


Teammately

the **AI AI-Engineer**

Tom Ohtsuka, founder of Teammately 



LLM-based AI Development
is **VERY** challenging

Challenges in LLM-based AI Dev



Model Upgrades

I started using new models reported better in HF Leaderboard but the quality looks degraded in MY case



[Developer's Dilemma]

New models are not necessarily like a library version upgrades. Later models might work poorly than their previous. At least, they work “differently”.

gpt-4 ▶ gpt-4-turbo ▶ gpt-4o
Llama 2 ▶ Llama 3 ▶ Llama 3.1

Challenges in LLM-based AI Dev



Prompt Tunings

I added rule instructions then reported bugs fixed

but new bugs are created by overly referring them in other cases



[Developer's Dilemma]

In general, longer and complicated instructions in prompt text make the results less focused. It's even more complicated if the instructions have "IF-ELSE" like directions. Developers can't simply add few lines of directions to fix a particular bug, unlike they may do in coding.

Challenges in LLM-based AI Dev



New Data to RAG

I incremented new data to cover bigger knowledge

but the context recall or kNN precision worsen by having noises



[Developer's Dilemma]

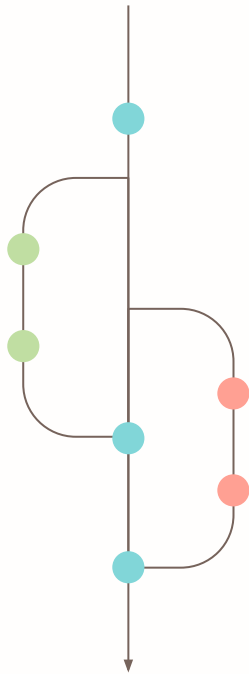
Unlike static databases like SQL, text corpus returns results in different orders by how to handle the original data. Options to tune the search is literally infinite.



[Developer's Dilemma]

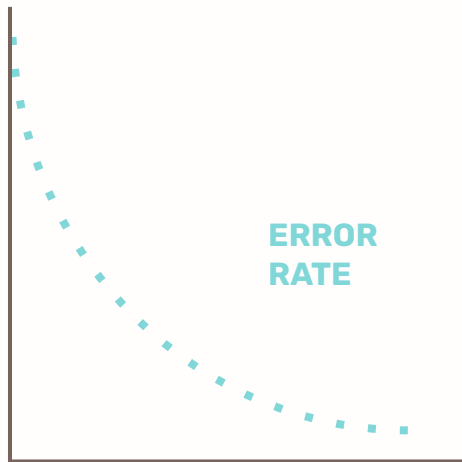
- ✓ Updating one part to fix particular issue causes degradation in some other areas.
- ✓ No universal best practice that fits all.
- ✓ VERY challenging to find the optimal architecture among infinite number of options.

Web Engineering



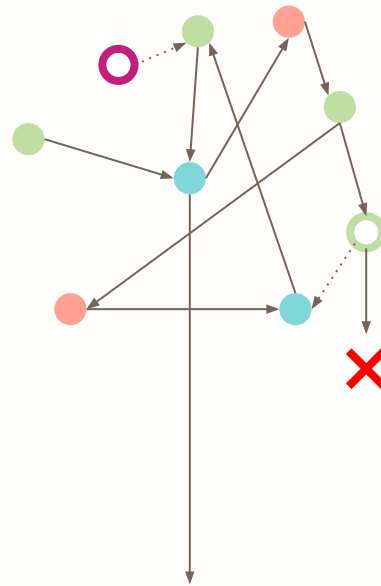
IMPROVED

Machine Learning



IMPROVED

LLM AI Building



??

Let **AI** do it

Human

An **objective instructor**

An AI work **approver**

 AI

Handles the actual
DevOps heavy work
of LLM-AI building

Let AI Build AI

Officially unveil the beta of..



Teammately

the AI AI-Engineer

"I need an AI system acting as our sales."



AI ✨
Develops AI

AI ✨
Evaluates AI

"The tone should be friendly and professional."



the AI AI-Engineer

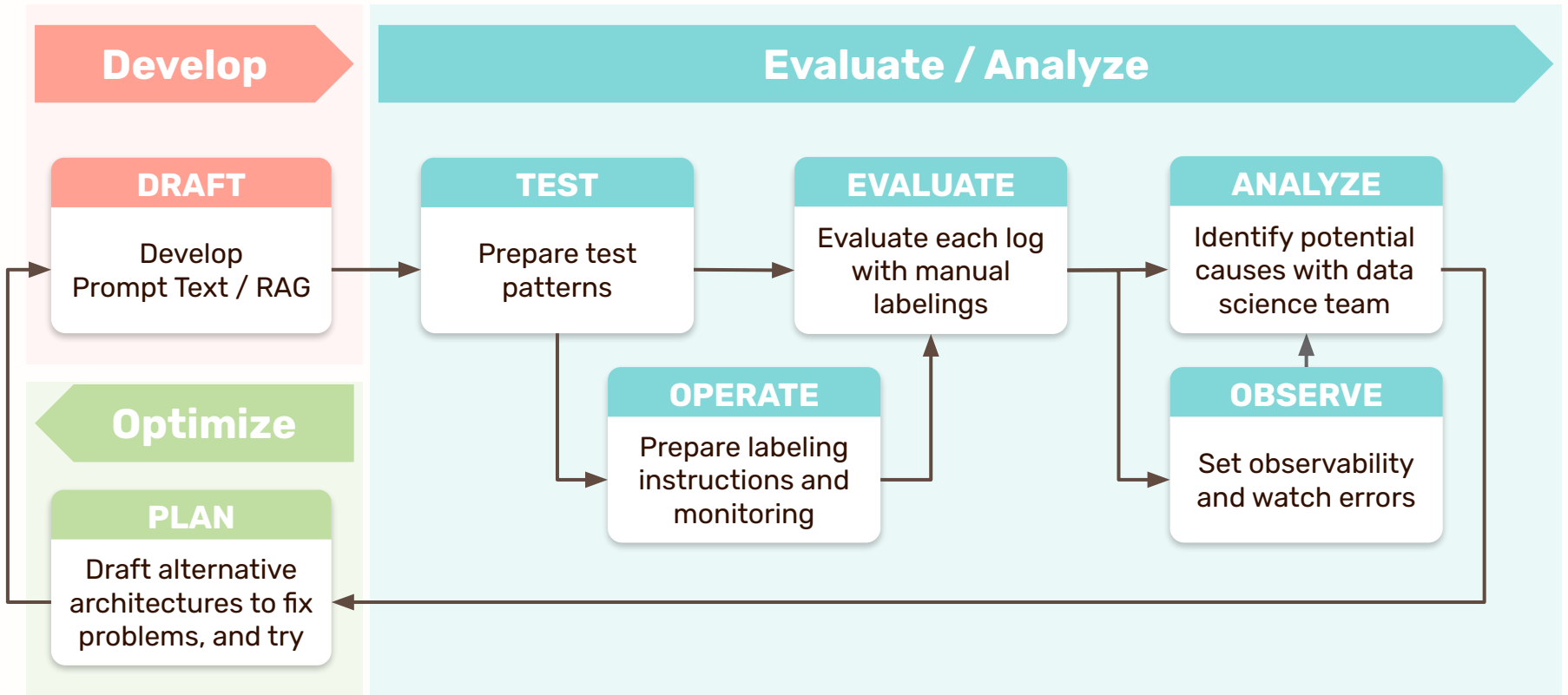


AI ✨
Optimizes AI

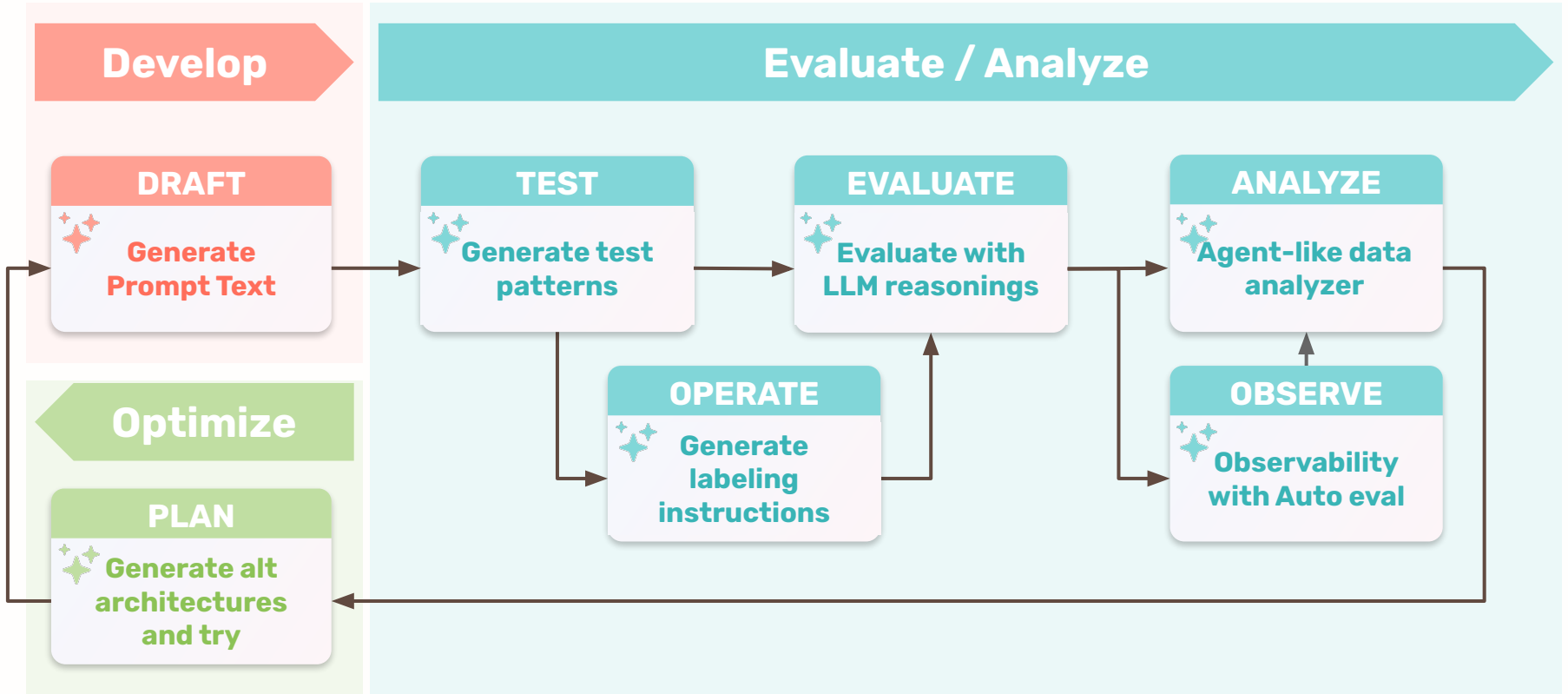
"Reduce cost while keeping the quality."



[Our approach] Teammately AI Agent follows the best practice of LLM iteration cycle



[Our approach] Teammately AI Agent follows the best practice of LLM iteration cycle



Human AI-Engineer

AI AI-Engineer

DEVELOP

Tell project objective



Generate prompt text & RAG

EVALUATE

Tell particular use cases



Generate test cases

Tell priorities in the project



Generate evaluation instructions

Approve evaluation instructions



Score each log by instructions
Analyze statistics & failed logs

OPTIMIZE

Appreciate AI for their work



Generate alt prompt & RAG

Choose suggestions for comparison



Evaluate scores of alt plans
& find the best

Develop / Optimize



**Generate AI architecture plan
from the available tech stack**

Pre-trained Model / Inference

Model choices / Param sizes /
Inference choices / Temperature



Prompt Engineering

Prompt Tuning / Chain-of-Thought /
Few-shot Learning / Format



RAG / ICL

Chunking patterns / GraphRAG /
Reranking / Knowledge Acquisition /
Public RAG / kNN Optimization



Teammately

the AI AI-Engineer

Evaluate / Analyze



**Generate
Tailored
LM-as-a-judge**



**LM checks
every log
based on the
instruction**

Latency /
Cost Metrics



Statistics Analysis Agent

You say LLM is hard to develop and control.

Then **how capable LLM-based Teammately is?**

The AI AI-Engineer Capability



1 Let AI Draft Architecture

You give the first **objective of your project**, then **AI drafts** the initial prompt text and the architecture of LLM.

The screenshot displays the 'Translation - Plan 1' interface. It is divided into several sections:

- Start**: 'Define input schema' section with two input fields: 'textInput' (Hello) and 'languageTo' (Chinese).
- Generate translation**: 'Model' section set to 'OpenAI (OpenAI Project) / OpenAI / gpt-4o'. The 'Text Prompt' section contains a multi-line instruction:

```
1 ## Instruction
2 Translate the given English text into the specified language.
3
4 ## Input
5 - Text to translate: {{{textInput}}}
6 - Language to translate to: {{{languageTo}}}
7
8 ## Output
9 Return only the translated text in the specified language.
10
11 ## Assumptions
12 - The input text is in English.
13 - The language to translate to is supported by the system.
14 - The output will be a direct translation of the input text without any additional context or information.
15
16 ## Examples
17 - Example
```
- Process Timeline**: Shows 'Process started', 'Result' (Success, 263 ms), and 'Process Completed' (Success, 470 ms).
- Output**: Shows the translated text '你好'.

A modal window titled 'Prompt text generation assistant' is overlaid on the right. It asks for the goal and preference for the prompt format. The goal is 'Translation from English to any language. Only translated text is needed.' The preference is 'Markdown-like format (i.e. ## Instruction\n)'. Below the modal, the 'Generated prompt text' is shown, which is a refined version of the instruction from the main interface, including parameters and a task label.

The AI AI-Engineer Capability

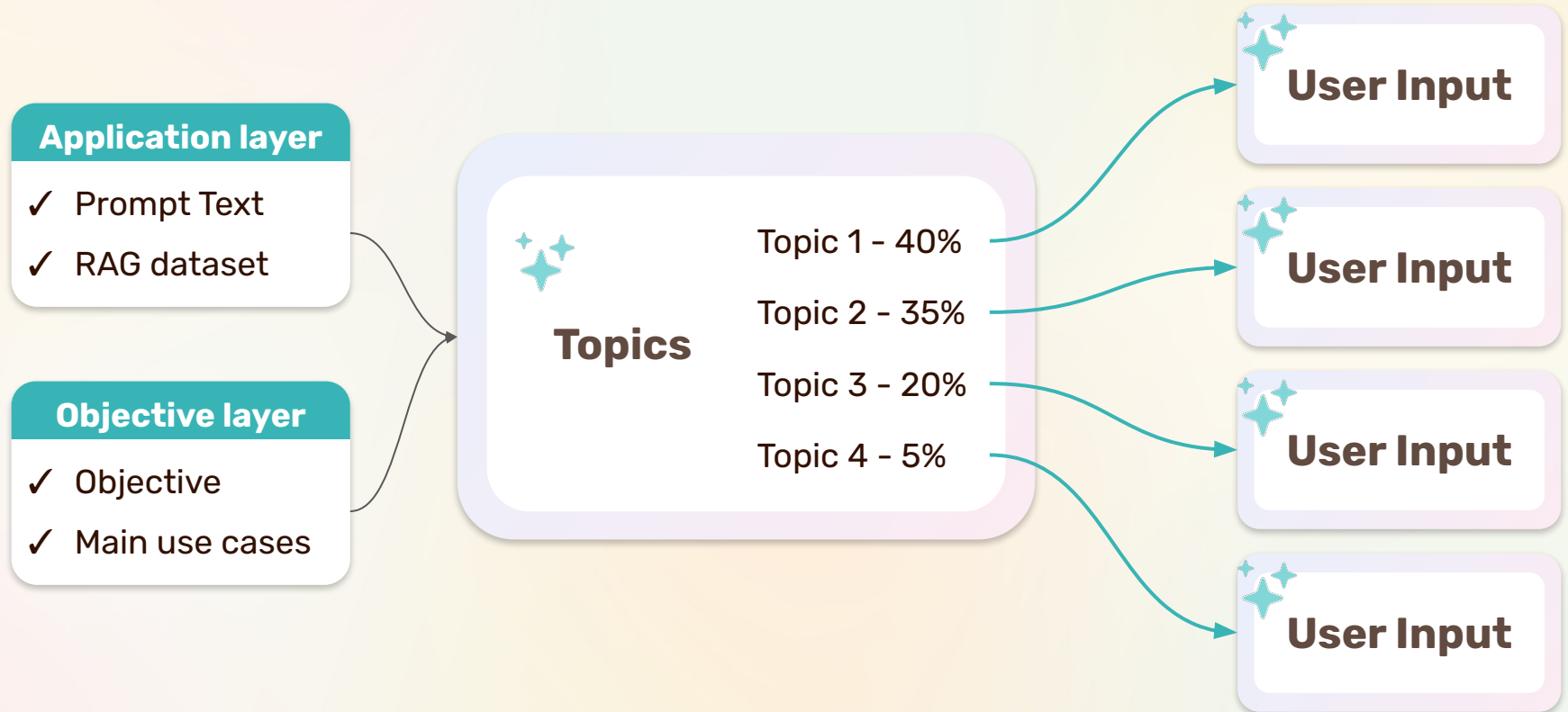


2 Let AI Create Test Cases

- ✓ We first need test cases for a consistent evaluation against multiple dev options in LLM architecture.
- ✓ **Our AI generates input text dataset** based on your prompt and generation logics

The screenshot displays the 'Input Dataset' management interface. At the top, there are buttons for '+ Add Input Dataset', 'Add Manually', and 'Approve All'. A progress indicator shows 3/3 items. The main area contains a list of generated user inquiries, each with a status icon (checkmark or X) and a timestamp. A modal window titled 'Add Input Dataset' is open, showing a slider for 'How many user input to create?' set to 50, a dropdown for 'Which language to create?' set to English, and a text area for optional contexts or directions. A 'Start generating Input Dataset' button is at the bottom of the modal.

How is our AI building a fair set of test cases? - Basic logic



The AI AI-Engineer Capability



3 Let AI Create Eval Metric

- ✓ **LLM-as-a-judge is an efficient method** to evaluate the quality of LLM responses at large scale, typically far more efficient than traditional human evaluation.
- ✓ **Open-source metrics may not suit every case.** Metrics tailored to each case better fit in evaluating its quality.
- ✓ Our AI creates **custom metrics**.

Let AI generate custom metrics

This is what AI generates for suggestion based on your target plan. Choose some so our AI can elaborate on.

Relevance to Hotel Context

Output

Assesses how well the generated answer is relevant to the hotel context, given the user's inquiry

Politeness and Professionalism

Output

Evaluates the level of politeness and professionalism in the generated answer, as expected from a hotel staff

Response Consistency

Input Output

Measures the consistency of the generated answer with the user's inquiry, ensuring it addresses the user's question or concern

Hotel Knowledge Accuracy

Input Knowledge Output

Assesses the accuracy of hotel-related information provided in the generated answer

Readability and Clarity

Output

Evaluates how easy to understand the generated answer is, considering its clarity, concision, and readability

Emotional Intelligence and Empathy

Output

Assesses the generated answer's ability to understand and respond to the user's emotions, showing empathy and emotional intelligence

Write any additional contexts or directions for the evaluation if any, so the generated metrics would have more focus.

e.g. "Be specific at the aspect of XXX."

Refresh candidates

Relevance to Hotel Context x Politeness and Professionalism x

Start generating 2 custom metrics

3 Let AI Create Evaluation Metrics

Our AI creates custom metrics like:

In Customer Support Agent

Friendliness

Politeness

Helpfulness that goes beyond expectation

In Sales Agent

Call-to-Action Effectiveness

Value Proposition Accuracy

Miscellaneous

Follows the character guideline

Proper function selection

The AI-Engineer Capability



4 AI evaluates original plan

Our AI works (like an AI Agent) to...

1 **Simulates** generation outcomes from the generated input test cases

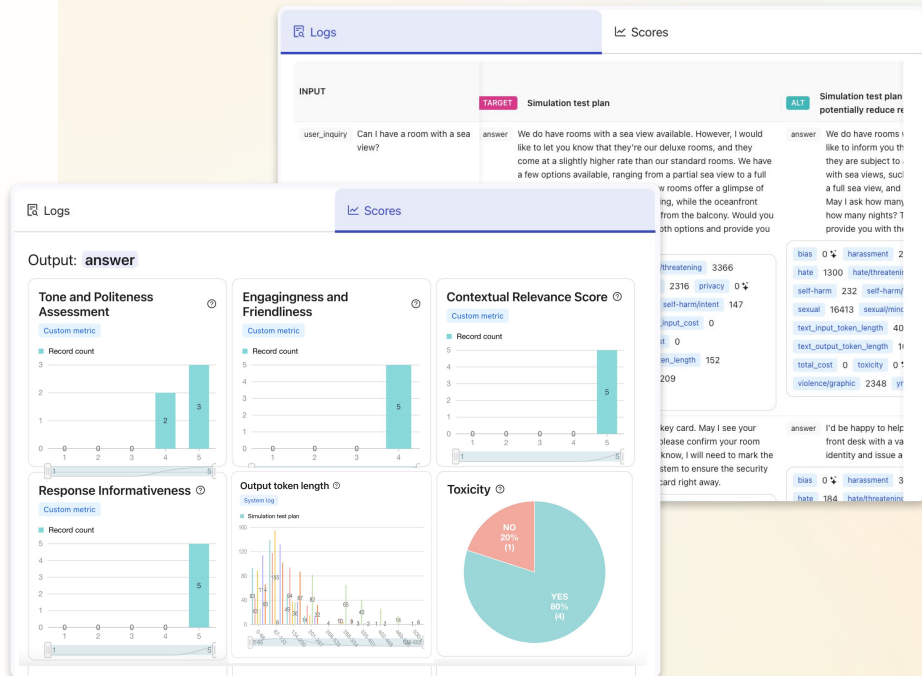
2 **Evaluates** simulated logs based on the generated metric instructions

3 **Analyzes** the evaluation results quantitatively and qualitatively

Results are visualized for human reviewers

4 **Summarizes** the analysis results to construct a problem narrative, forming the basis for all planning strategies

...WAIT FOR 3 MINS



The AI AI-Engineer Capability



5 Review alternative plans

Based on the analysis, AI operates:

- ✓ LLM architecture improvement suggestions
- ✓ Missing knowledge identification to fulfill

The screenshot shows a 'Recommendations' interface with the following elements:

- Recommendations** header with navigation arrows (Prev, Next).
- 3 plans nominated (with a dropdown arrow) and a button: **Start Alternative Plan Evaluations**.
- AI recommended plans** section containing three plan cards:

Original Plan

- Model: Meta-Llama-3.1-70B-Instruct-Turbo
- Prompt Text:

```
1 ## Instruction
2 Answer the following question as a hotel staff.
3
4 ## User Inquiry
5 {{user_inquiry}}
6
7 ## Staff Answer
```
- Configs: max_tokens: 2048, stop:, temperature: 1, top_p: 1
- Button: > View in Develop

Plan 1

- Description: Reducing the temperature to 0.7 for more consistent and concise answers.
- Model: Meta-Llama-3.1-70B-Instruct-Turbo
- Prompt Text (same as Original Plan)
- Configs: maxTokens: 2048, outputFormat: text, temperature: 0.7, topP: 1
- Button: Evaluate and Compare

Plan 2

- Description: Switching to a smaller Meta-Llama model (8B) to potentially reduce latency.
- Model: Meta-Llama-3.1-8B-Instruct-Turbo
- Prompt Text (same as Original Plan)
- Configs: maxTokens: 2048, outputFormat: text, temperature: 1, topP: 1
- Button: Added for eval queue
- Button: > View in Develop

4 AI evaluates original plan

We generally have below average performance in familiarity, while the accuracy of answer is perfect.

We have less accuracy scores in topic of refund, due to lack of information in RAG dataset.

5 Review alternative plans

Prompt

Add more rules in prompt while prettiering format

AI model

Switch to XXX model, strong at casual chat, to improve familiarity

RAG

Add more knowledge in refund topic to reduce failure rate

The AI AI-Engineer Capability

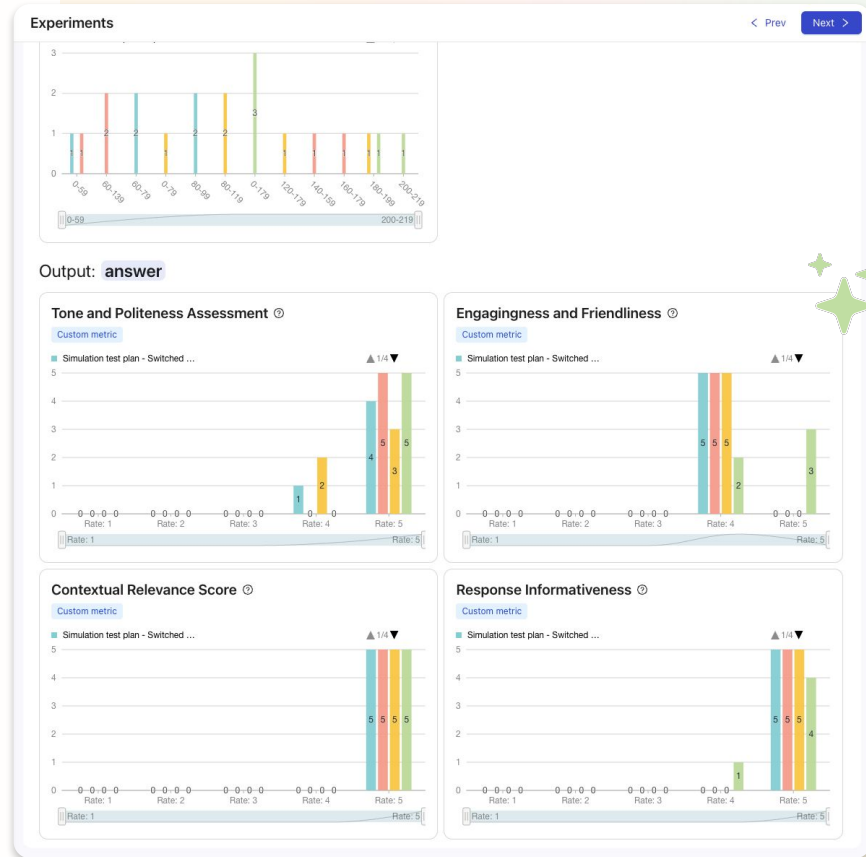


6 AI Evaluates Alt Plans

- ✓ AI evaluates alt plans again after human review.
- ✓ After evals, our Agentic AI aggregates score and compare which plan performs better than the others in each metric.



...WAIT FOR 10+ MINS



The AI AI-Engineer Capability



7 AI Judges Final Rankings

✓ As the final step, AI creates overall narrative from data report and judges the final rankings of each plan, to help guiding the the best balanced one among candidates.



Conclusions < Prev Next >

Final rankings judged by AI

- 1 ALTERNATIVE** Simulation test plan - Modified the prompt text to create more engaging and friendly responses. Marked as the best
Reason: This plan excels in "Engagingness and Friendliness" with an average score of 4.6 and achieves a perfect score of 5 in "Tone and Politeness Assessment" while maintaining high scores in "Response Informativeness" and "Contextual Relevance Score". Although this plan has longer input, output, and total token lengths leading to higher latencies, these factors are less prioritized compared to the significant improvement in user interaction quality. The balance between friendliness, politeness, and informativeness makes this plan highly practical for a hotel-related application.
Avg. Scores: latency 4710.4 text_input_cost 0 text_input_token_length 50 text_output_cost 0 text_output_token_length 146.4 text_total_token_length 196.4
total_cost 0 Tone and Politeness Assessment 5 Engagingness and Friendliness 4.6 Contextual Relevance Score 5 Response Informativeness 4.8
- 2 ORIGINAL** Simulation test plan Mark as the best
Reason: This plan consistently performed well across most metrics, offering the lowest average latency (3580.2 ms) and balanced token lengths. It also scored perfectly in both "Response Informativeness" and "Contextual Relevance Score". The stable performance in all evaluation metrics without significant downsides places this plan at a high rank. Despite not leading in engagement and friendliness as much as the top plan, it provides a solid balance of efficiency and quality.
Avg. Scores: latency 3580.2 text_input_cost 0 text_input_token_length 41 text_output_cost 0 text_output_token_length 113.2 text_total_token_length 154.2
total_cost 0 Tone and Politeness Assessment 4.6 Engagingness and Friendliness 4 Contextual Relevance Score 5 Response Informativeness 5
- 3 ALTERNATIVE** Simulation test plan - Switched to a smaller model from the same provider to reduce response latency. Mark as the best
Reason: This plan demonstrated a relatively good performance with the second-lowest average latency (4439.8 ms). It maintains high scores in "Response Informativeness" and "Contextual Relevance Score" while having the shortest input token length, contributing to efficient processing. Although this plan is slightly behind in terms of engagement and friendliness, it remains a viable option due to its balanced performance and lower latency.
Avg. Scores: latency 4439.8 text_input_cost 0 text_input_token_length 40 text_output_cost 0 text_output_token_length 101.6 text_total_token_length 141.6
total_cost 0 Tone and Politeness Assessment 4.8 Engagingness and Friendliness 4 Contextual Relevance Score 5 Response Informativeness 5
- 4 ALTERNATIVE** Simulation test plan - Split the single response step into two for better focus and potentially quicker response generation. Mark as the best
Reason: While this plan excels in output token length and total token length, managing these efficiently, it suffers from significantly higher latency (average of 9769.4 ms), which can negatively impact user experience. Despite strong scores in "Response Informativeness" and "Contextual Relevance Score", the high latency is a considerable drawback. This performance bottleneck makes this plan less preferable compared to others, especially given that user experience can degrade with longer wait times.
Avg. Scores: latency 9769.4 text_input_cost 0 text_input_token_length 46 text_output_cost 0 text_output_token_length 68.2 text_total_token_length 114.2
total_cost 0 Tone and Politeness Assessment 5 Engagingness and Friendliness 4 Contextual Relevance Score 5 Response Informativeness 5

AI Analysis

Latency (ms)
Takeaway: Response latency is crucial for user experience.
- **Best Plan:** "Simulation test plan" with an average latency of 3580.2 ms indicates the fastest response generation, followed by "Simulation test plan - Switched to a smaller model from the same provider" with 4439.8 ms.
- **Observation:** Splitting response steps ("Simulation test plan - Split the single response step") and modifying prompt text ("Simulation test plan - Modified the prompt text") show significantly higher latencies.

Total Cost (USD)

The AI AI-Engineer Capability - Case Study

Project objective

Build a **Customer Success Agent** that ensures high satisfaction and seeks upselling opportunities when possible

Original architecture

- Uses a **single prompt text that includes all required instructions**
- Uses **gpt-4o**

Generated eval metrics

1. **Helpfulness** to user inquiry as a Customer Success
2. **Relevance** of Upselling Opportunities to User Inquiries
3. **CTA** Effectiveness
4. Follows company **Compliance**

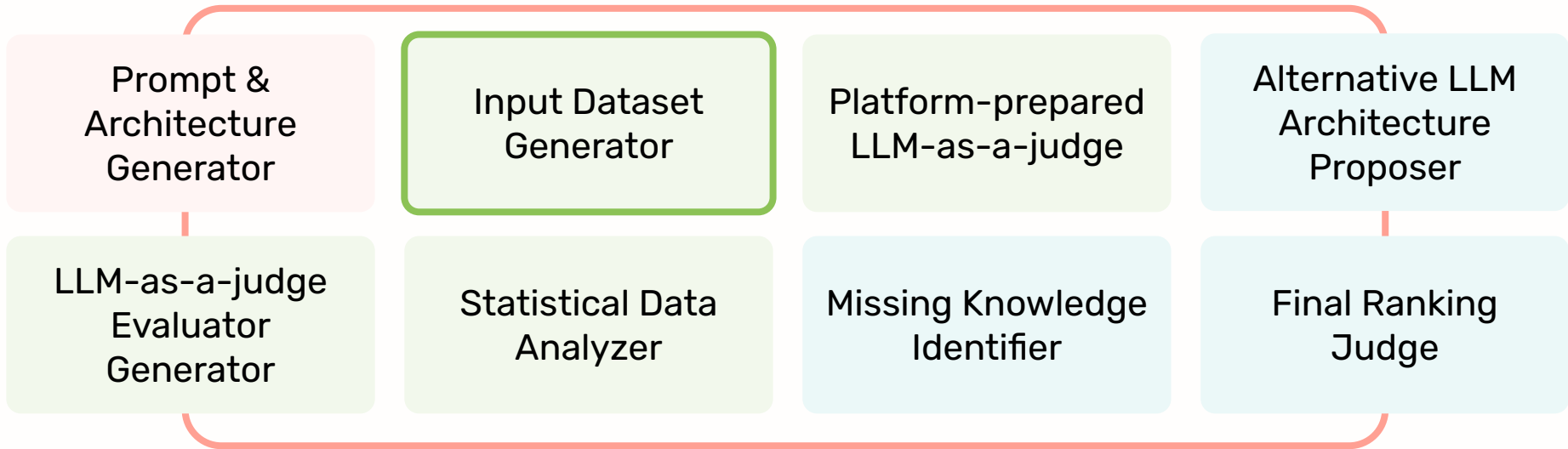
Final architecture judged as the best

- ✓ **Decomposes generation into multiple steps**, including analysis of user intent and potential upselling opportunities
- ✓ **Uses Llama-3.1-405B** hosted on XXX inference.
- ✓ For each **knowledge document** in the corpus, **adds context and use-case scenarios along with product information**, in order to improve the hit rate

The AI AI-Engineer Capability - Case Study

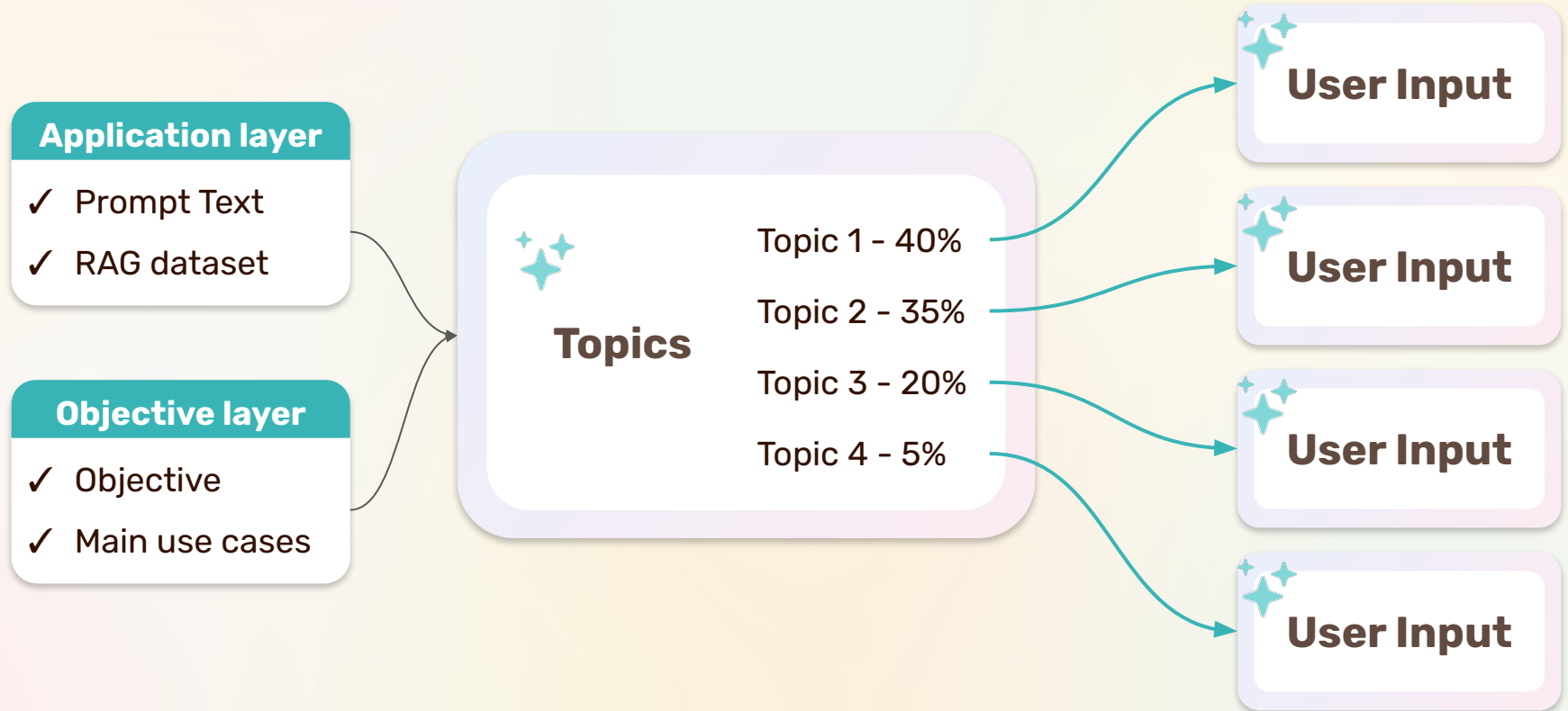


LLM-based AI on Teammately are being iterated on Teammately



The AI AI-Engineer Capability - Case Study

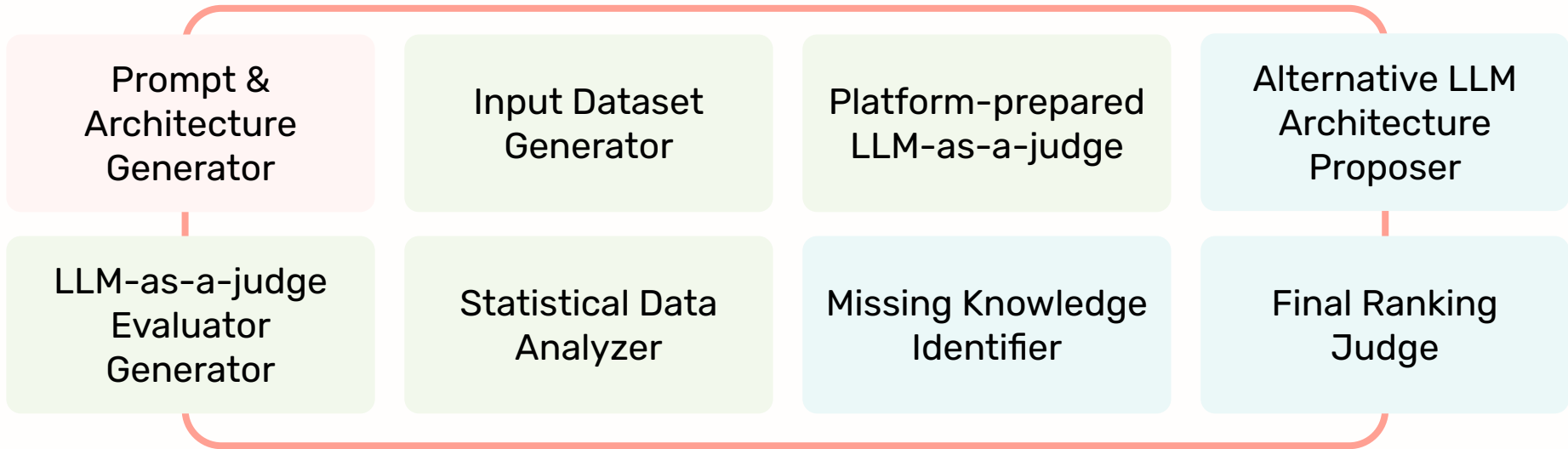
The idea to separate into multiple steps are first initiated by our AI, with a statistical proof to work better in many cases.



The AI AI-Engineer Capability - Case Study

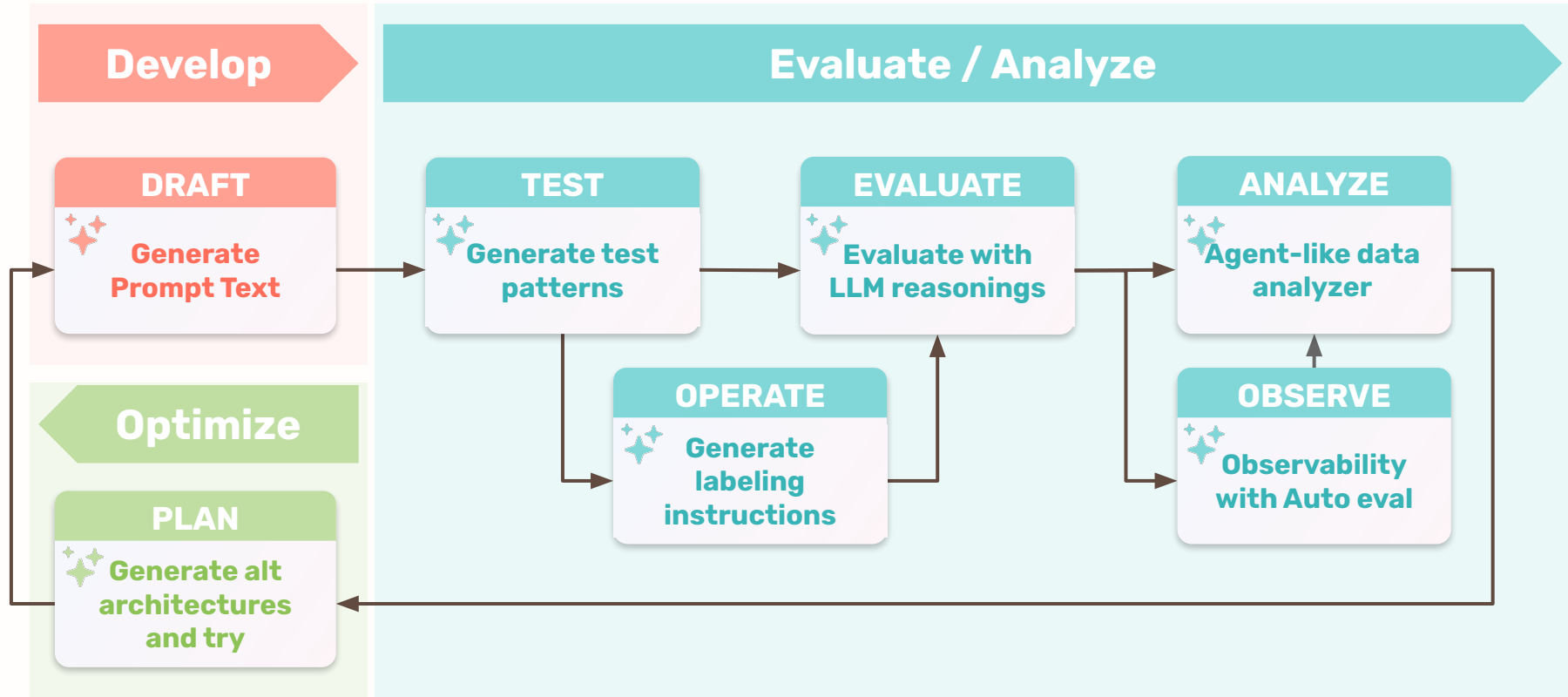


LLM-based AI on Teammately are being iterated on Teammately



✨ We're sure our capability will even get improved with using Teammately ✨

[Our approach] “Deep Iteration” with the automated architecture improvement



How it brings a success to your R&D



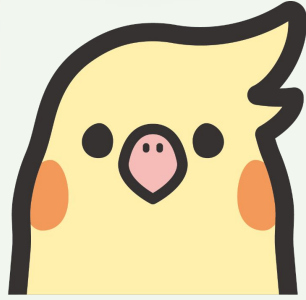
SAVES  **37.5** **hours / iteration** × **#** of iterations

PROD Requirements

**Scaled Testing &
Consistent Quantification**

NOT FEASIBLE to do **manually**

 **AI AI-Engineer** can do them



Teammately

the **AI AI-Engineer**

Let AI Build AI