ramp

Why We Built an Agent, Not an AI Oncall Engineer

Ryan Stevens, Director of Applied Sciences at Ramp
July 17, 2025

WE'RE BUILDING AN AI WORKER
WE'RE AUTOMATING BORING TASKS
imgflip.com

01 Tasks Framework
02 Applying Tasks to AE Oncall
03 Building our Agent

01 **Tasks Framework**
02 Applying Tasks to AE Oncall
03 Building our Agent

AI is automating ***tasks***, not ***jobs***

Subtitle

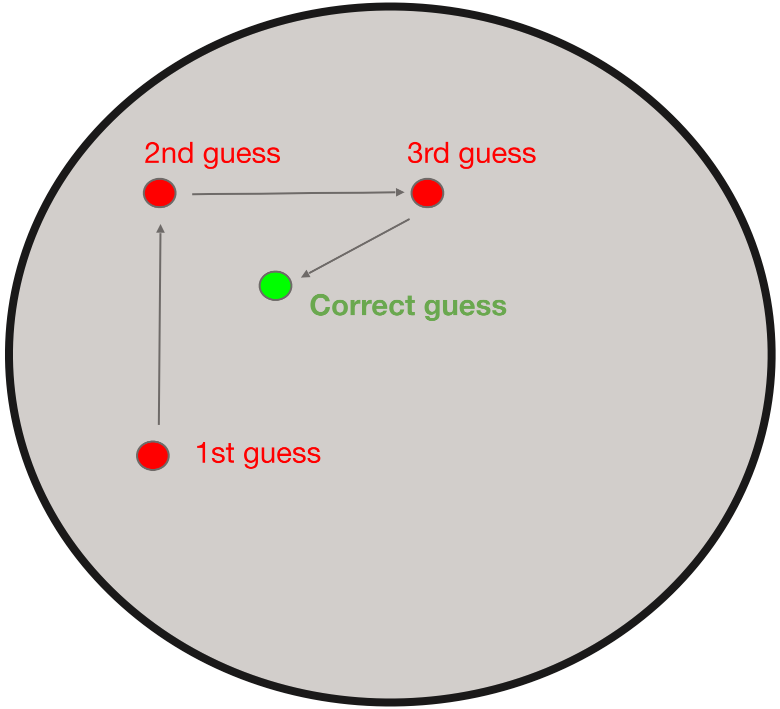
About 27% of firms using AI report replacing worker tasks, but only about 5% experience employment change due to AI use.

Bonney et al, Nov 2024

Tasks are bounded, tactical, metric oriented while jobs are unbounded, strategic, goal oriented

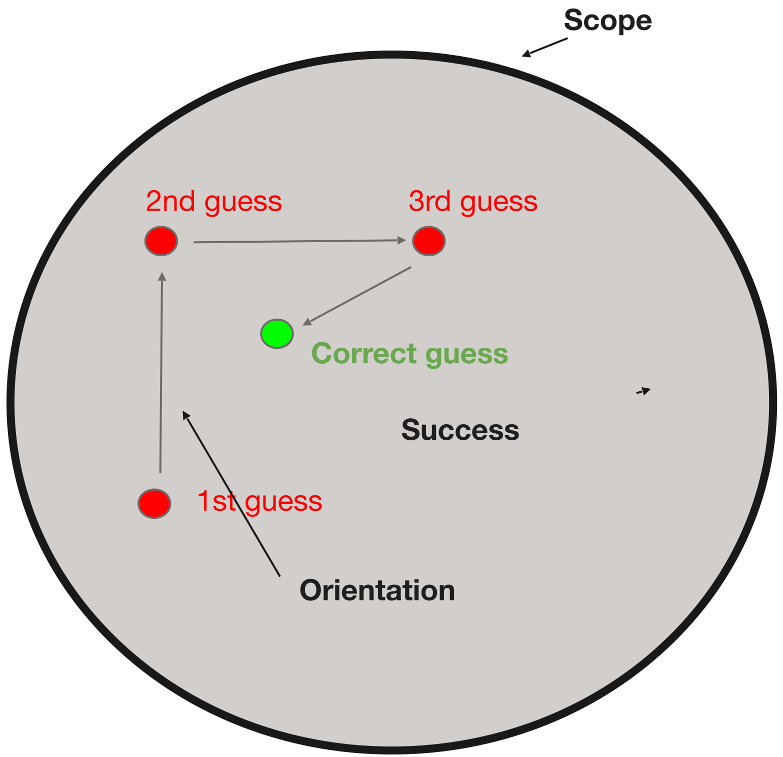
	Tasks (Agents handle well)	Job (Agents struggle with)
Scope	**Bounded** Clearly defined start, end, and scope	**Unbounded** Evolving scope, ambiguous or shifting boundaries
Orientation	**Tactical** Execution-focused, short-term actions	**Strategic** Prioritization and adaptation to new domains
Success Criteria	**Metric-Oriented** Measured by precise outputs (e.g., accuracy, speed)	**Goal-Oriented** Judged by broader outcomes or evolving objectives

Tasks give agents direction (orientation), a defined space to explore (scope), and concept of correctness (success)



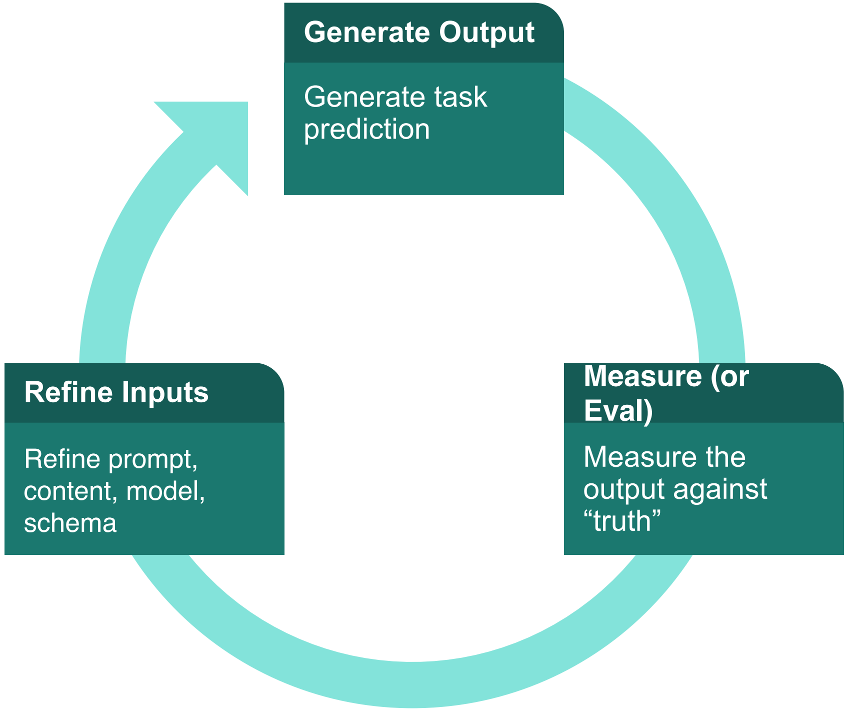
Guesses	Accuracy
First	60%
Second	70%
Third	65%

Tasks give agents direction (orientation), a defined space to explore (scope), and concept of correctness (success)

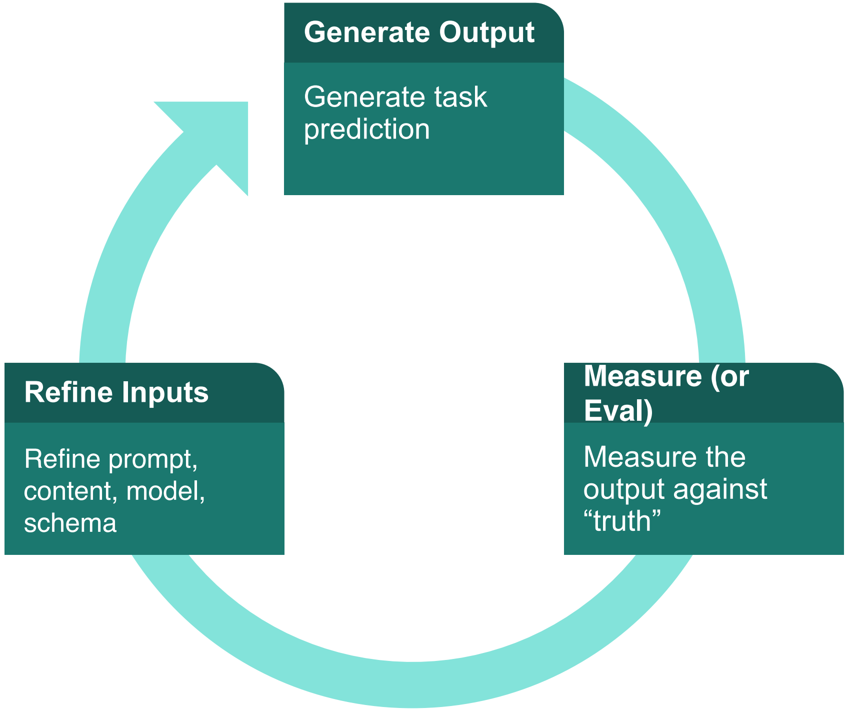


Guesses	Accuracy
First	60%
Second	70%
Third	65%

Tasks → Feedback Loops → Performant Agents


Generate Output
Generate task prediction
Measure (or Eval)
Measure the output against “truth”
Refine Inputs
Refine prompt, content, model, schema

Tasks → Feedback Loops → Performant Agents


Generate Output
Generate task prediction
Measure (or Eval)
Measure the output against "truth"
Refine Inputs
Refine prompt, content, model, schema

This isn't new!

ML / Predictive Modelling folks have been doing for multiple decades

AI literally reinvented this wheel

01 Tasks Framework

02 **Applying Tasks to AE Oncall**

03 Building our Agent

Analytics Engineering (AE) function provides vital data services

Procure Ingredients › **Clean and Prep Raw Materials** › **Creating Recipes** › **Neatly Package Recipes and Portioned Ingredients for Delivery**

Extract Data Sources Using API connections › **Clean and Prep Raw Data** › **Creating Data Marts** › **Neatly Package Data Marts and Documentation for BI tool / Querying**

Fivetran

snowflake

dbt_

AE Oncall’s job is get ETL jobs finished and assist with investigations

Subtitle

Coordinate with upstream (engineers) and downstream (end-users) stakeholders

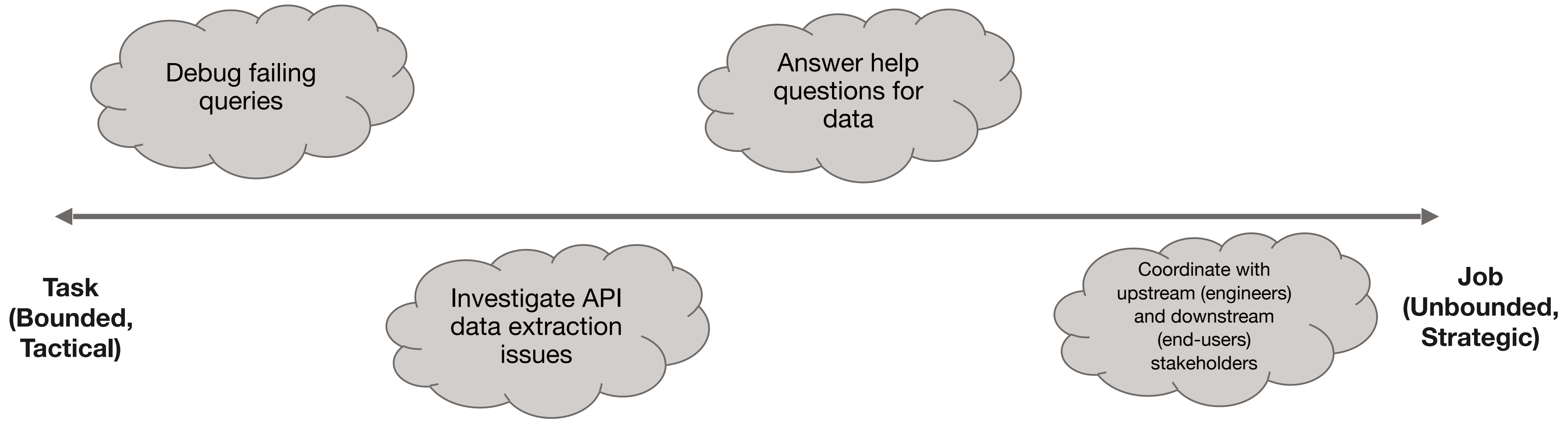
Debug failing queries

Investigate API data extraction issues

Answer help questions for data

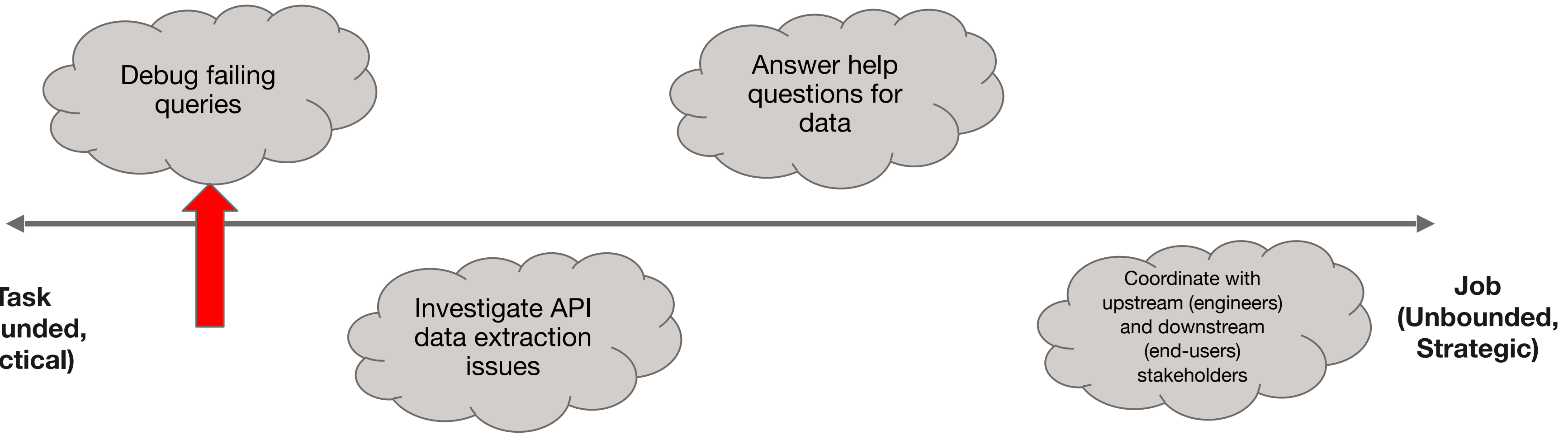
Let’s organize from tasks → job

Subtitle


Debug failing queries
Answer help questions for data
Investigate API data extraction issues
Coordinate with upstream (engineers) and downstream (end-users) stakeholders
Task (Bounded, Tactical)
Job (Unbounded, Strategic)

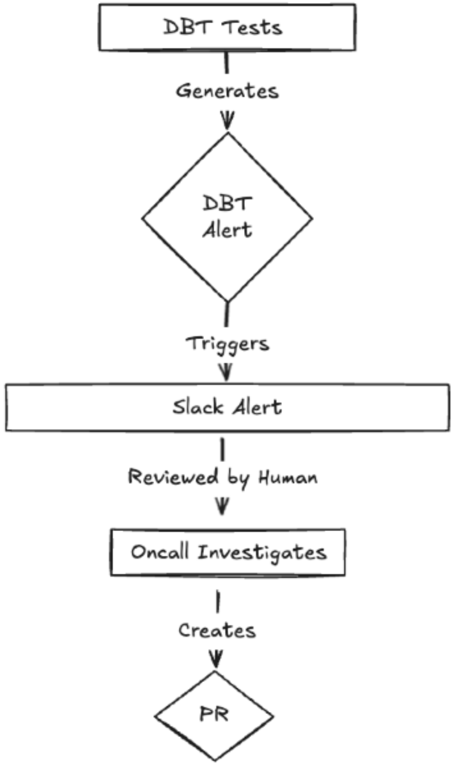
Let’s organize from tasks → job

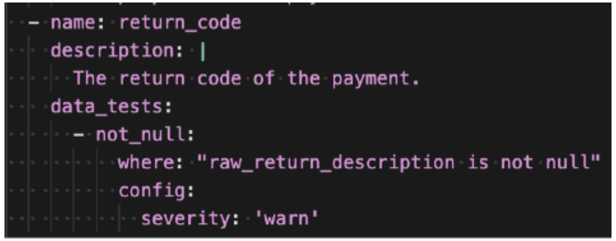
Subtitle


Debug failing queries
Answer help questions for data
Task (Bounded, Tactical)
Investigate API data extraction issues
Coordinate with upstream (engineers) and downstream (end-users) stakeholders
Job (Unbounded, Strategic)

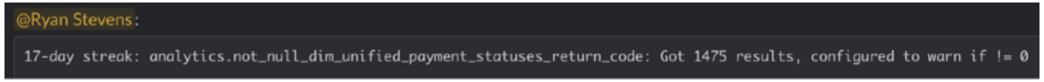
Query debugging was a heavily manual process

Subtitle



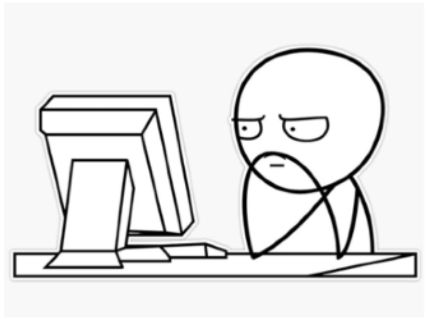


```
  - name: return_code
    description: |
      The return code of the payment.
    data_tests:
      - not_null:
          where: "raw_return_description is not null"
          config:
            severity: 'warn'
```



@Ryan Stevens:

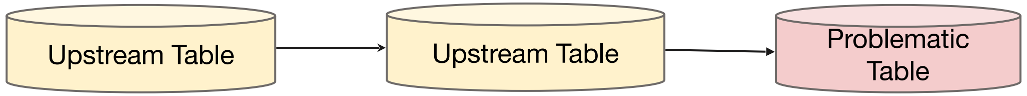
```
17-day streak: analytics.not_null_dim_unified_payment_statuses_return_code: Got 1475 results, configured to warn if != 0
```



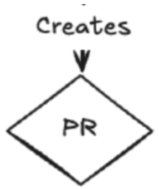
Debugging follows the task framework

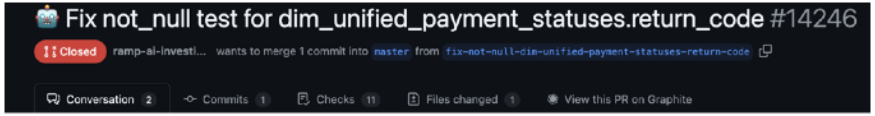
Subtitle

Bounded



Tactical Goal



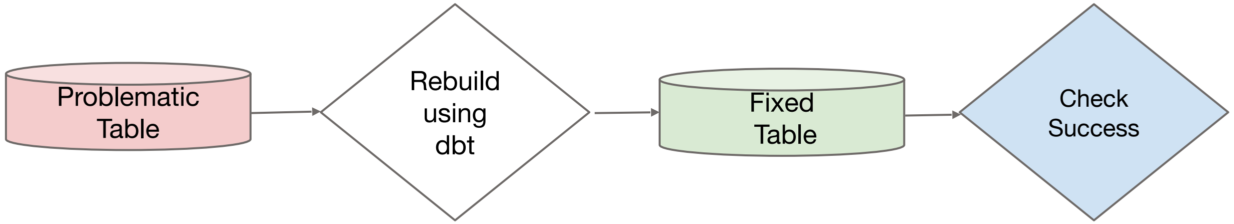


🤖 Fix not_null test for dim_unified_payment_statuses.return_code #14246

Closed ramp-ai-investi... wants to merge 1 commit into `master` from `fix-not-null-dim-unified-payment-statuses-return-code`

Conversation 2 | Commits 1 | Checks 11 | Files changed 1 | View this PR on Graphite

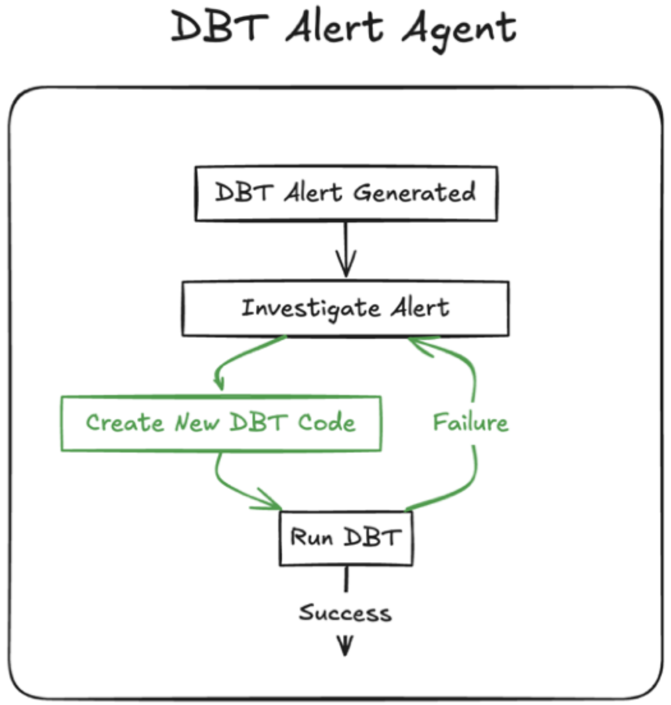
Success Criteria



01 Tasks Framework
02 Applying Tasks to AE Oncall
03 **Building our Agent**

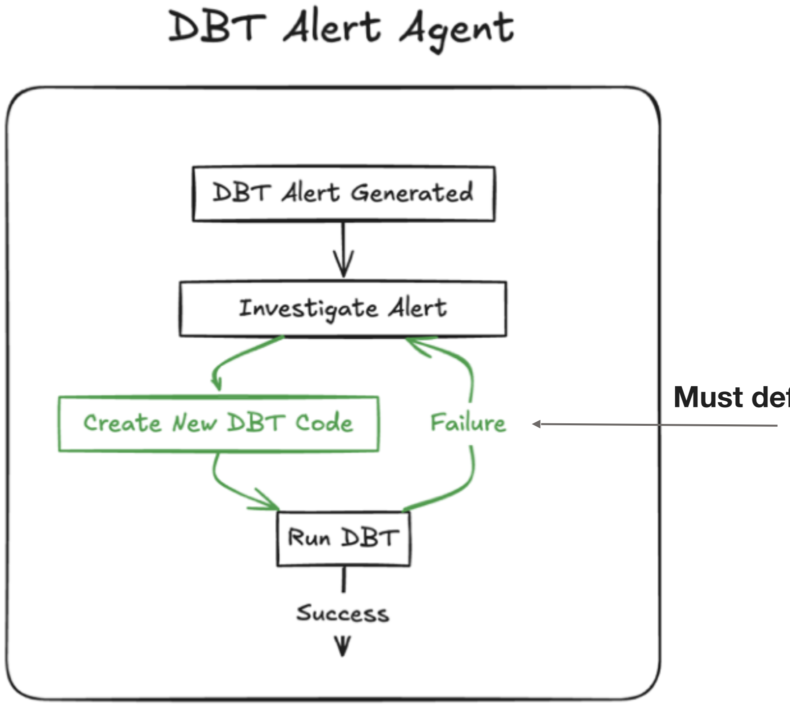
Task framework allows code generation with feedback loop

Subtitle



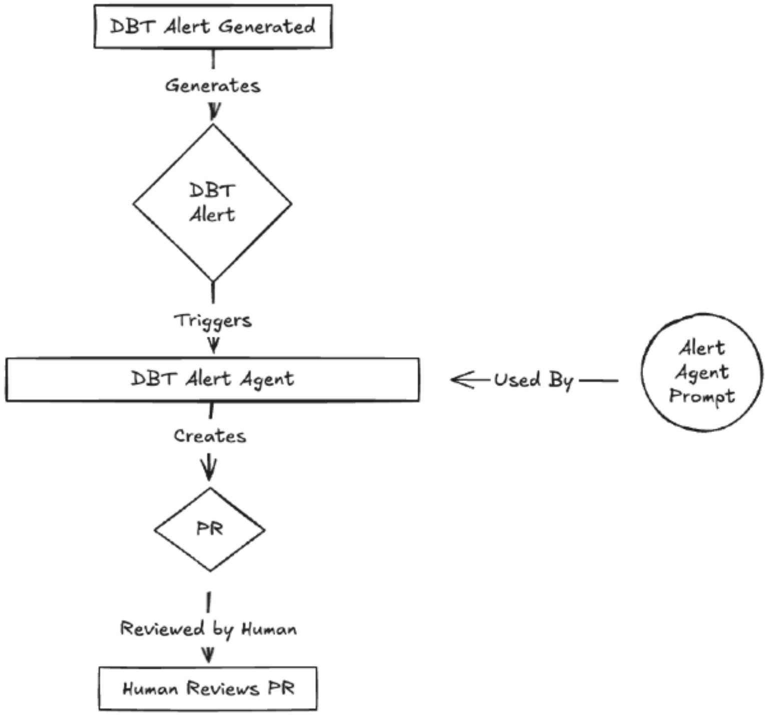
Success criteria is critical to this loop

Subtitle

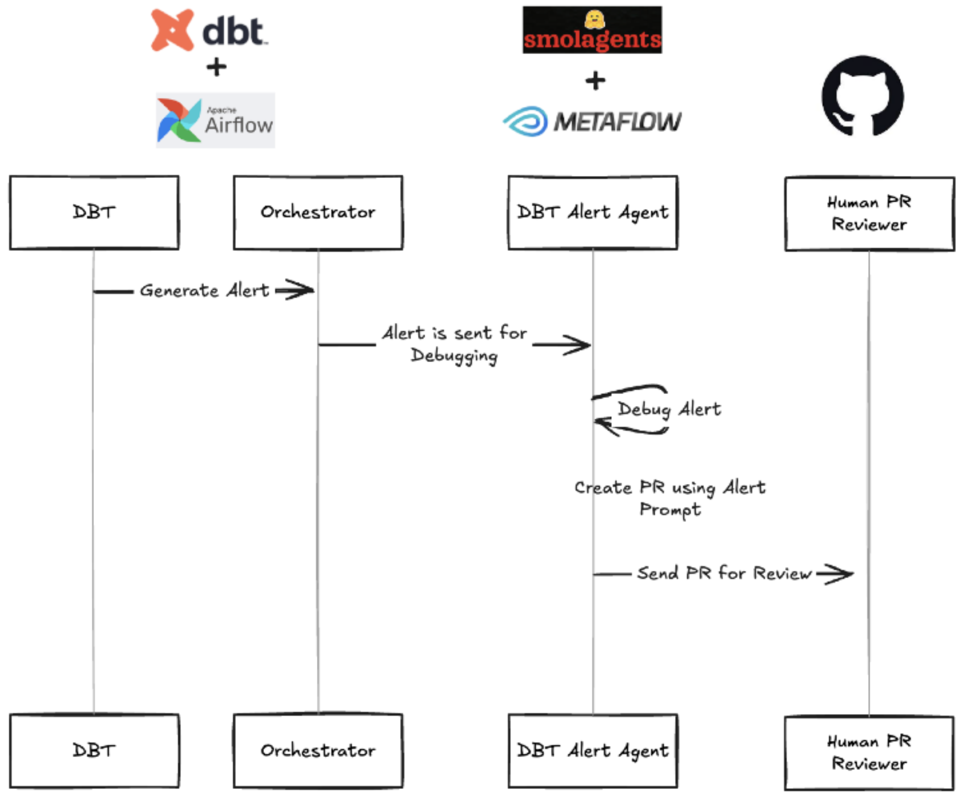


Must define success to have this step

First version simply focused on PR generation

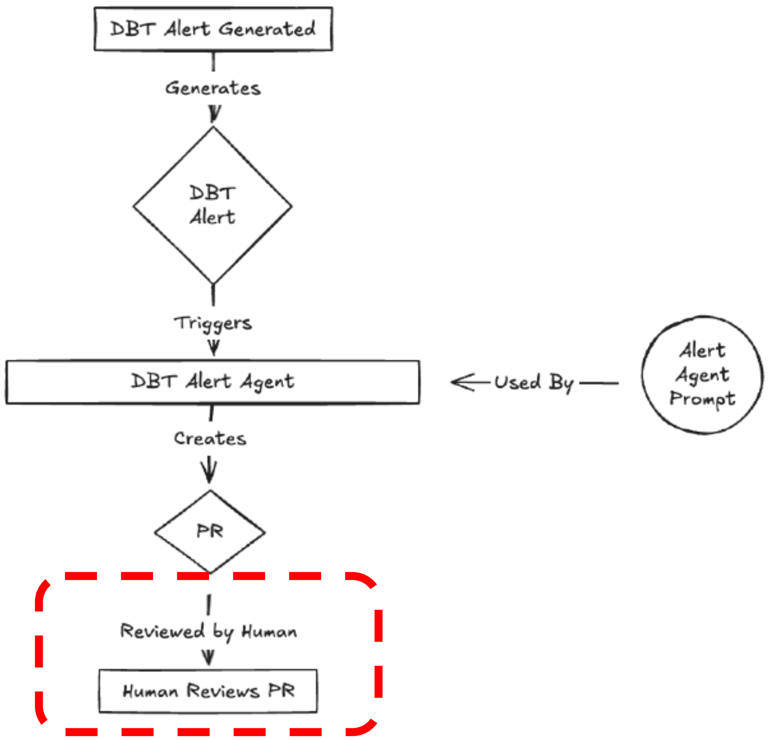


My mandatory tech architecture slide



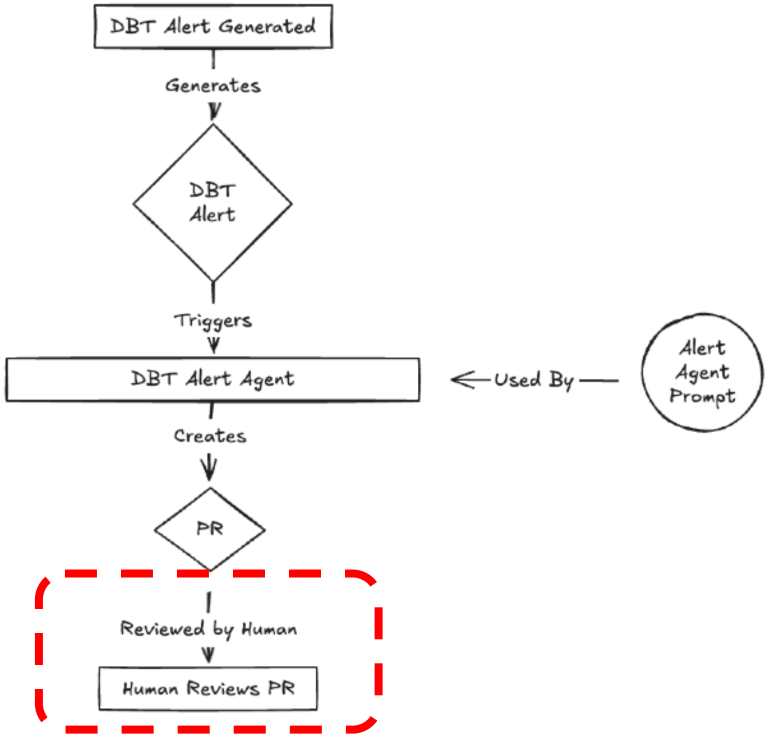
Code generation success requires human review

- Code Syntactics: does the code run?
  - Agent can grade
  - Hardest part is getting LLM to run in "production" environments, especially for legacy systems
- Code Semantics: does the code use good architectural principles?
  - Agent can grade, so long as "architecture" can fit into context (or is in LLM memory)
  - However, agents don't optimize for maintainability
- Business Context: does the code truly solve the problem?
  - Agent only grades against a metric (no test failure)
  - However, you can fix an alert, but not fix root cause



Code generation success still requires humans

- Code Syntactics: does the code run?
  - Agent can grade
  - Hardest part is getting LLM to run in “production” environments, especially for legacy systems
- **Code Semantics: does the code use good architectural principles?**
  - Agent can grade, so long as “architecture” can fit into context (or is in LLM memory)
  - However, agents don’t optimize for maintainability
- **Business Context: does the code truly solve the problem?**
  - Agent only grades against a metric (no test failure)
  - However, you can fix an alert, but not fix root cause



Side Quest: Getting humans to label is harder than you think

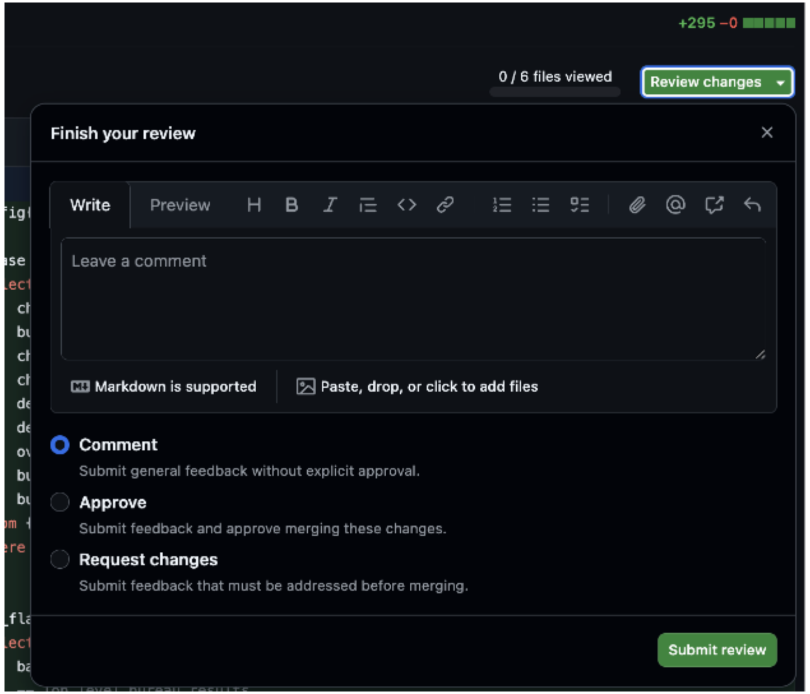
Getting good human review is hard:

- ✔ Easy to get people to label once — hard to get continuous, high-quality labeling
- 🧠 Humans often have important context missing from the model
- 😓 Poor labeling interfaces kill participation

We already have everything we need: code review → high-quality, structured labels

We considered Slack messages and summarization, but they don't produce usable labels

Key takeaway:
🏷️ Labels = Measurement = Feedback Loop



Finish your review

Leave a comment

Markdown is supported

Paste, drop, or click to add files

[✓ Comment]
Submit general feedback without explicit approval.

Approve
Submit feedback and approve merging these changes.

Request changes
Submit feedback that must be addressed before merging.

High-quality labels allows feedback loop to automatically tune prompt

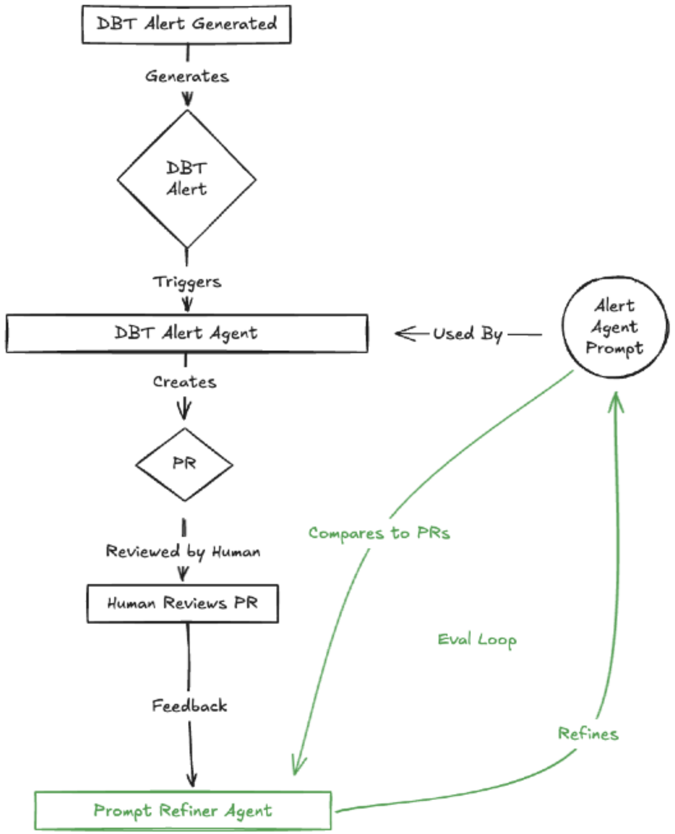
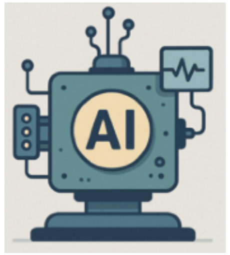


Figure out your tasks to find your feedback loops

- Focus on tasks, not jobs, when you are starting to build your agentic systems
- Good feedback loops are critical to your systems long run success
- Bound scope, set a goal, and define success to generate high quality labels

Figure out your tasks to find your feedback loops

- Focus on tasks, not jobs, when you are starting to build your agentic systems
- Good feedback loops are critical to your systems long run success
- Bound scope, set a goal, and define success to generate high quality labels



System Design >>> Tech