is sophisticated pattern matching
- They don’t know things. They complete patterns from training data

“Learning” is not “Understanding.”

Let’s all play a game

Predict the next likely word in the sentence

- The customer requested a _____
- The customer requested a refund because the product _____
- The customer requested a refund because the product arrived _____

Step 1: Get a bunch of text

Original document

The customer requested a refund because the product arrived **damaged.**

The sentence is both an example and its own answer key.

Step 2: The next word stays hidden

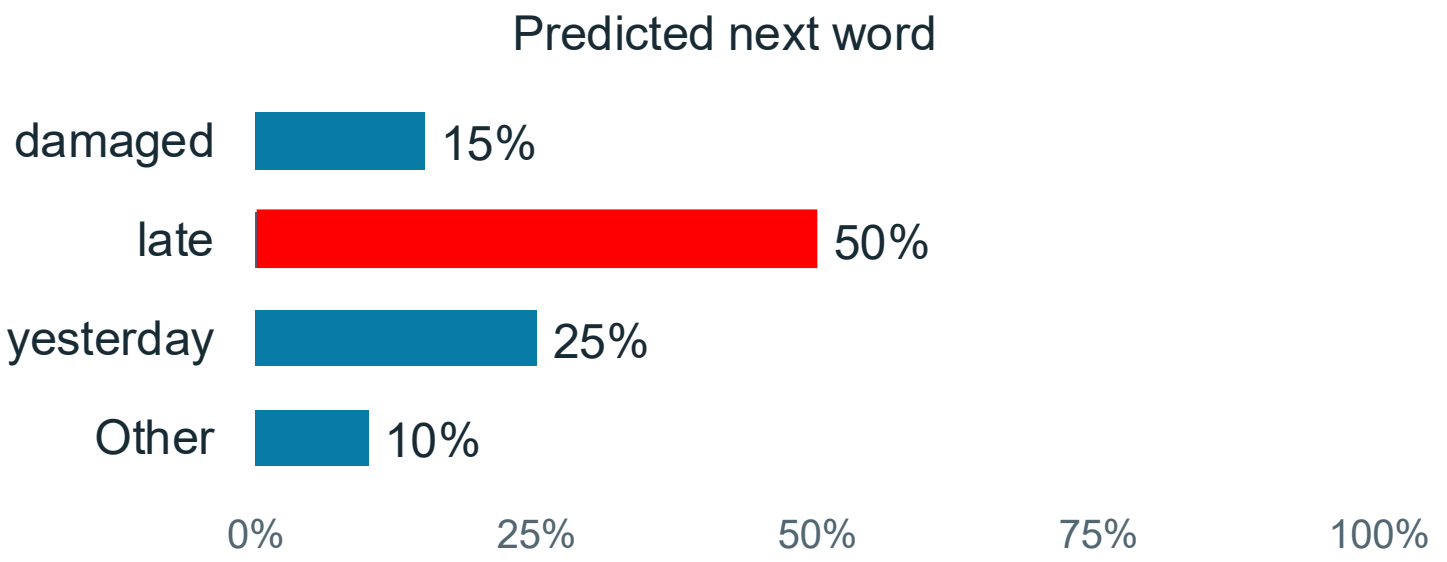
Text the model sees

The customer requested a refund because the product arrived _____.

The model must predict what came next.

Step 3: The model predicts probabilities

The customer requested a refund because the product arrived _____.



Highest prediction

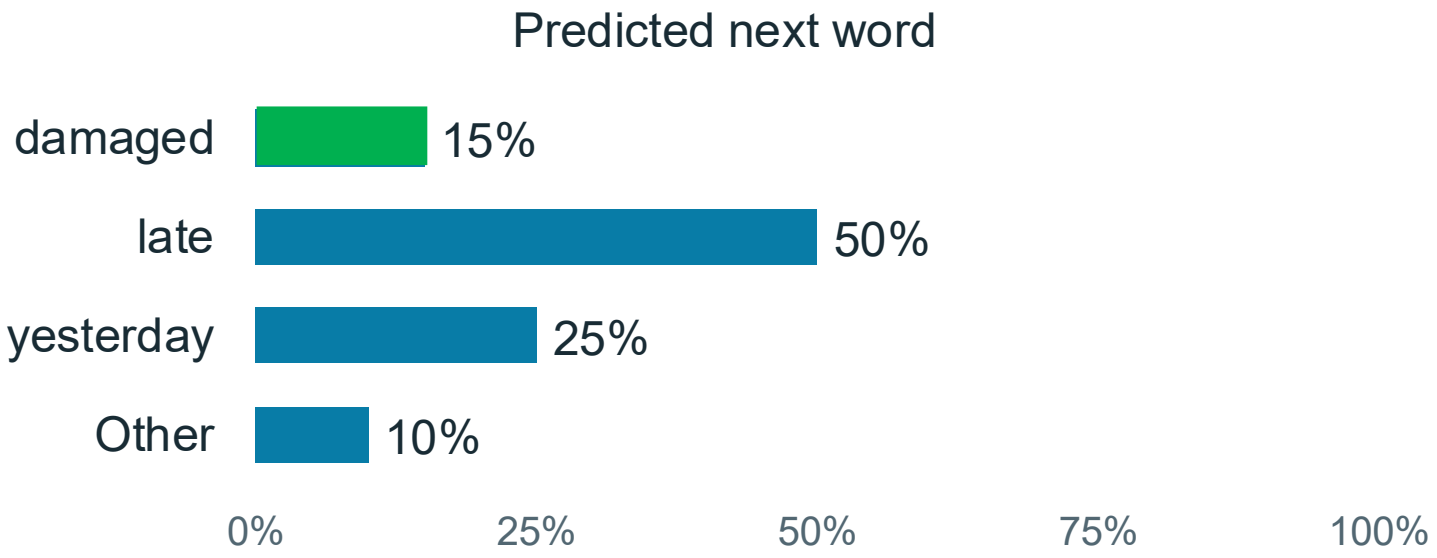
late

50% in this illustration

Illustrative probabilities for the next **token** (i.e. a word, part of a word, or punctuation).

Step 4: The text supplies the answer

The customer requested a refund because the product arrived **damaged** .



Illustrative probabilities for the next **token** (i.e. a word, part of a word, or punctuation).

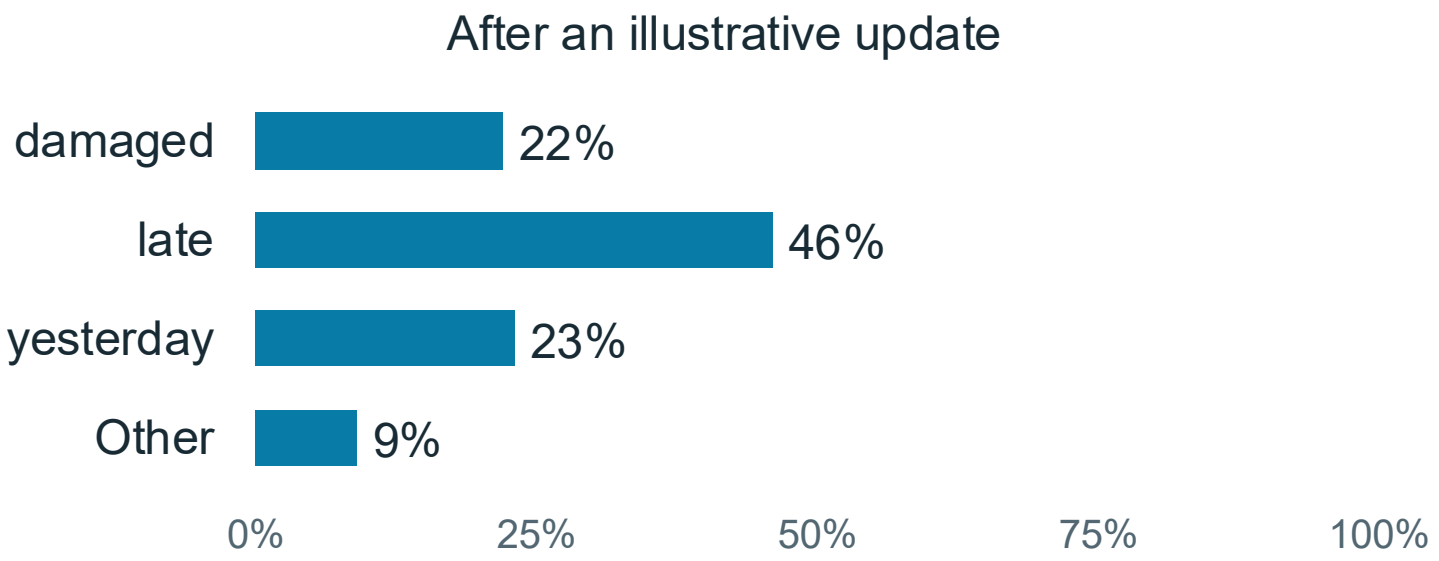
Original text word

damaged

The model assigned it only 15%.

Step 5: An update changes the predictions

The customer requested a refund because the product arrived _____.



Illustrative probabilities for the next **token** (i.e. a word, part of a word, or punctuation).

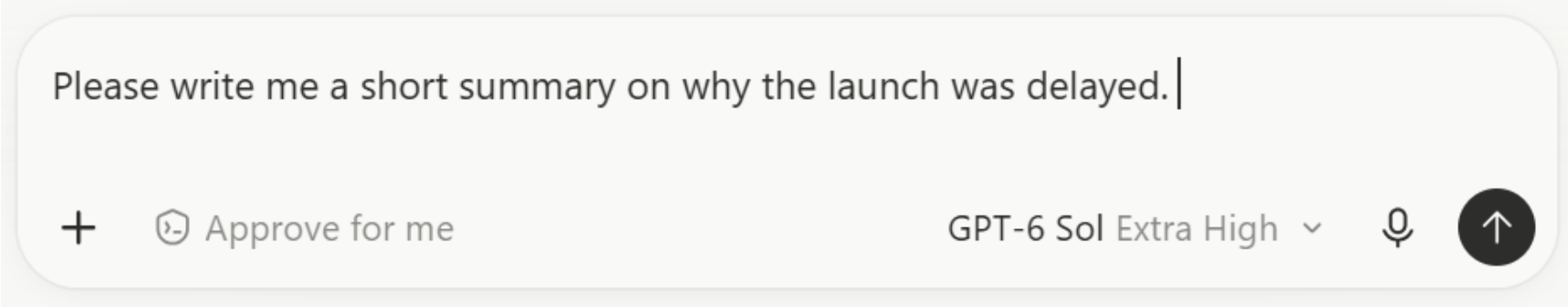
Probability of “damaged”

22%

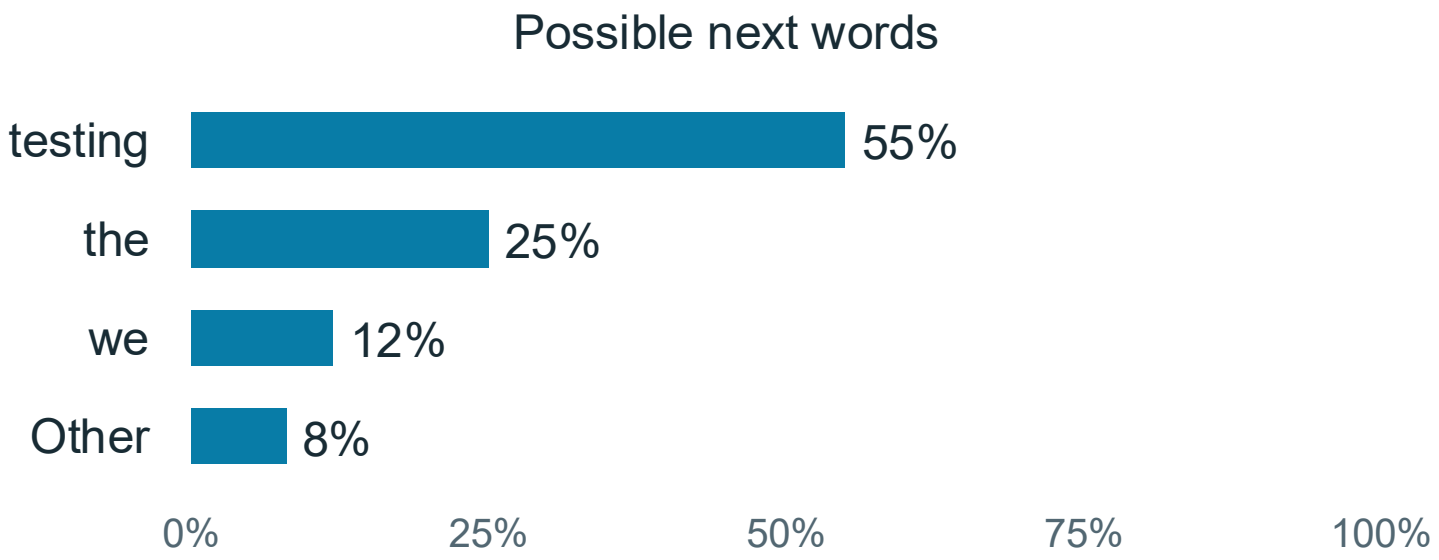
Previously 15%

The model’s weights change.

Generating: one prediction at a time



The launch is delayed because _____



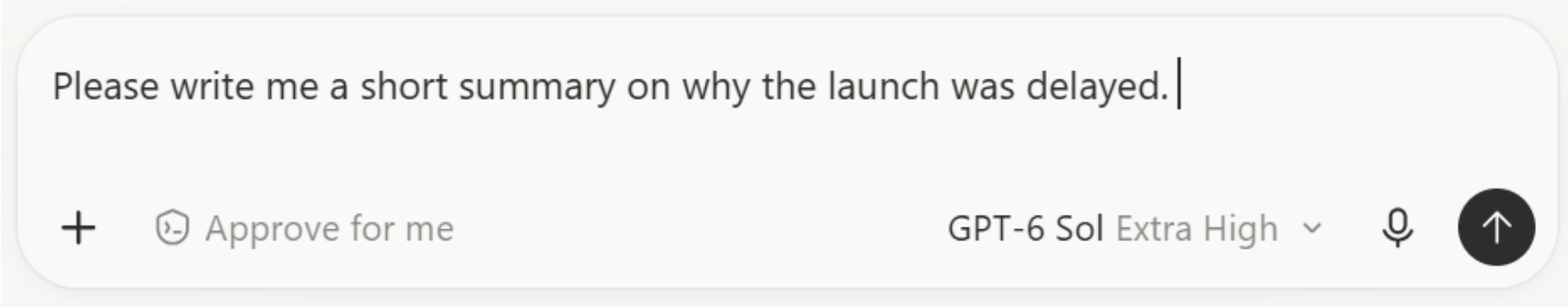
Illustrative probabilities. Tokens can be words or word pieces.

Next addition in this example

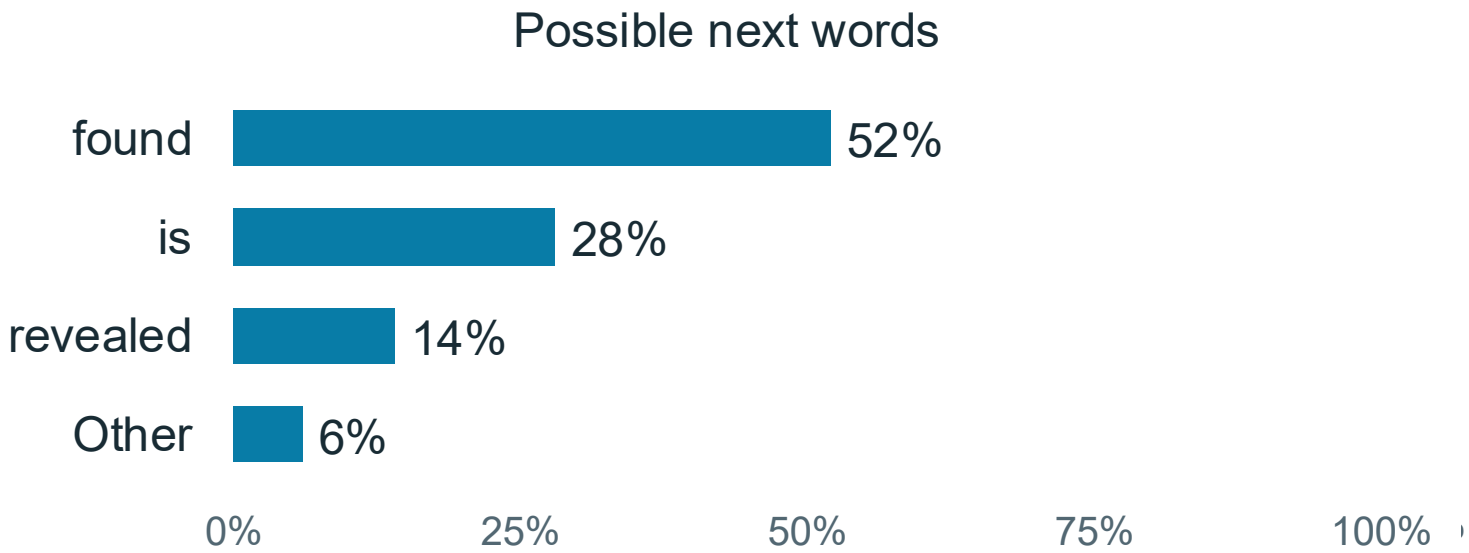
testing

The model uses learned patterns.

Generating: one prediction at a time



The launch is delayed because testing _____



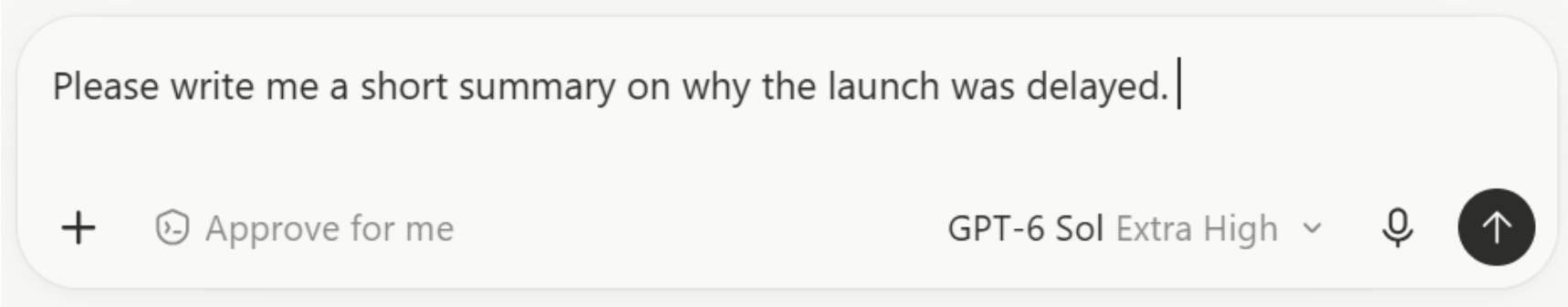
Illustrative probabilities. Tokens can be words or word pieces.

Next addition in this example

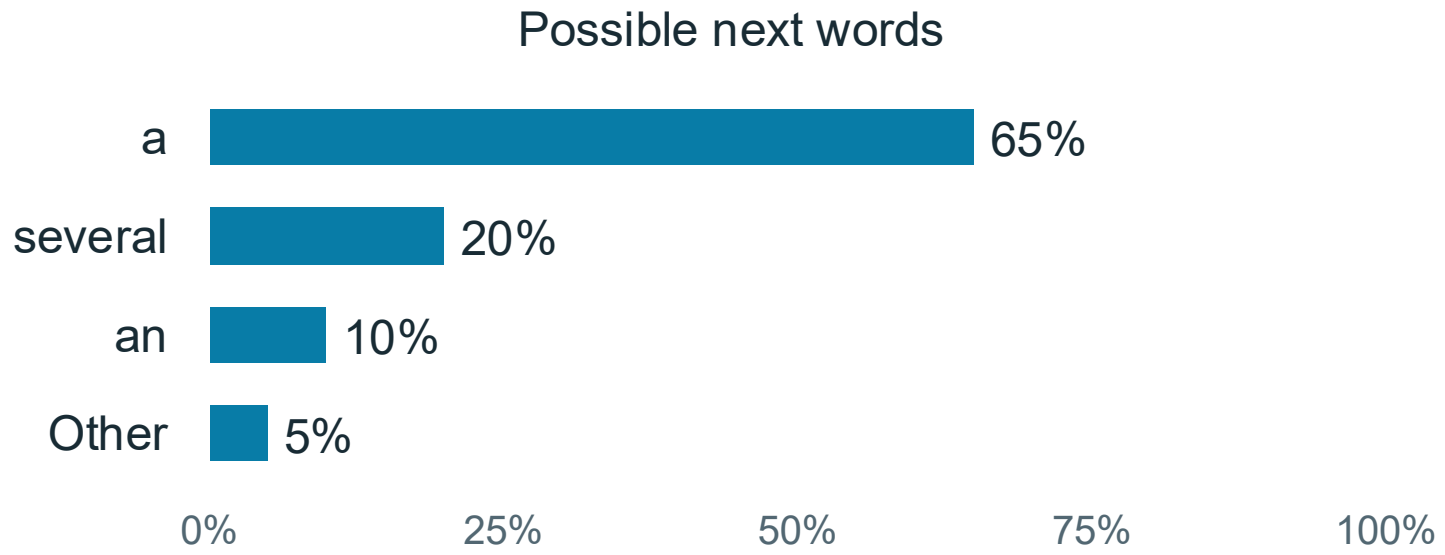
found

The response grows, and the context changes.

Generating: one prediction at a time



The launch is delayed because testing found _____



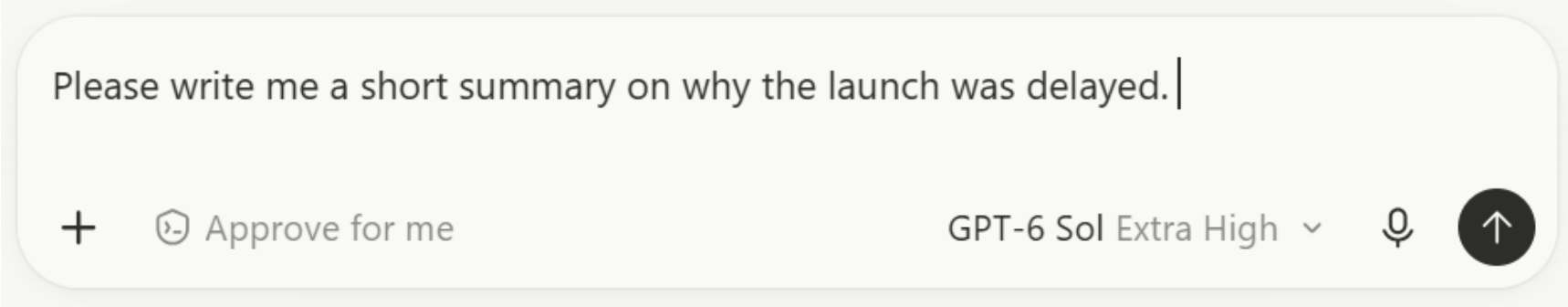
Illustrative probabilities. Tokens can be words or word pieces.

Next addition in this example

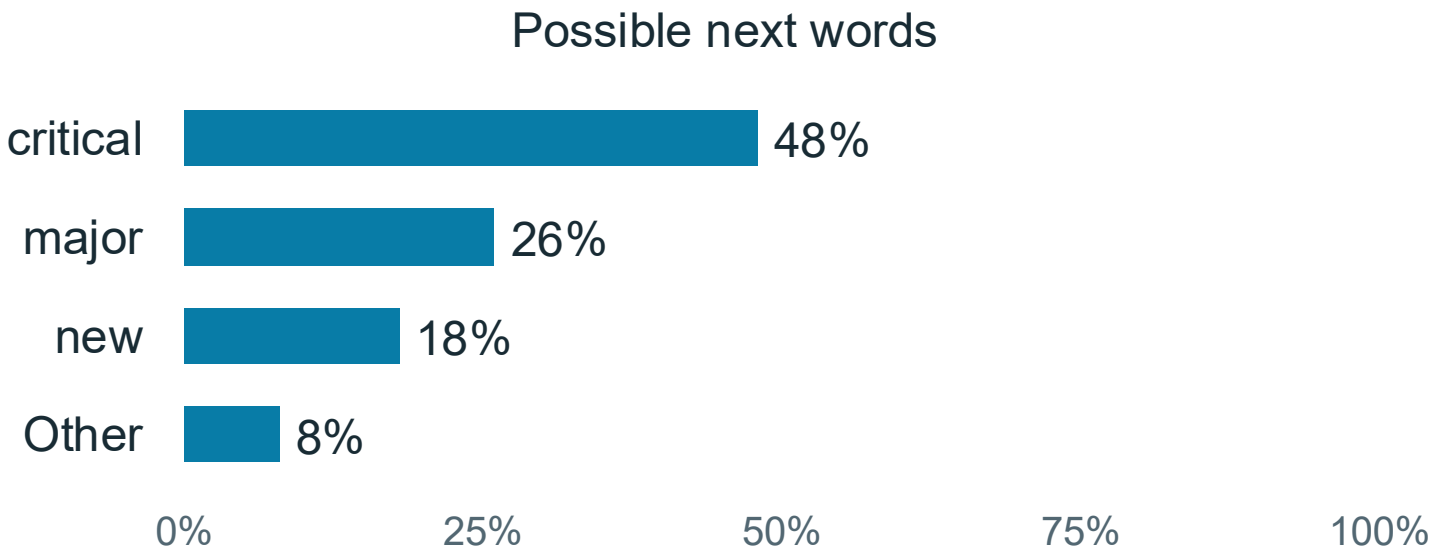
a

The response grows, and the context changes.

Generating: one prediction at a time



The launch is delayed because testing found a _____



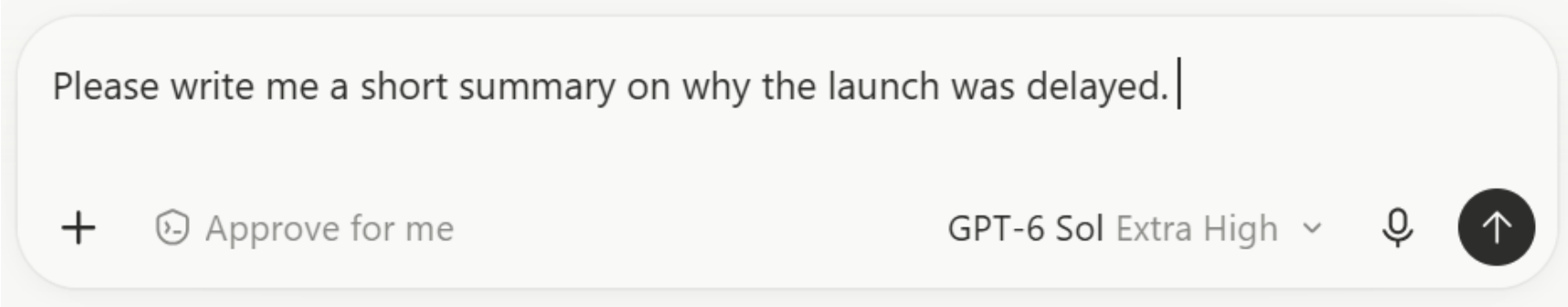
Illustrative probabilities. Tokens can be words or word pieces.

Next addition in this example

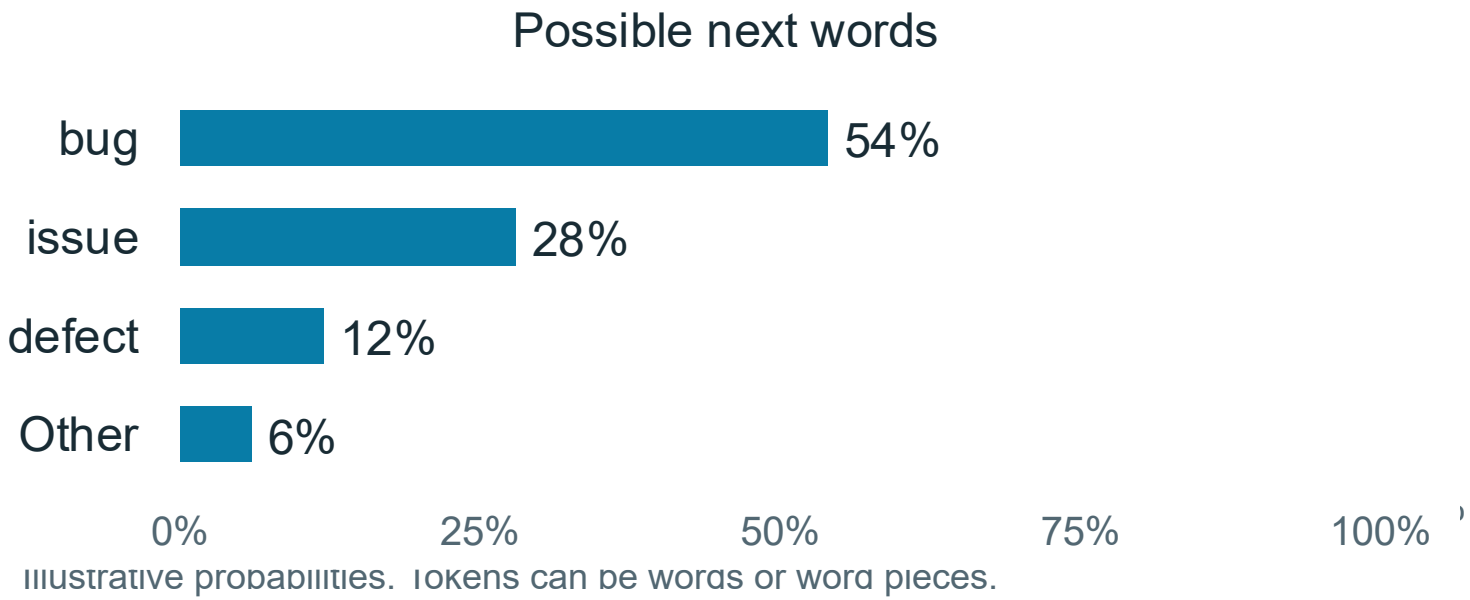
critical

The response grows, and the context changes.

Generating: one prediction at a time



The launch is delayed because testing found a critical _____



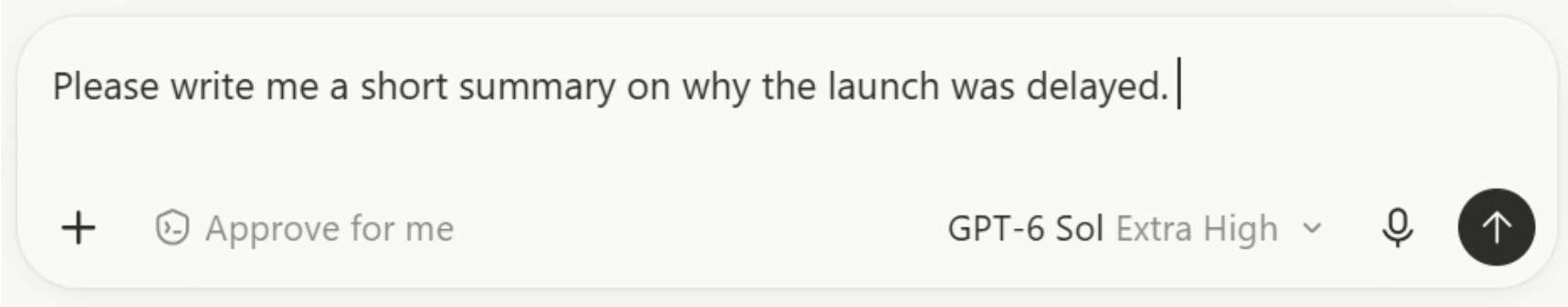
Illustrative probabilities. Tokens can be words or word pieces.

Next addition in this example

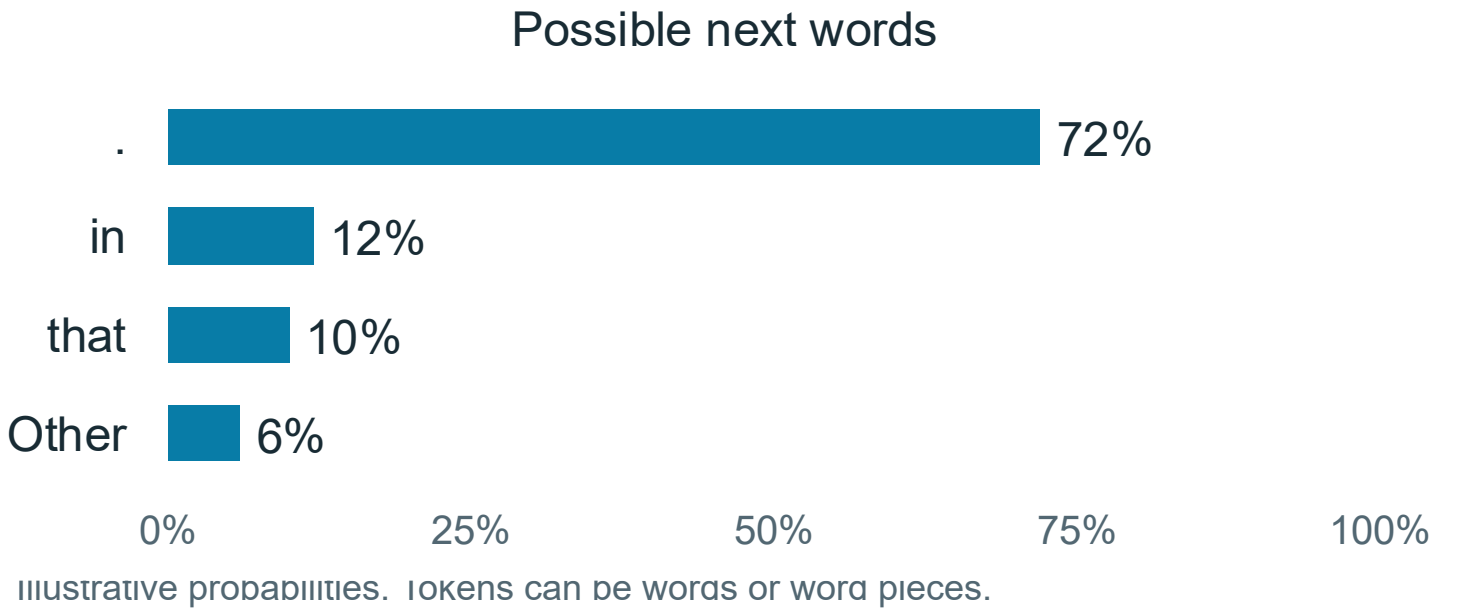
bug

The response grows, and the context changes.

Generating: one prediction at a time



The launch is delayed because testing found a critical _____



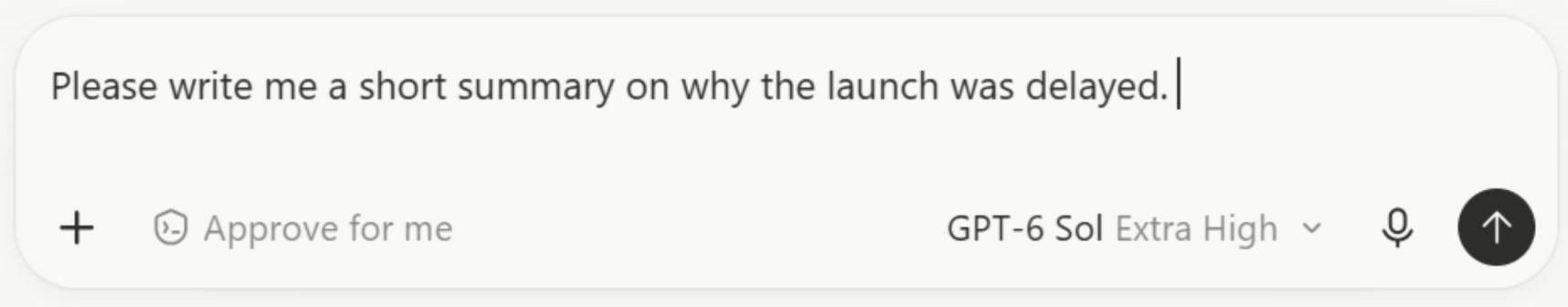
Illustrative probabilities. Tokens can be words or word pieces.

Next addition in this example

period (.)

The response grows, and the context changes.

A plausible update still needs verification



The launch is delayed because testing found a critical bug.

NOTE: We gave it no report or context, yet it confidently answered with little accuracy

Key takeaways:

1. Context and grounding prompts in data matters.
2. You can’t blindly trust outputs. We must verify.

How AI Builds an Answer, One Prediction at a Time

How LLMs actually works

- Given a sequence of words, predict the most likely next word
- Repeat billions of times across training data to learn statistical patterns
- The result looks like understanding but this is sophisticated pattern matching
- They don’t know things. They complete patterns from training data

What all this means

1. They don't know things
2. They complete patterns from training data
3. This is why they can be confidently wrong (hallucinations)
4. This is why AI judgement is more important than fluency

So why does AI think “parrot” is a color?

- Green is the most common parrot color, though most species show some red or other accent colors — cockatoos being the exception, typically appearing in black or white with red, pink, or yellow highlights.
- Parrots are unique among birds in producing their own color pigments, called psittacofulvins, which generate their yellow and red hues entirely from within.
- Parrots don't actually make green pigment — their green feathers result from a combination of yellow psittacofulvin and a light-refracting blue nanostructure in the feather.
- The dusky lory of Papua New Guinea displays either red or yellow plumage, and yellow is genetically dominant over red in this species.
- Melanin pigments produce the darker tones in parrot plumage — browns, blacks, and grays — and also strengthen the feathers against physical wear.
- No bird produces true blue pigment; the blue seen in parrots is entirely structural, produced by microscopic feather architecture that refracts light.
- Red psittacofulvin-pigmented feathers resist bacterial degradation as effectively as melanin-pigmented ones, suggesting parrot colors serve a protective purpose beyond mere appearance.
- Larger parrot species tend to be more colorful, likely because lower predation pressure lets them be conspicuous, and vivid plumage aids mutual mate selection in long-term pair bonds.
- A single enzyme determines whether a parrot's feathers turn red or yellow — making parrots the only birds known to produce bright colors through this mechanism.
- Pink is rare in parrots, found as a primary color in only two cockatoo species, while purple is rarer still — no species is primarily purple, with only a few showing any trace of it.

Colors are in proximity to “parrot” within training data

- Green is the most common parrot color, though most species show some red or other accent colors — cockatoos being the exception, typically appearing in black or white with red, pink, or yellow highlights.
- Parrots are unique among birds in producing their own color pigments, called psittacofulvins, which generate their yellow and red hues entirely from within.
- Parrots don't actually make green pigment — their green feathers result from a combination of yellow psittacofulvin and a light-refracting blue nanostructure in the feather.
- The dusky lory of Papua New Guinea displays either red or yellow plumage, and yellow is genetically dominant over red in this species.
- Melanin pigments produce the darker tones in parrot plumage — browns, blacks, and grays — and also strengthen the feathers against physical wear.
- No bird produces true blue pigment; the blue seen in parrots is entirely structural, produced by microscopic feather architecture that refracts light.
- Red psittacofulvin-pigmented feathers resist bacterial degradation as effectively as melanin-pigmented ones, suggesting parrot colors serve a protective purpose beyond mere appearance.
- Larger parrot species tend to be more colorful, likely because lower predation pressure lets them be conspicuous, and vivid plumage aids mutual mate selection in long-term pair bonds.
- A single enzyme determines whether a parrot's feathers turn red or yellow — making parrots the only birds known to produce bright colors through this mechanism.
- Pink is rare in parrots, found as a primary color in only two cockatoo species, while purple is rarer still — no species is primarily purple, with only a few showing any trace of it.

TRUST BUT VERIFY

“[These systems] have no sense of the limits of their own knowledge or abilities — they won’t tell you when they’re off the map.”

Michael Wooldridge — Oxford

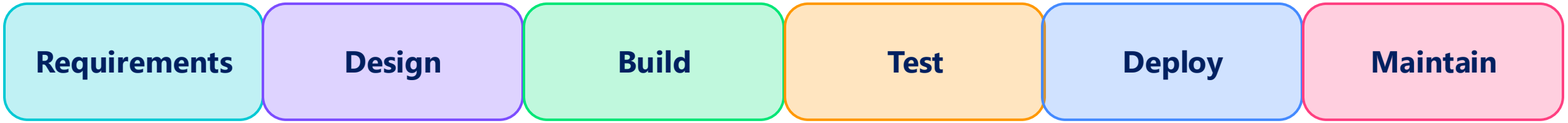
first computer scientist to win the Royal Society’s science communication prize

- *This explains why AI thought parrot was also a color.*
- *Verification isn't optional → The model can't flag its own mistake for you.*
- *You can recreate this experiment by asking for similarity scores and weights*

Step 2

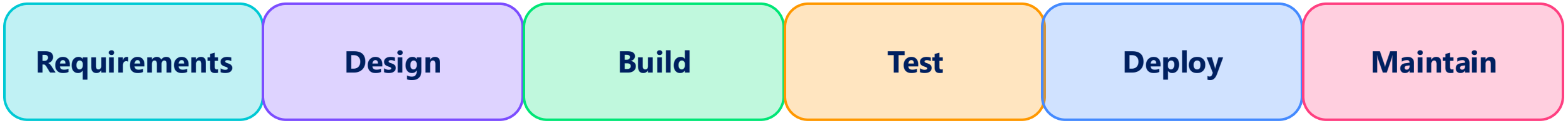
Understand AI’s place in the DLC and it’s impact

A typical software development lifecycle



Many shifts have centered on compressing parts of the lifecycle

A typical software development lifecycle



Many shifts have centered on compressing parts of the lifecycle

- The Internet

A typical software development lifecycle



Many shifts have centered on compressing parts of the lifecycle

- The Internet

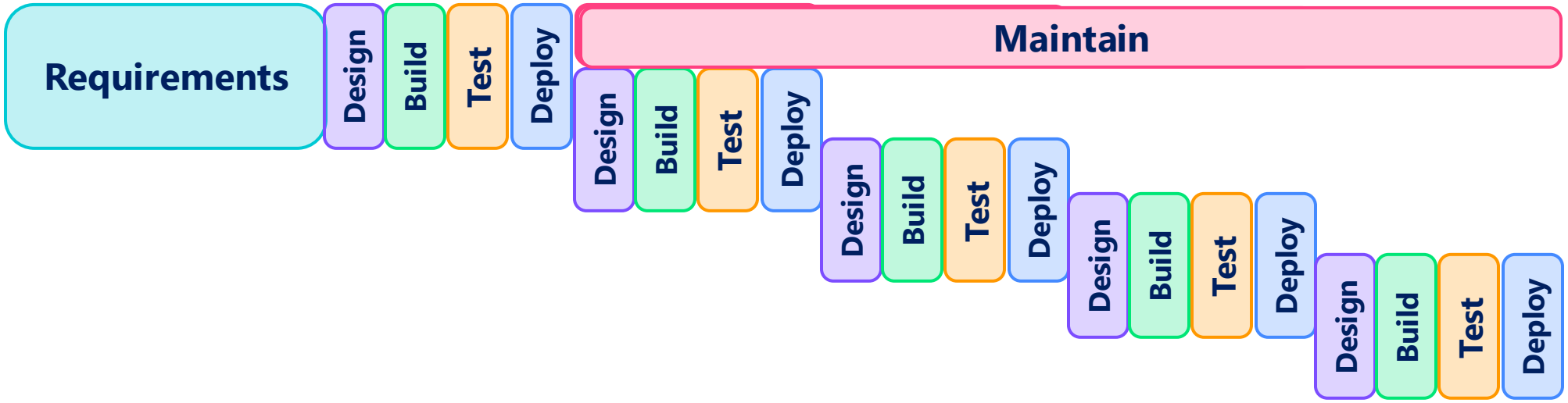
A typical software development lifecycle



Many shifts have centered on compressing parts of the lifecycle

- The Internet
- Agile

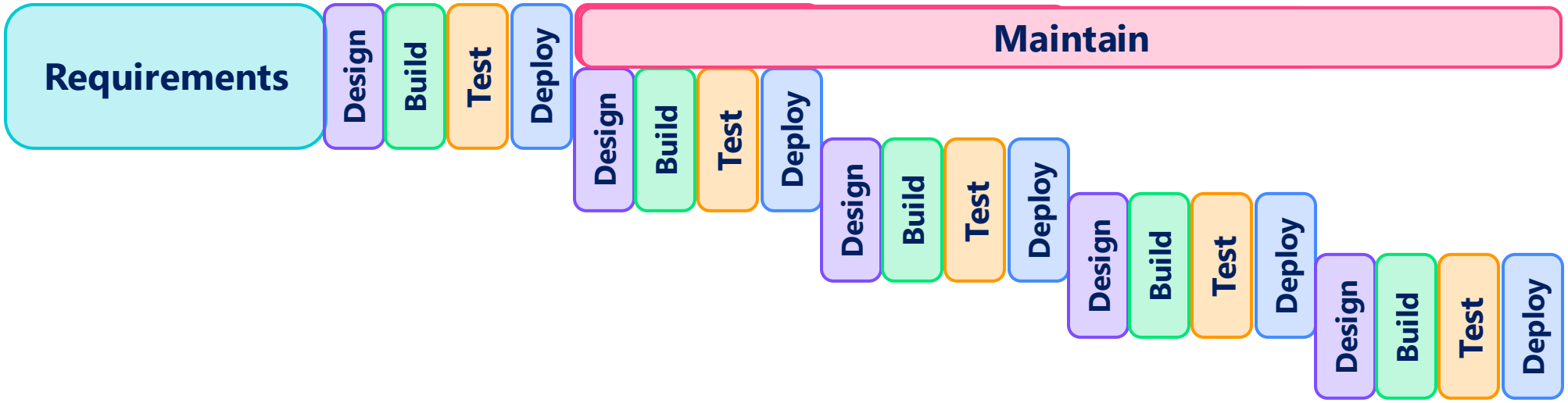
A typical software development lifecycle



Many shifts have centered on compressing parts of the lifecycle

- The Internet
- Agile

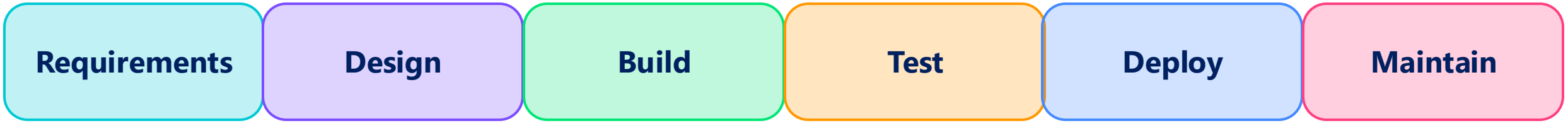
A typical software development lifecycle



Many shifts have centered on compressing parts of the lifecycle

- The Internet
- Agile
- Gen AI

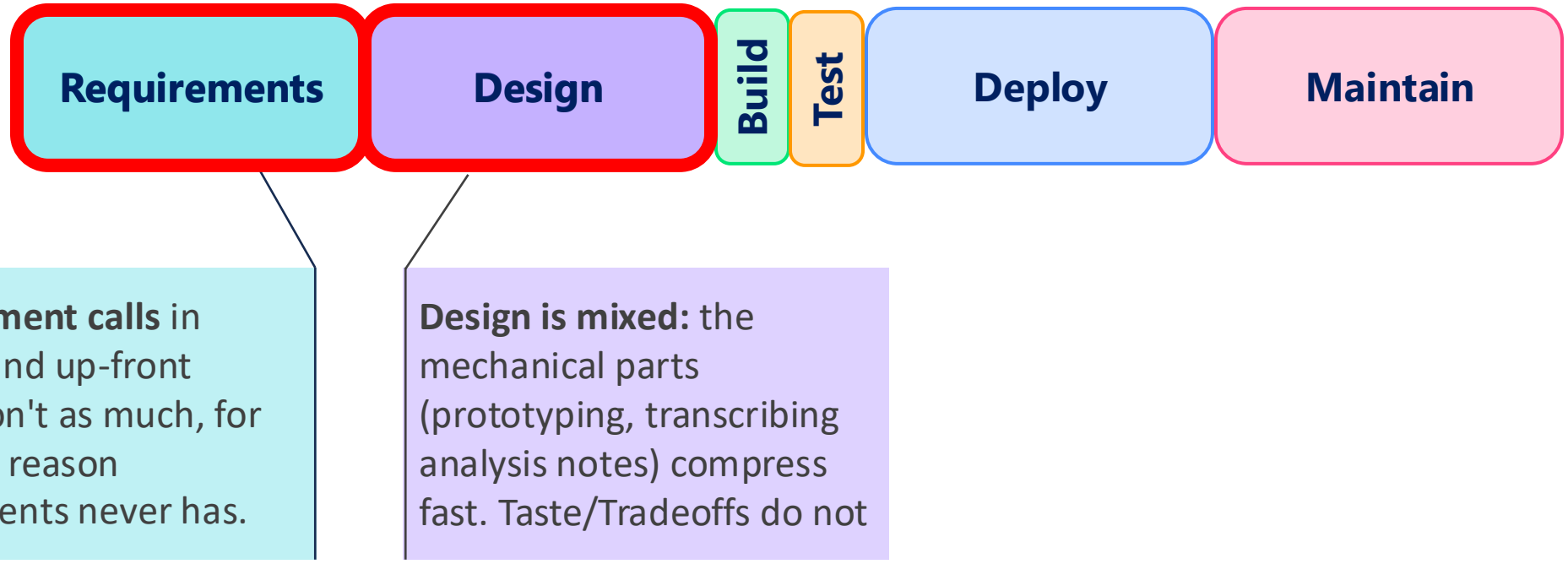
A typical software development lifecycle



Many shifts have centered on compressing parts of the lifecycle

- The Internet
- Agile
- Gen AI

A typical software development lifecycle



Many shifts have centered on compressing parts of the lifecycle

- The Internet
- Agile
- Gen AI

Step 03

Where does AI responsibly fit into work?

It doesn’t replace roles. It replaces tasks. Roles are a collection of tasks/judgements

THE FRAMEWORK

Scoring a Task: AI, Shared, or Human?

Score each task 1–5 on every row. Score tasks, not whole jobs.

Group	Dimension	Question	Score
Can AI do it?	Digital Input + Data readiness	Is the input digital, accessible, and trustworthy?	
	Patterned-based work	Does it follow repeatable patterns or standard workflow?	
	Informational output	Is the output text, data, visuals, or code?	
	Volume/Scale limits for humans	Would doing this at the needed volume take a crowd of people?	
Can we trust it?	Verifiability	Is there a fast, reliable way to check it's correct — tests, ground truth, documented rules, a cited source?	
	Stakes & reversibility	If it's wrong, how costly is that, and how hard is it to undo?	
Must a human own it?	Physical presence	Does it require being there, or physical dexterity?	
	Interpersonal trust	Does it need negotiation, empathy, or a relationship?	
	Novelty	Is it unusual or ambiguous — off the map of past examples?	
	Accountability	Does a named person need to own the outcome, or does regulation require a human decision?	

Don't add the scores up. *A strong "can AI" score can't cancel out a trust or accountability problem — the decision rules decide.*

THE DECISION RULES

From Scores to Ownership

Apply in order. The first rule that matches wins.

1. **Low verifiability + high stakes**
   The veto — holds even if everything else says AI.
   → **Human-only**

2. **Any "must a human" dimension scores high**
   Presence, trust, novelty, or accountability.
   → **At least Shared**

3. **High "can AI" + high verifiability + low stakes**
   Still spot-checked; still someone's accountability.
   → **AI-only candidate**

4. **Everything else**
   AI proposes, a human confirms.
   → **Shared**

Human-owned vs Human & AI shared vs AI-owned

Requirements Tasks — Pre-AI

All 9 tasks performed by humans before AI

#	Task
1	Elicit needs via interviews, workshops, and surveys
2	Analyze and prioritize requirements
3	Write user stories, use cases, or requirement specs
4	Define acceptance criteria
5	Identify constraints (budget, timeline, regulatory, technical)
6	Negotiate scope and resolve conflicting stakeholder needs
7	Create a requirements traceability matrix
8	Conduct feasibility analysis
9	Review and sign off on requirements documents

Human-owned vs Human & AI shared vs AI-owned

Requirements Tasks — Post-AI

All 9 tasks performed by humans before AI

#	Task	Human	Both	AI
1	Elicit needs via interviews, workshops, and surveys	✓	—	—
2	Analyze and prioritize requirements	—	✓	—
3	Write user stories, use cases, or requirement specs	—	✓	—
4	Define acceptance criteria	—	✓	—
5	Identify constraints (budget, timeline, regulatory, technical)	✓	—	—
6	Negotiate scope and resolve conflicting stakeholder needs	✓	—	—
7	Create a requirements traceability matrix	—	—	✓
8	Conduct feasibility analysis	—	✓	—
9	Review and sign off on requirements documents	✓	—	—

Design — Pre-AI Tasks

All 10 tasks performed by humans before AI

#	Task
1	Define system architecture
2	Select technology stack and frameworks
3	Create high-level design documents and diagrams
4	Design data models, schemas, and database structures
5	Design APIs and integration contracts
6	Create UI/UX wireframes, mockups, and prototypes
7	Define security architecture and threat models
8	Plan for scalability, performance, and fault tolerance
9	Establish coding standards, naming conventions, and patterns
10	Conduct design reviews and walkthroughs

Design — Post-AI Classification

Human: 1 · Both: 9 · AI: 0

#	Task	Human	Both	AI
1	Define system architecture	—	✓	—
2	Select technology stack and frameworks	—	✓	—
3	Create high-level design documents and diagrams	—	✓	—
4	Design data models, schemas, and database structures	—	✓	—
5	Design APIs and integration contracts	—	✓	—
6	Create UI/UX wireframes, mockups, and prototypes	—	✓	—
7	Define security architecture and threat models	—	✓	—
8	Plan for scalability, performance, and fault tolerance	—	✓	—
9	Establish coding standards, naming conventions, and patterns	—	✓	—
10	Conduct design reviews and walkthroughs	✓	—	—

Build — Pre-AI Tasks

All 10 tasks performed by humans before AI

#	Task
1	Set up development environments and toolchains
2	Write application source code
3	Implement database scripts and migrations
4	Write unit tests alongside production code
5	Integrate third-party libraries, SDKs, and services
6	Conduct code reviews and PR approvals
7	Refactor code for readability and maintainability
8	Write inline documentation and code comments
9	Manage branches, merges, and resolve conflicts
10	Build and configure CI pipelines

Build — Post-AI Classification

Human: 0 · Both: 5 · AI: 5

#	Task	Human	Both	AI
1	Set up development environments and toolchains	—	—	✓
2	Write application source code	—	✓	—
3	Implement database scripts and migrations	—	—	✓
4	Write unit tests alongside production code	—	—	✓
5	Integrate third-party libraries, SDKs, and services	—	✓	—
6	Conduct code reviews and PR approvals	—	✓	—
7	Refactor code for readability and maintainability	—	✓	—
8	Write inline documentation and code comments	—	—	✓
9	Manage branches, merges, and resolve conflicts	—	✓	—
10	Build and configure CI pipelines	—	—	✓

Test — Pre-AI Tasks

All 11 tasks performed by humans before AI

#	Task
1	Write and maintain test plans and test cases
2	Perform functional testing against acceptance criteria
3	Execute integration and end-to-end testing
4	Conduct regression testing after changes
5	Perform exploratory and ad-hoc testing
6	Run performance, load, and stress tests
7	Conduct security and penetration testing
8	Test for accessibility and cross-browser compatibility
9	Log, triage, and prioritize defects
10	Verify bug fixes and conduct retesting
11	Perform user acceptance testing (UAT)

Test — Post-AI Classification

Human: 2 · Both: 4 · AI: 5

#	Task	Human	Both	AI
1	Write and maintain test plans and test cases	—	✓	—
2	Perform functional testing against acceptance criteria	—	—	✓
3	Execute integration and end-to-end testing	—	—	✓
4	Conduct regression testing after changes	—	—	✓
5	Perform exploratory and ad-hoc testing	✓	—	—
6	Run performance, load, and stress tests	—	—	✓
7	Conduct security and penetration testing	—	✓	—
8	Test for accessibility and cross-browser compatibility	—	✓	—
9	Log, triage, and prioritize defects	—	✓	—
10	Verify bug fixes and conduct retesting	—	—	✓
11	Perform user acceptance testing (UAT)	✓	—	—

Deploy — Pre-AI Tasks

All 10 tasks performed by humans before AI

#	Task
1	Prepare release notes and deployment documentation
2	Configure target environments (staging, production)
3	Execute database migrations in production
4	Run deployment scripts or orchestrate release pipelines
5	Perform smoke testing and sanity checks post-deploy
6	Manage feature flags and rollout strategies
7	Coordinate release communication with stakeholders
8	Execute rollback procedures if issues detected
9	Update configuration and environment variables
10	Validate monitoring and alerting is active

Deploy — Post-AI Classification

Human: 1 · Both: 3 · AI: 6

#	Task	Human	Both	AI
1	Prepare release notes and deployment documentation	—	—	✓
2	Configure target environments (staging, production)	—	—	✓
3	Execute database migrations in production	—	✓	—
4	Run deployment scripts or orchestrate release pipelines	—	—	✓
5	Perform smoke testing and sanity checks post-deploy	—	—	✓
6	Manage feature flags and rollout strategies	—	✓	—
7	Coordinate release communication with stakeholders	✓	—	—
8	Execute rollback procedures if issues detected	—	—	✓
9	Update configuration and environment variables	—	—	✓
10	Validate monitoring and alerting is active	—	✓	—

Maintain — Pre-AI Tasks

All 11 tasks performed by humans before AI

#	Task
1	Monitor system health, uptime, and performance metrics
2	Triage and prioritize production bug reports
3	Diagnose and resolve production incidents
4	Apply security patches and dependency updates
5	Refactor and optimize existing code for performance
6	Manage technical debt backlog
7	Update documentation to reflect system changes
8	Conduct post-incident reviews and write retrospectives
9	Plan and execute database maintenance and optimization
10	Manage infrastructure scaling and capacity planning
11	Communicate system status and changes to stakeholders

Maintain — Post-AI Classification

Human: 2 · Both: 5 · AI: 4

#	Task	Human	Both	AI
1	Monitor system health, uptime, and performance metrics	—	—	✓
2	Triage and prioritize production bug reports	—	✓	—
3	Diagnose and resolve production incidents	—	✓	—
4	Apply security patches and dependency updates	—	—	✓
5	Refactor and optimize existing code for performance	—	✓	—
6	Manage technical debt backlog	✓	—	—
7	Update documentation to reflect system changes	—	—	✓
8	Conduct post-incident reviews and write retrospectives	—	✓	—
9	Plan and execute database maintenance and optimization	—	✓	—
10	Manage infrastructure scaling and capacity planning	—	—	✓
11	Communicate system status and changes to stakeholders	✓	—	—

Human-owned vs Human & AI shared vs AI-owned

Phase	Human	Both	AI	Total
Requirements	4	4	1	9
Design	1	9	0	10
Build	0	5	5	10
Test	2	4	5	11
Deploy	1	3	6	10
Maintain	2	5	4	11

% human only	% human involved	% AI only task
44%	89%	11%
10%	100%	0%
0%	50%	50%
18%	55%	45%
10%	40%	60%
18%	64%	36%

Human Both AI

REAL EXAMPLE — ANALYSIS, INVERTED

The Prototype Came Before the Requirements Doc

John

Business SME — owned why

- Built a working prototype at small, local scale
- Proved the concept was worth pursuing
- Owned the product vision from day one

REAL EXAMPLE — ANALYSIS, INVERTED

The Prototype Came Before the Requirements Doc

John

Business SME — owned why

- Built a working prototype at small, local scale
- Proved the concept was worth pursuing
- Owned the product vision from day one

Clint

Technical Engineer — owned how & scale

- Architected and deployed on Azure — AI Search, managed identity, full pipeline
- Infrastructure and access John didn't have or know
- Nothing had to be walked back on requirements once it scaled

Live in production today. Current work: tightening the data pipelines to bring in more faculty work.

Screenshot of final tool

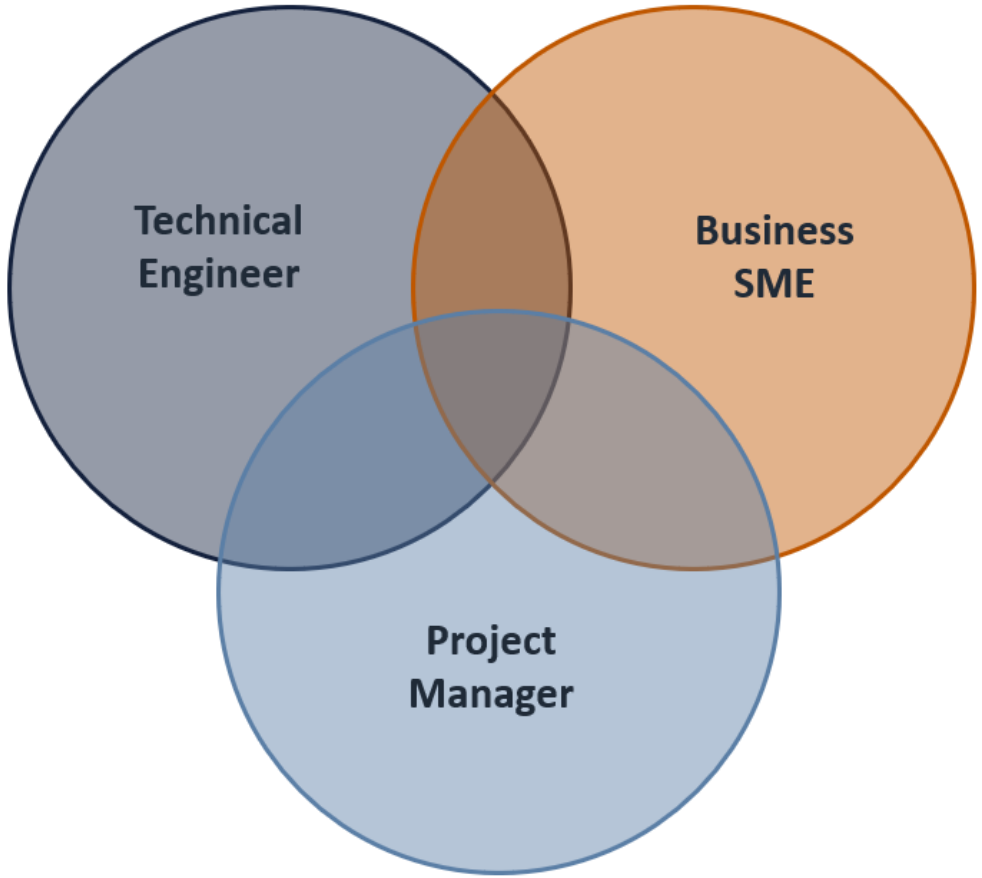
WARNING: A Fast Prototype Can Also Anchor the Wrong Answer

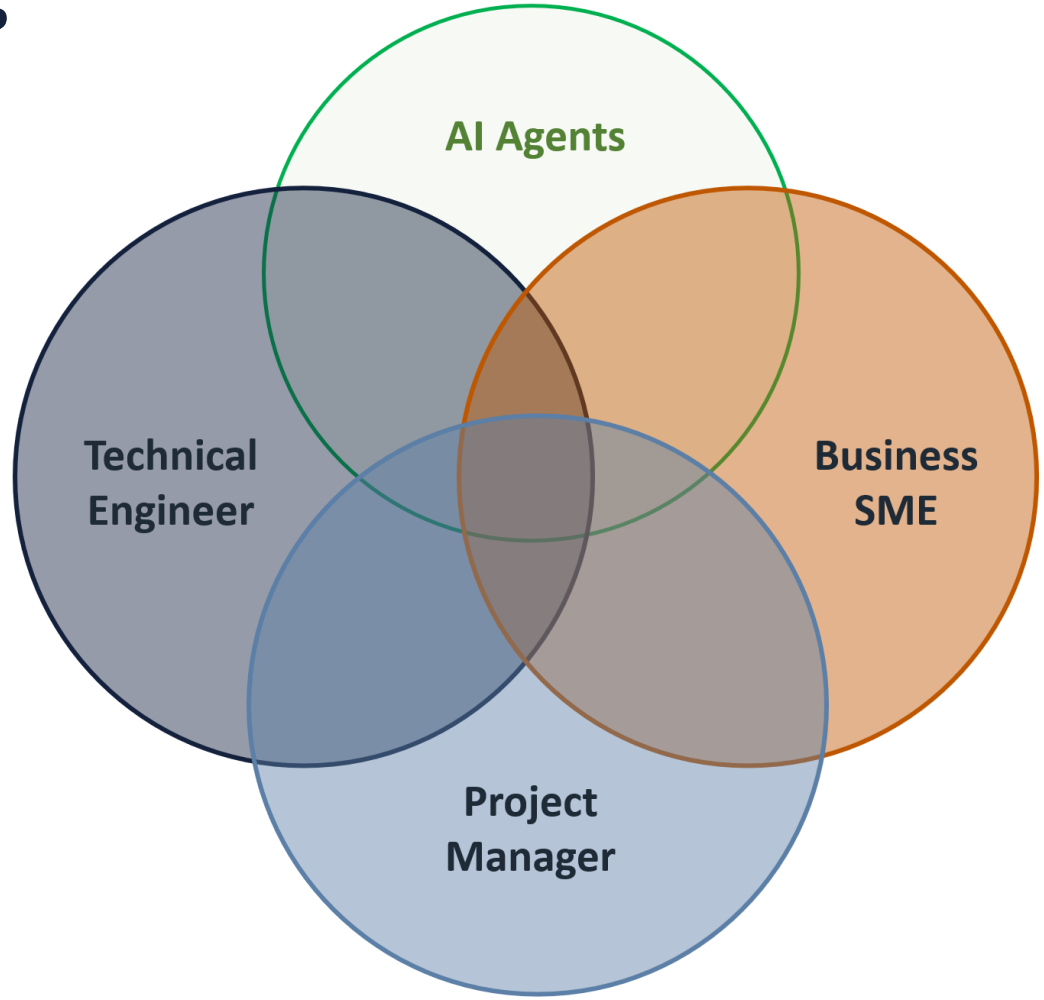
Andrew Ambrosino — OpenAI, leads the Codex desktop app

A working prototype can answer questions nobody asked yet — and because it runs, it can look settled. The conversation quietly shifts from "is this the right problem" to "does this work correctly." That second question is easier, so teams drift toward it.

The Big Ideas KB story didn't hit this failure because we always came back to strategy goals and the "why"

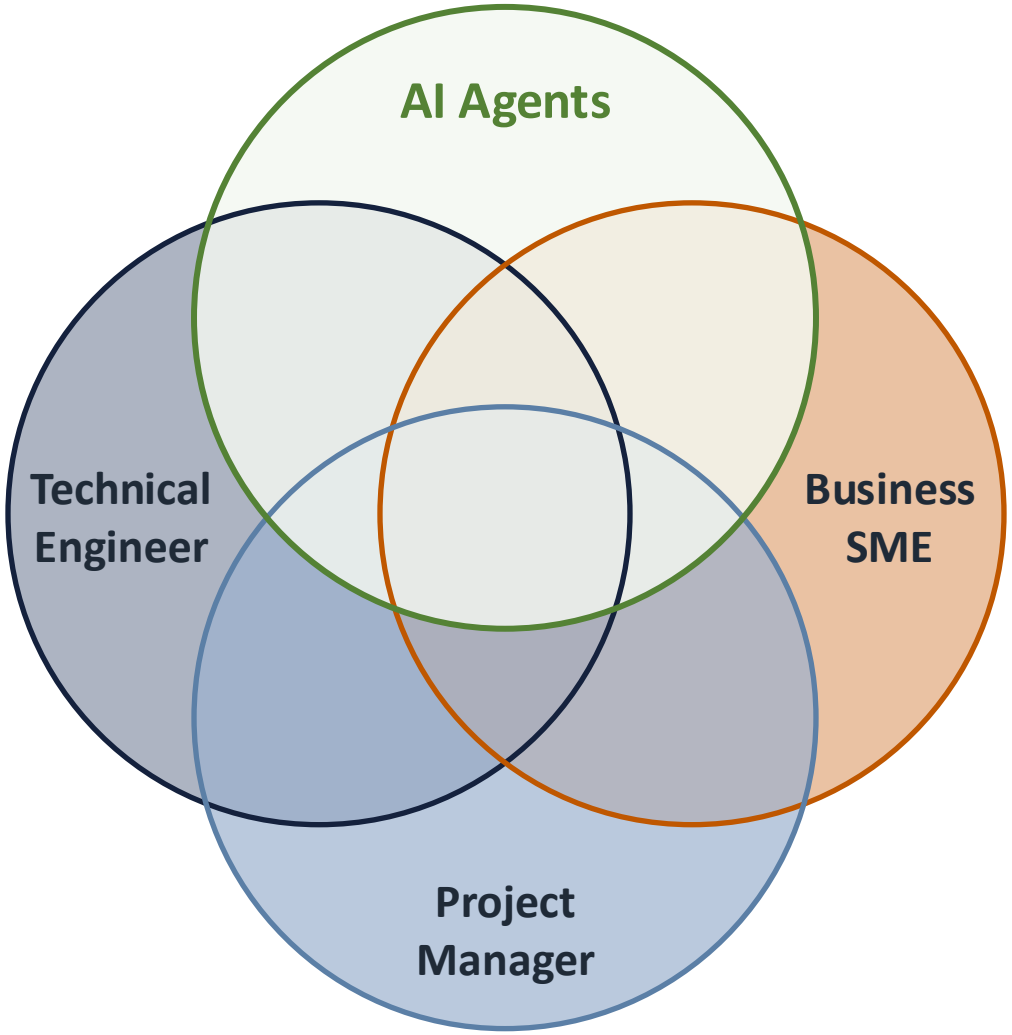
The Team as we knew it is changing





THE TEAM, WITH AI ON IT

What Stays Human in Every Role



Technical Engineer

Architecture, security, and scale — making it work beyond the demo.

Business SME

The why — strategy, domain judgment, and whether this is the right problem.

Project Manager

Direction, alignment, and estimating the work that didn't compress.

AI overlaps part of every role — including yours. ***Your edge is the part it doesn't reach.***

WHO DOES WHAT

Where Each Role's Work Lands

	Technical Engineer	Business SME	Project Manager
Human-only	• Architecture & tradeoffs • Security & identity design • Production sign-off & ownership	• Defining the problem & why • Defining what "good" means • Judging domain exceptions	• Direction — including saying no • Estimating work that didn't compress • Go/no-go & risk acceptance
Shared	• Code: AI drafts, engineer owns • Debugging & test cases • Infra config vs. org policy	• Prototyping (vibe-coding) • Interviews → requirements • Reviewing AI answers for accuracy	• Estimation: AI sizes the build • Risk register: PM rates stakes • Evaluating what a prototype decided
AI-only	• Boilerplate & scaffolding • Lint fixes & dependency bumps • Format conversion	• Transcribing & summarizing • Tagging & routing questions • Reformatting content	• Status reports & dashboards • Meeting notes & action items • Scheduling & reminders

*AI does a lot of the same things for each role, **while they remain distinct at the human-only level***

LIVE EXERCISE

You Decide First. Then We Compare.

Hands up per task — AI-only, shared, or human-only — before the reveal

1. Student submits a waiver request + documents
2. Check the request against eligibility criteria & records
3. Evaluate a borderline or unusual case
4. Make and sign the approve / deny decision

Hands up, tally, reveal, move on — disagreement here is the point, not a problem.

THE REVEAL

The Waiver Process, Classified

Submit request + documents	AI-only
Check against eligibility & records	Shared
Evaluate a borderline case	Human-only
Approve / deny decision	Human-only

Before you automate anything, ask:

- Is it strategically aligned?
- Has the risk been assessed?
- Who maintains it once it's live?

The sealing question: when the AI gets it wrong, who is accountable? Until you can answer that for a step, it isn't ready to automate.

Step 04

How should we implement AI?

Consider not where to put AI but when to use AI

- What you can do is
  - Automate
  - Augment
  - Invent
- Buy vs build
  - MBA Bot as the example — built in Copilot Studio for exactly that reason.
  - **Not everything needs a custom build:** MBA Bot works partly *because* it has guardrails built in and is easier to steer than a bespoke solution would be — an argument for buy-and-configure that isn't just about cost.
- Grounded
- Guardails – Self harm example, privacy (what data I give it)
- **Evaluation -** What to ship or not? Ungrounded
- **Monitoring –** The agent has to be treated like anyone with a role and task ownership. Regular review.

THE DECISION

Buy the Model. Build on Top of It.

Buy

- Fast, cheap to try, best-in-class out of the box
- Commoditized — everyone has access
- Best for general tasks and prototypes

Build

- Deep, hard-to-copy differentiation
- Expensive and slower — talent, compute
- Best for core, defensible use cases

MBA Bot — buy-and-configure, on purpose

- Built in Copilot Studio — a low-code layer that makes it business-maintainable, not engineering-owned
- Comes with guardrails built in and is easier to steer than a bespoke solution — not everything needs a custom build
- The honest part: MTS and UT are still actively figuring out how to govern deployments of user-written applications — genuinely unresolved, not a finished story

LIVE EXERCISE

Three Real Answers. You Decide.

For each one — Ship, Fix First, or Don't Ship? Vote before the reveal.

What's the minimum GMAT score to waive the requirement?

AI

McCombs MBA doesn't set a minimum GMAT score for a waiver. Waivers are evaluated case-by-case based on quantitative coursework, professional experience, and other standardized test scores.

mccombs.utexas.edu/mba/admissions/faq

When is the scholarship application deadline?

AI

The McCombs MBA Merit Scholarship deadline is January 15th, and awards are announced by February 1st.

mccombs.utexas.edu/mba/blog/2023/scholarships_deadlines

Do I need to submit a resume with my application?

AI

Yes — a current resume is required. It should reflect your professional experience and any leadership roles.

— *Sarah T., Admissions*

THE REVEAL

Ship. Fix First. Don’t Ship.

GMAT waiver score

Specific claim, real citation, the link resolves and is current — nothing left to check.

Ship

Scholarship deadline

The citation is real — but dated 2025. Deadlines change year to year; an old link doesn’t verify a current claim.

Don’t Ship

Resume requirement

A name isn’t a verifiable source — there’s nothing to click through and confirm.

Fix First

Ship it, Fix it, or Don’t Ship it?

The checklist:

- ❑ *Is it grounded in a real source?*
- ❑ *Has anyone verified it?*
- ❑ *If it’s wrong, who owns that?*

Takeaways

What Future Project Work Looks Like

With AI on the team — what changes, and what your job becomes.

Key takways

- Business gets more power to MVP before engineers get involved at all.
- You can more power
- Rapid, cheap experimentation
- Be thoughtful about AI placement. Not tech for sake of tech and be careful with users thinking AI is a magic solution
  - Automate – where does it make sense?
  - Augmentation/New – Speed to build/test/change is faster. Humans still matter in decision and design.
- AI is software technically but we have to treat it like a teammate. It can take on a role and tasks. It also needs regular reviews. Careful what you ship especially now that you can build/test/ship faster
  - Verification > authorship

THE SHIFT

Verification Becomes the Job

Verification, not authorship	The core skill moves from producing the answer to checking it — callback to Section 01.
Rapid, cheap experimentation	A prototype takes an afternoon. The bottleneck is knowing what's worth trying.
The artifact hierarchy inverts	Business gets more power to MVP before engineers get involved at all.

Named tension: this only compresses analysis. Approval, procurement, and security review haven't compressed at all — that may be exactly where a PM's value moves to.

FROM ROWING TO STEERING

Fluency Isn’t the Bar. Steering Is.

Her argument: as AI agents take on more execution, work shifts from rowing to steering — people spend their time on direction and judgment, not the mechanics of getting there.

“Writing as thinking is something I never will automate. You shouldn’t replace your thinking with it.”

Tara Seshan — OpenAI, leads product for Codex and ChatGPT Work

Same finding, from a different direction:

KPMG’s field study found AI fluency didn’t predict who added value — steering behavior did. Especially useful framing if you’re newer to a role: fluency isn’t the bar. Steering is.

Thank You

Questions & Discussion

Clint Tuttle · McCombs School of Business, UT Austin