



Build the AI Data Stack

What Every AI Product Team Needs To Get Right

Kamesh Sampath
Lead Developer Advocate | Snowflake

Safe Harbor and Disclaimers

Other than statements of historical fact, all statements contained in these materials and any accompanying oral commentary (collectively, the “Materials”) are forward-looking statements within the meaning of Section 27A of the Securities Act of 1933, as amended, and Section 21E of the Securities Exchange Act of 1934, as amended, including statements regarding (i) Snowflake’s business strategy, plans, opportunities, or priorities; (ii) Snowflake’s new or enhanced products, services, and technology offerings, including those that are under development or not generally available; (iii) market size and growth, trends, and competitive considerations; (iv) Snowflake’s vision, strategy, and expected benefits relating to artificial intelligence (“AI”), Snowpark, Snowflake Marketplace, the AI Data Cloud, and AI Data Clouds for specific industries or product categories, including the expected benefits and network effects of the AI Data Cloud; and (v) the integration, interoperability, and availability of Snowflake’s products, services, or technology offerings with or on third-party platforms or products, including public cloud platforms. These forward-looking statements are subject to a number of known and unknown risks, uncertainties, and assumptions, including those described under the heading “Risk Factors” and elsewhere in the Annual Reports on Form 10-K and the Quarterly Reports on Form 10-Q that Snowflake files with the Securities and Exchange Commission from time to time. In light of these risks, uncertainties, and assumptions, the future events and trends discussed in the Materials may not occur, and actual results could differ materially and adversely from those expressed or implied in the forward-looking statements. As a result, you should not rely on any forwarding-looking statements as predictions of future events. Forward-looking statements speak only as of the date the statements are first made and are based on information available to Snowflake at the time those statements are made and/or management’s good faith belief as of that time. Except as required by law, we undertake no obligation, and do not intend, to update the forward-looking statements in these Materials.

Any future product or roadmap information (collectively, the “Roadmap”) is intended to outline general product direction. The Roadmap is not a commitment, promise, or legal obligation for Snowflake to deliver any future products, features, or functionality; and is not intended to be, and shall not be deemed to be, incorporated into any contract. The actual timing of any product, feature, or functionality that is ultimately made available may be different from what is presented in the Roadmap. The Roadmap information should not be used when making a purchasing decision. In case of conflict between the information contained in the Materials and official Snowflake documentation, official Snowflake documentation should take precedence over these Materials. Further, note that Snowflake has made no determination as to whether separate fees will be charged for any future products, features, and/or functionality which may ultimately be made available. Snowflake may, in its own discretion, choose to charge separate fees for the delivery of any future products, features, and/or functionality which are ultimately made available.

The Materials may contain information provided by third-parties. Snowflake has not independently verified this information, and usage of this information does not mean or imply that Snowflake has adopted this information as its own or independently verified its accuracy.



Kamesh Sampath

Lead Developer Advocate



The Core Profile: Infrastructure as a Person

innovator:

identity: "Veteran Tech Innovator"

experience: "years $\geq$ 20 in the trenches"

specs:

- Author
- Developer Advocate
- Core Infrastructure Architect
- [linkedin.com/in/kameshsampath](https://www.linkedin.com/in/kameshsampath)

Runtime Parameters

mission:

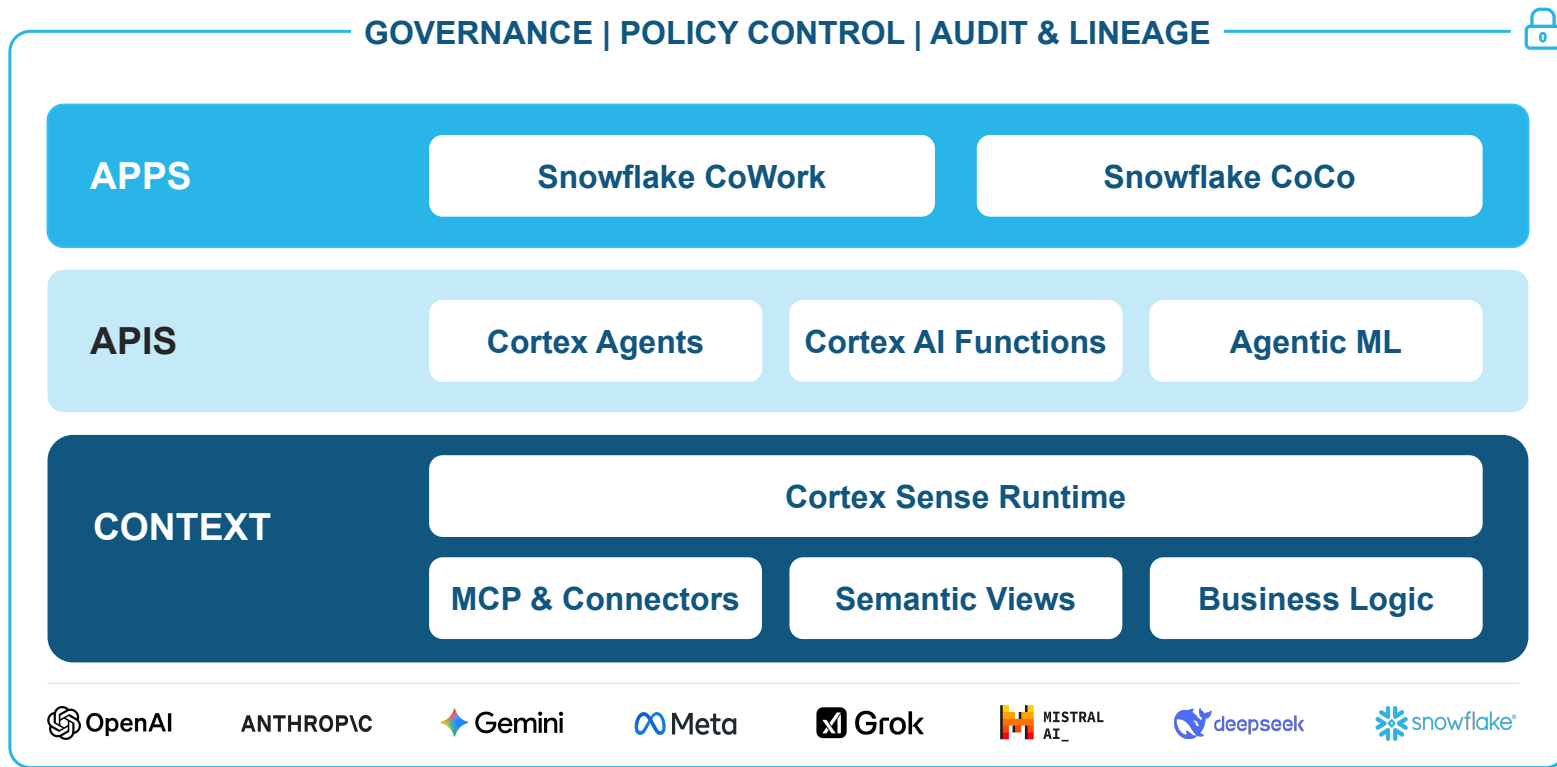
target: "Empower developers to build the impossible"

strategy: "Demystify complex data & cloud ecosystems"

The Shift



Snowflake Cortex AI Architecture



Stages of AI Maturity

01. Tool Adoption

Day 1

Use AI to boost speed,
reduce repetition

02. Workflow Transformation

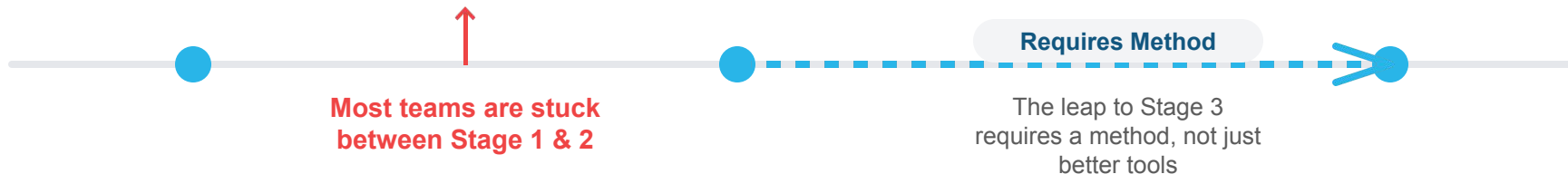
Day 90

Embed AI into team processes,
shift toward co-creation

03. AI-Native

6 Months

Agent-led orchestration,
humans steer strategy and
oversight



Run It, Then Decide

- AI makes prototyping nearly free
- A running system teaches faster than any document
- But cheap building has a shadow:
 - Everyone re-solves the same problem
- First step out of chaos:
 - Survey what exists?
 - Then build what is missing

Documents (Old Map)

8 weeks to first learning

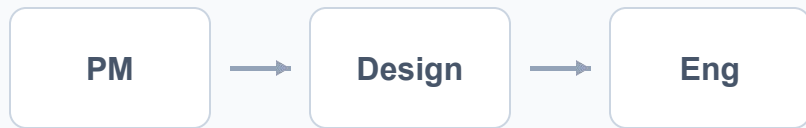
Systems (New Map)

2 hours to first learning

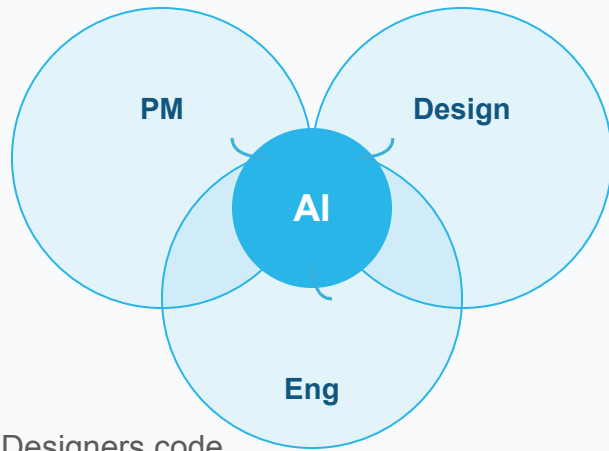


The New Industry Map

OLD: SILOED & SEQUENTIAL



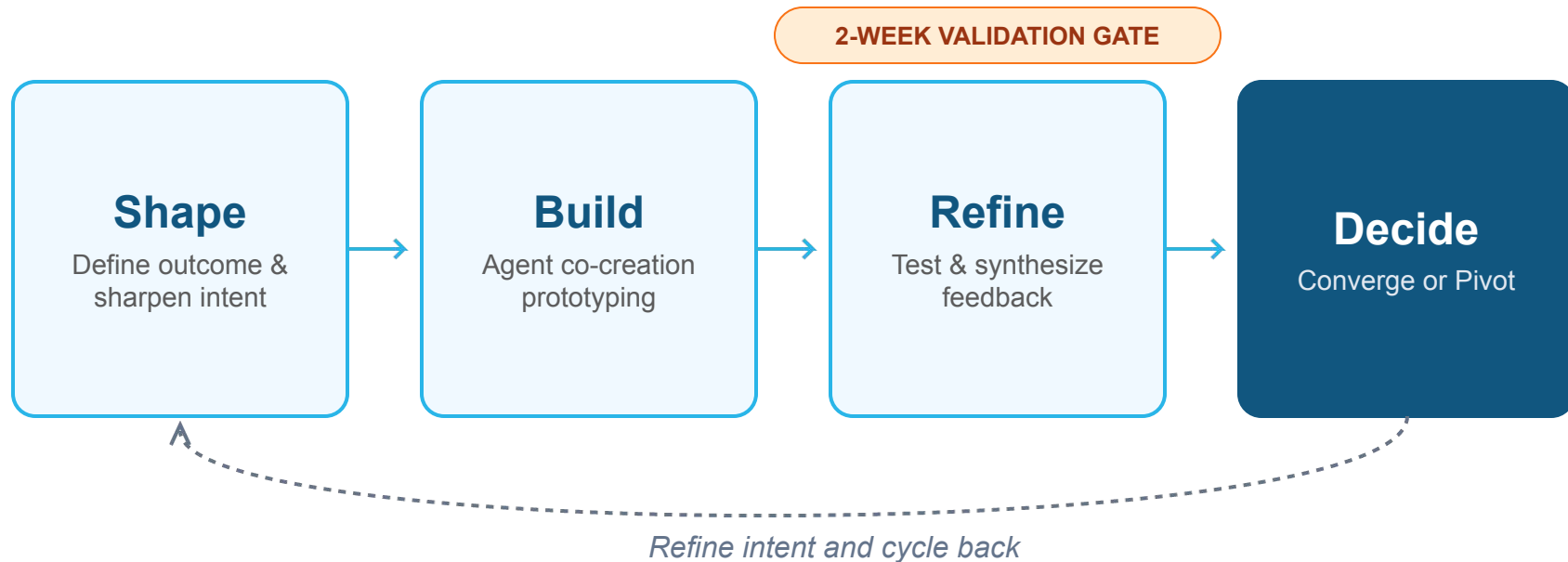
NEW: AI-COLLAPSED DISTANCE



PMs prototype. Engineers craft UX. Designers code.
Fluency across domains, not elimination of roles.



The AI-Native Development Process



"If you cannot validate within 2 weeks, your intent was not clear enough"



THE METHOD



Mental Model: How vs. What

The Syntaxist Traditional / Imperative

- ✗ Focuses on **Imperative** instructions.
- ✗ Spends 80% of time debugging syntax.
- ✗ Struggles with boilerplate and repetition.

The Semanticist AI-Native / Intent-Driven

- ✓ Focuses on **Intent** and outcomes.
- ✓ Spends 80% of time on **Verification**.
- ✓ Uses CoCo as an extension of thought.

Intent-Driven Development (IDD)



Intent-Driven Development (IDD)

OLD STATE: THE SYNTAXIST

Focus on Execution

Knows functions, APIs, and libraries. Operates at the level of implementation details.

NEW STATE: THE SEMANTICIST

Focus on Intent

Knows intent, constraints, and desired state. Operates at the level of outcomes.

- **Philosophy:** Culture requires structure. **Scale your intent.**
- **Paradigm:** Prompts are artifacts. **Treat them as source code.**
- **Outcome-Based:** Vague intent produces noise; precise intent produces solutions.

Don't just build **faster**. Build **better**.



Structuring Intent: The IDD Template



IDD Structure Template

[Goal]

What should exist after execution

[Requirements]

What must be true, not how to get there

[Constraints]

Boundaries, what not to do

[Output]

Verifiable success criteria



IDD In Practice

GOAL

Stored Procedure Loader

Loads CSV files from stage into a Snowflake table.

REQUIREMENTS

Schema Drift & Logging

- Handles schema drift
- Logs row counts during execution

CONSTRAINTS

Boundaries

- Do not drop existing data
- No external libraries

OUTPUT

Verifiable Success

Test load of 100 rows succeeds and log table has entry.



Common Pitfalls to Avoid



Ambiguity Leak

Define the **result**. Don't say "clean the data." Say **what** *clean* means.



Skipping Constraints

Without constraints, the assistant follows the path of least resistance.



Monolithic Trap

Think **modular**. Never prompt an entire pipeline. Build in chunks.



The Visual Delusion

Verify the proof. Code that "looks right" is an illusion. If you can't prove the result, the intent is unfinished.



The Proof



CoCo: The Case Study



Skills > One-off Prompting

Prompt Once

= One Result

A Skill

Reusable intent, runs
forever

Durability: Prompts work once. Skills works every time.

Scope: Agents replace processes, not roles.

Proliferation: Skills solve proliferation: discover first, build only if nothing exists.

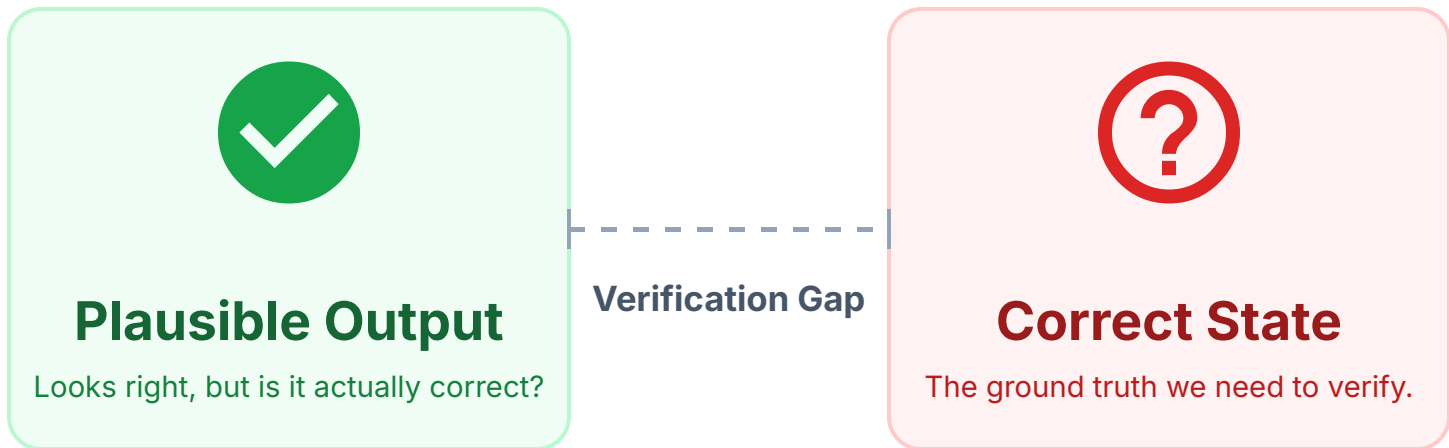
Durable productivity = context in code, not in people's heads



Demo



But How Do We Know It Works?



inspect-coco: Three Files

An evaluation needs three files. No platform, no dashboard, no ceremony.

instruction.md

Goal

Create a file `/workspace/hello.txt` containing exactly the text "Hello, World!".

Requirements

- The file must be created at the exact path `/workspace/hello.txt`
- The content must be exactly "Hello, World!" with no trailing newline

...

tests/test.sh

```
#!/bin/bash
```

```
# verification logic
```

```
exit 0
```

task.toml

```
[metadata]
```

```
epochs = 5
```

```
idd_threshold = 0.6
```

```
[agent]
```

```
timeout_sec = 300
```

```
hello-world (1 x 3 samples): none/none

epochs: 3, task_dir: examples/hello-world, dataset: hello-world

total time: 0:00:22

Verification      IDD Quality
Pass Rate    0.000  IDD Score    1.000

Log: logs/2026-06-20T06-43-02-00-00_hello-world_bCPSita9kvrshHtZemtB3BG.eval

hello-world


| Epoch  | Result | Score | IDD            |
|--------|--------|-------|----------------|
| 1      | PASS   | 1.00  | 1.00           |
| 2      | FAIL   | 0.00  | 1.00           |
| 3      | FAIL   | 0.00  | 1.00           |
| pass@3 | 1/3    | 0.33  | variance=0.222 |



inspect-coco > # View results: uv run inspect view --port 7575
zsh: command not found: #
inspect-coco >
```

What You Take Home

01

The shift is organizational.

Not technical.

02

Structure your intent.

Goal. Requirements. Constraints. Output.

03

Measure compression.

ICR: ops per decision.

04

Test your skills.

No evidence, no trust.

05

Run it, then decide.

The loop is the product.





**Intent Is the Source.
Evidence Is the Contract**



Resources



THANK YOU



© 2026 Snowflake Inc. All Rights Reserved

Ship with Confidence

- ✔ **IDD is a Discipline:** Move from "how" to "what".
- ✔ **Context is Fuel:** Ground intent in live schemas and rules.
- ✔ **Verify Everything:** CoCo executes, your tests to prove it.

**Don't just build faster.
Build better.**

