incident.io

BECOMING AI ENGINEERS

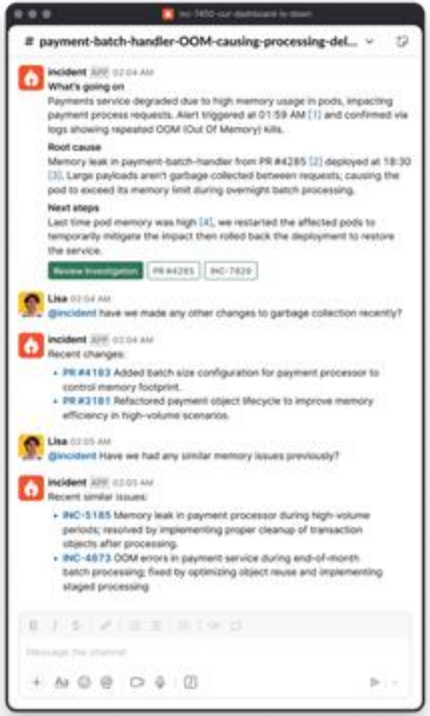
Lawrence Jones
LDX London (June 2025)

Hi!

- → Lawrence Jones (@lawrjones)
- → Product Engineer at incident.io, leading AI efforts
- → Joined three years ago as first hire
- → Now 100 people, dual-HQ in London and SF

The timeline to *fully automated* incident response.

Automated incident investigation and resolution



payment-batch-handler-OOM-causing-processing-del...

incident 02:04 AM

What's going on

Payments service degraded due to high memory usage in pods, impacting payment process requests. Alert triggered at 01:59 AM [1] and confirmed via logs showing repeated OOM (Out Of Memory) kills.

Root cause

Memory leak in payment-batch-handler from PR #4285 [2] deployed at 18:30 [3]. Large payloads aren't garbage collected between requests, causing the pod to exceed its memory limit during overnight batch processing.

Next steps

Last time pod memory was high [4], we restarted the affected pods to temporarily mitigate the impact then rolled back the deployment to restore the service.

Lisa 02:04 AM

@incident have we made any other changes to garbage collection recently?

incident 02:04 AM

Recent changes:

- PR #4183 Added batch size configuration for payment processor to control memory footprint.
- PR #3181 Refactored payment object lifecycle to improve memory efficiency in high-volume scenarios.

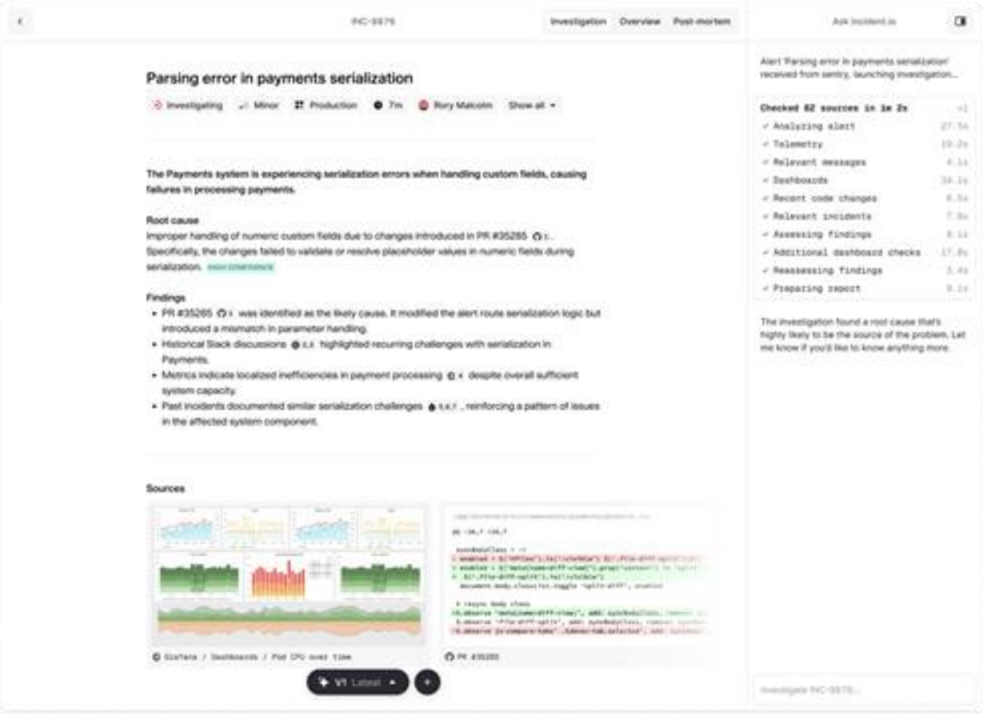
Lisa 02:05 AM

@incident Have we had any similar memory issues previously?

incident 02:05 AM

Recent similar issues:

- INC-5185 Memory leak in payment processor during high-volume periods; resolved by implementing proper cleanup of transaction objects after processing.
- INC-4873 OOM errors in payment service during end-of-month batch processing; fixed by optimizing object reuse and implementing staged processing.



Parsing error in payments serialization

The Payments system is experiencing serialization errors when handling custom fields, causing failures in processing payments.

Root cause

Improper handling of numeric custom fields due to changes introduced in PR #35285. Specifically, the changes failed to validate or resolve placeholder values in numeric fields during serialization.

Findings

- PR #35285 was identified as the likely cause. It modified the alert route serialization logic but introduced a mismatch in parameter handling.
- Historical Slack discussions highlighted recurring challenges with serialization in Payments.
- Metrics indicate localized inefficiencies in payment processing despite overall sufficient system capacity.
- Past incidents documented similar serialization challenges, reinforcing a pattern of issues in the affected system component.

Sources

Alert 'Parsing error in payments serialization' received from sentry, launching investigation...

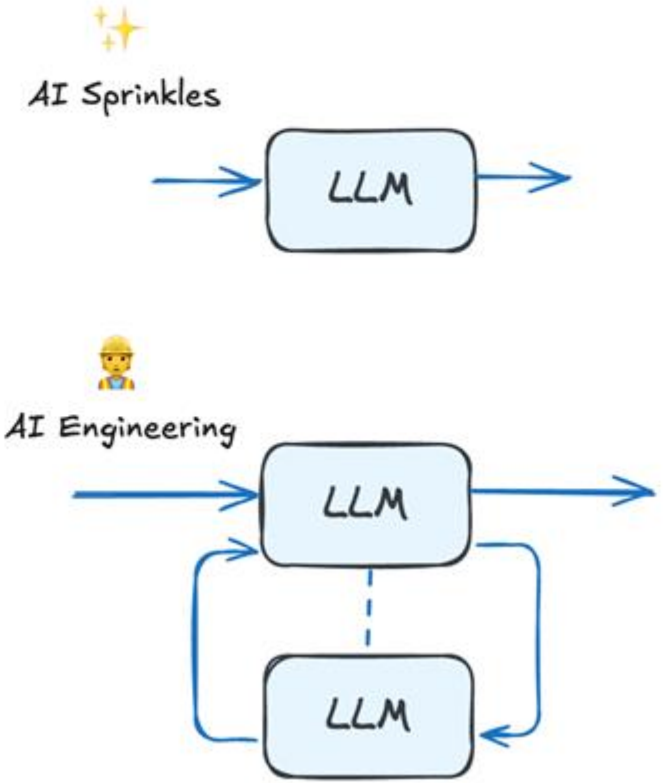
Checked 82 sources in 1m 2s

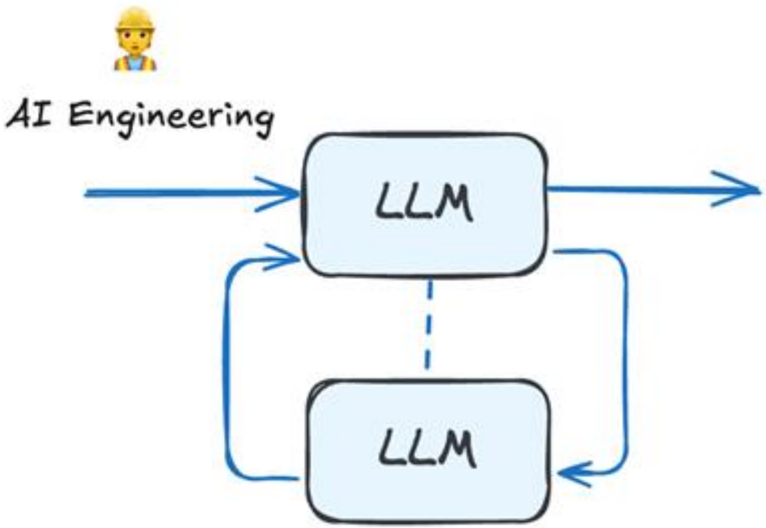
- ✓ Analyzing alert
- ✓ Telemetry
- ✓ Relevant messages
- ✓ Dashboards
- ✓ Recent code changes
- ✓ Relevant incidents
- ✓ Assessing findings
- ✓ Additional dashboard checks
- ✓ Reassessing findings
- ✓ Preparing report

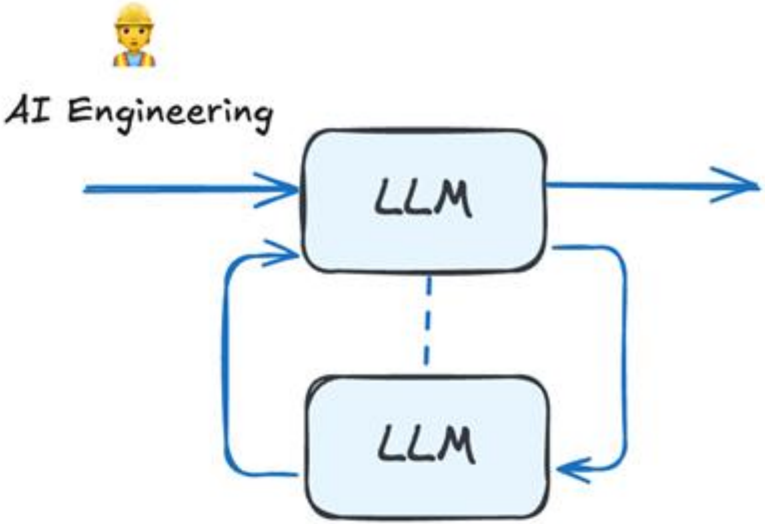
The investigation found a root cause that's highly likely to be the source of the problem. Let me know if you'd like to know anything more.

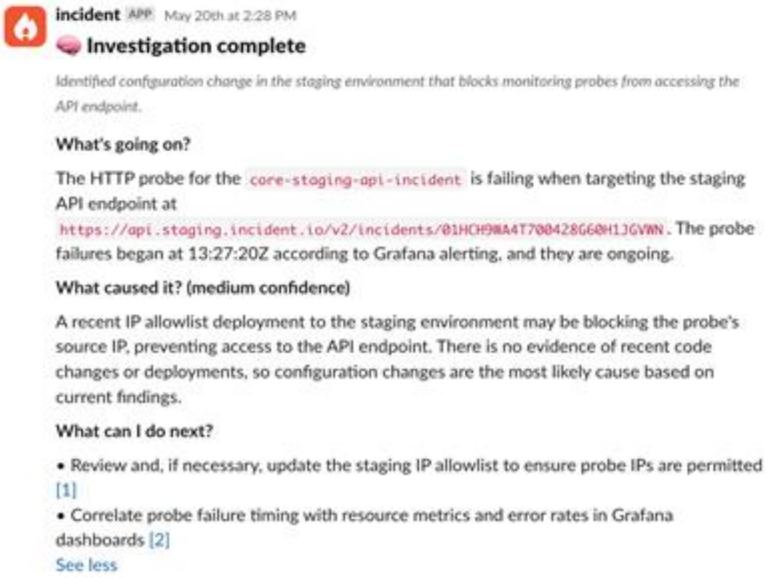
Spectrum of AI in product development

- → No AI
- → AI Sprinkles
- → AI Engineering
- → Foundational model engineering




AI Engineering
LLM
LLM





Investigation complete

Identified configuration change in the staging environment that blocks monitoring probes from accessing the API endpoint.

What's going on?

The HTTP probe for the `core-staging-api-incident` is failing when targeting the staging API endpoint at `https://api.staging.incident.io/v2/incidents/01HCH9MA4T700428G60H1JGVWN`. The probe failures began at 13:27:20Z according to Grafana alerting, and they are ongoing.

What caused it? (medium confidence)

A recent IP allowlist deployment to the staging environment may be blocking the probe's source IP, preventing access to the API endpoint. There is no evidence of recent code changes or deployments, so configuration changes are the most likely cause based on current findings.

What can I do next?

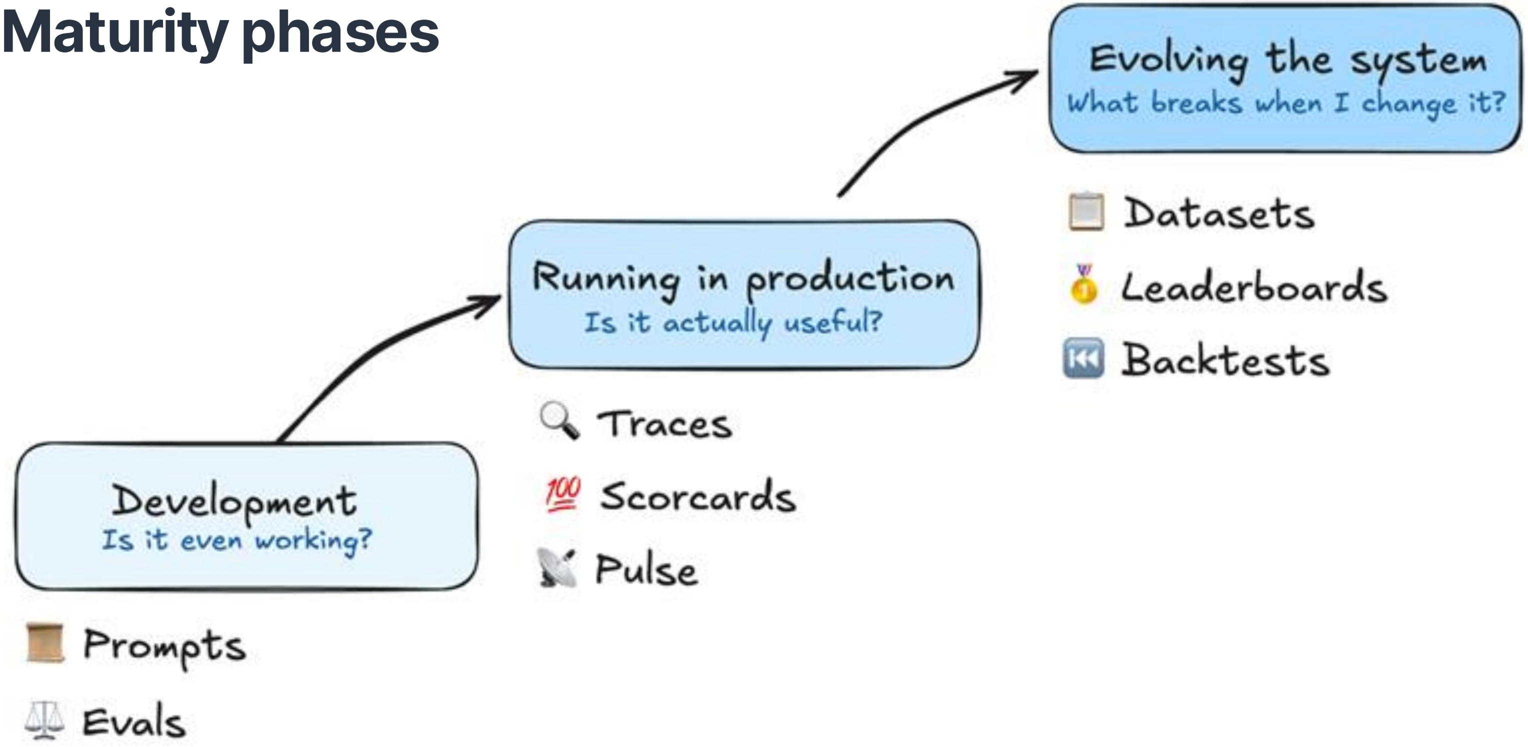
- Review and, if necessary, update the staging IP allowlist to ensure probe IPs are permitted [1]
- Correlate probe failure timing with resource metrics and error rates in Grafana dashboards [2]

😭 **107 LLM calls**

Maturity phases

Development
Running in production
Evolving the system

Maturity phases



Development
Is it even working?

- 📜 Prompts
- ⚖️ Evals

Running in production
Is it actually useful?

- 🔍 Traces
- 💯 Scorcards
- 📡 Pulse

Evolving the system
What breaks when I change it?

- 📋 Datasets
- 🥇 Leaderboards
- ⏮️ Backtests

Maturity phases

Development

Running in production

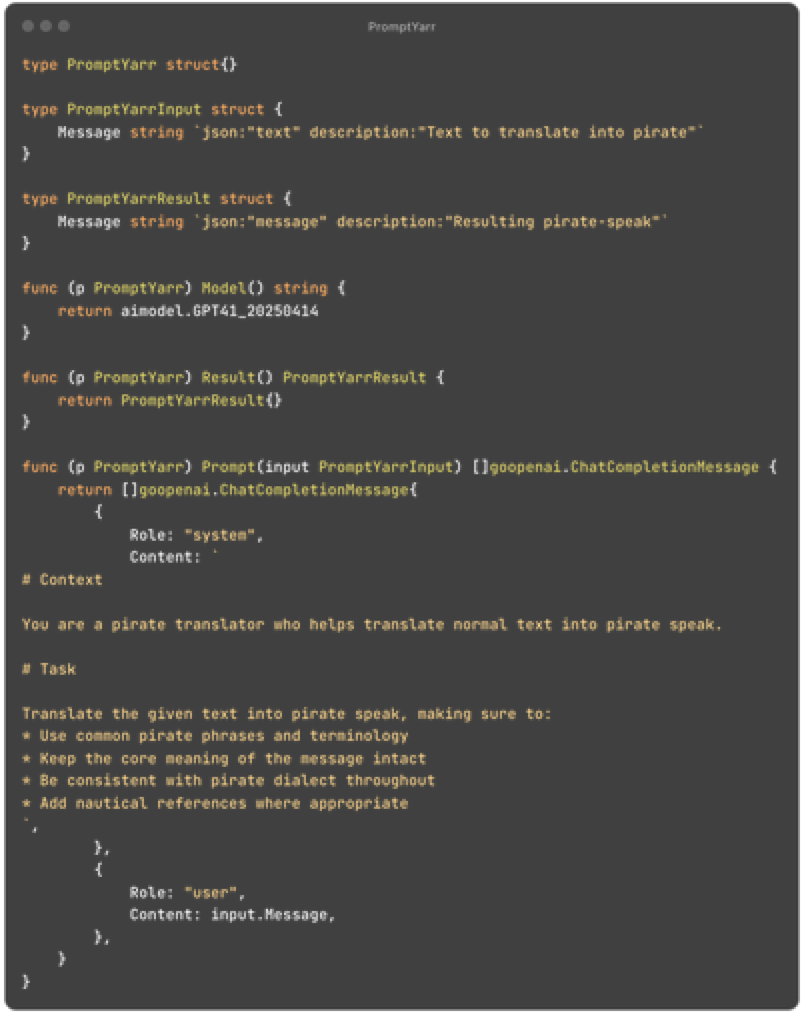
Evolving the system

You need to...

...call LLMs with your data, and receive structured output

Prompt

- ➔ Abstraction around LLM calls
- ➔ Integrate type-system to structured output where possible
- ➔ Support various LLM providers (OpenAI, Anthropic, Google)
- ➔ Apply instrumentation and cost tracking



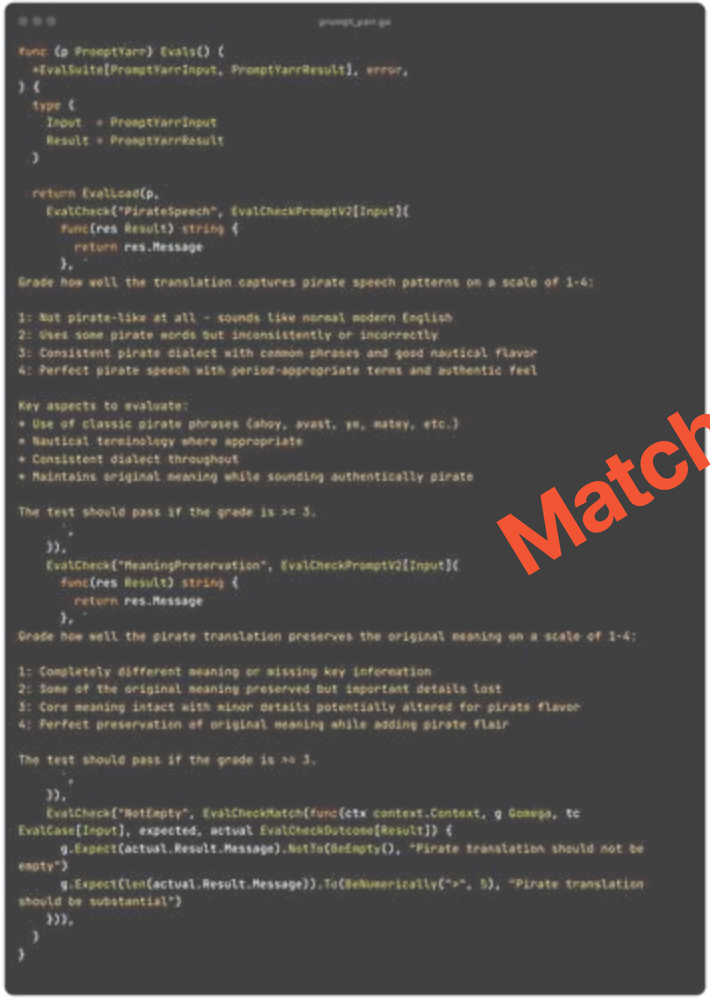
```go
type PromptYarr struct{}

type PromptYarrInput struct {
    Message string `json:"text" description:"Text to translate into pirate"`
}

type PromptYarrResult struct {
    Message string `json:"message" description:"Resulting pirate-speak"`
}

func (p PromptYarr) Model() string {
    return aimodel.GPT41_20250414
}

func (p PromptYarr) Result() PromptYarrResult {
    return PromptYarrResult{}
}

func (p PromptYarr) Prompt(input PromptYarrInput) []goopenai.ChatCompletionMessage {
    return []goopenai.ChatCompletionMessage{
        {
            Role: "system",
            Content: `
# Context

You are a pirate translator who helps translate normal text into pirate speak.

# Task

Translate the given text into pirate speak, making sure to:
* Use common pirate phrases and terminology
* Keep the core meaning of the message intact
* Be consistent with pirate dialect throughout
* Add nautical references where appropriate
`,
        },
        {
            Role: "user",
            Content: input.Message,
        },
    }
}
```

You need to…

…iterate on a prompt to test it works for some set of inputs

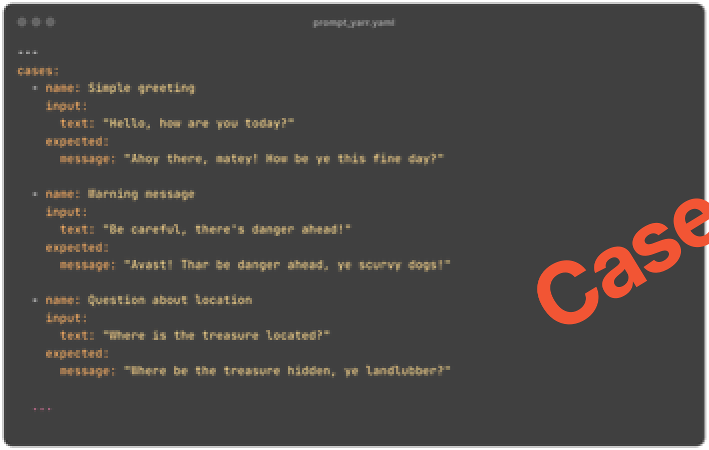
Evals

- ➔ These are LLM ‘unit tests’
- ➔ Each test case has prompt input data and expected result
- ➔ Eval suites define matchers on result and expected
- ➔ Non-determinism is the challenge



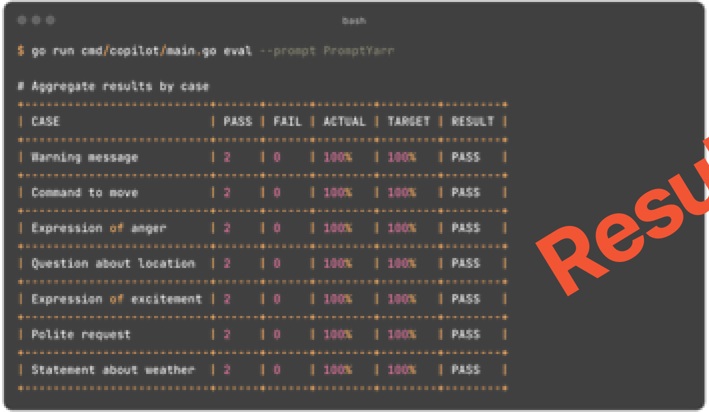
```go
func (p PromptYarr) Evals() (
  *EvalSuite[PromptYarrInput, PromptYarrResult], error,
) {
  type (
    Input  = PromptYarrInput
    Result = PromptYarrResult
  )

  return EvalLoad(p,
    EvalCheck("PirateSpeech", EvalCheckPromptV2[Input](
      func(res Result) string {
        return res.Message
      }, `
Grade how well the translation captures pirate speech patterns on a scale of 1-4:

1: Not pirate-like at all - sounds like normal modern English
2: Uses some pirate words but inconsistently or incorrectly
3: Consistent pirate dialect with common phrases and good nautical flavor
4: Perfect pirate speech with period-appropriate terms and authentic feel

Key aspects to evaluate:
• Use of classic pirate phrases (ahoy, avast, ye, matey, etc.)
• Nautical terminology where appropriate
• Consistent dialect throughout
• Maintains original meaning while sounding authentically pirate

The test should pass if the grade is >= 3.
      `,
    )),
    EvalCheck("MeaningPreservation", EvalCheckPromptV2[Input](
      func(res Result) string {
        return res.Message
      }, `
Grade how well the pirate translation preserves the original meaning on a scale of 1-4:

1: Completely different meaning or missing key information
2: Some of the original meaning preserved but important details lost
3: Core meaning intact with minor details potentially altered for pirate flavor
4: Perfect preservation of original meaning while adding pirate flair

The test should pass if the grade is >= 3.
      `,
    )),
    EvalCheck("NotEmpty", EvalCheckMatch(func(ctx context.Context, g Gomega, tc EvalCase[Input], expected, actual EvalCheckOutcome[Result]) {
      g.Expect(actual.Result.Message).NotTo(BeEmpty(), "Pirate translation should not be empty")
      g.Expect(len(actual.Result.Message)).To(BeNumerically(">", 5), "Pirate translation should be substantial")
    })),
  )
}
```

Matchers



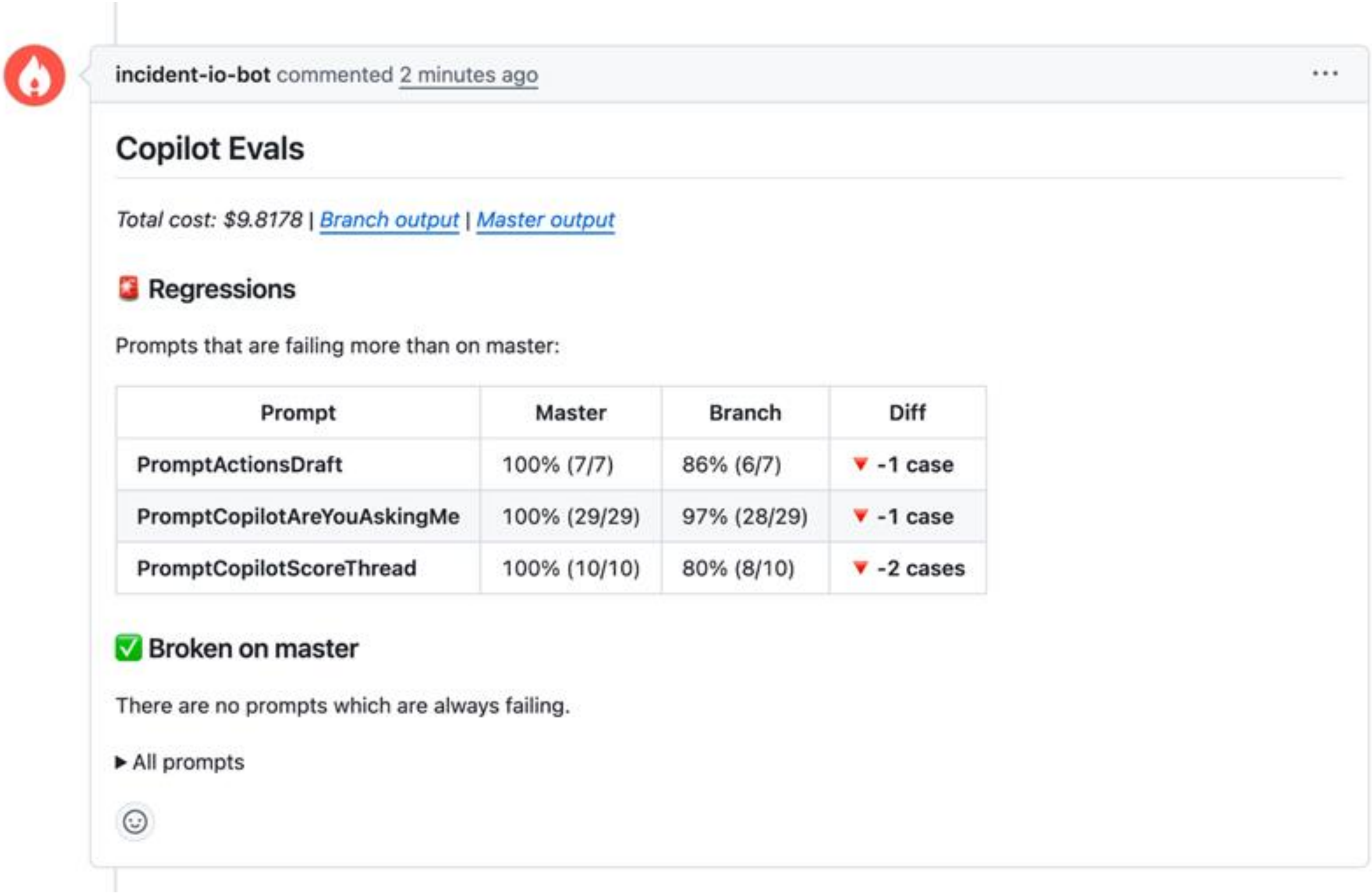
```yaml
---
cases:
  - name: Simple greeting
    input:
      text: "Hello, how are you today?"
    expected:
      message: "Ahoy there, matey! How be ye this fine day?"

  - name: Warning message
    input:
      text: "Be careful, there's danger ahead!"
    expected:
      message: "Avast! Thar be danger ahead, ye scurvy dogs!"

  - name: Question about location
    input:
      text: "Where is the treasure located?"
    expected:
      message: "Where be the treasure hidden, ye landlubber?"

  ...
```

Cases



```
$ go run cmd/copilot/main.go eval --prompt PromptYarr

# Aggregate results by case
+--------------------------+------+------+--------+--------+--------+
| CASE                     | PASS | FAIL | ACTUAL | TARGET | RESULT |
+--------------------------+------+------+--------+--------+--------+
| Warning message          | 2    | 0    | 100%   | 100%   | PASS   |
+--------------------------+------+------+--------+--------+--------+
| Command to move          | 2    | 0    | 100%   | 100%   | PASS   |
+--------------------------+------+------+--------+--------+--------+
| Expression of anger      | 2    | 0    | 100%   | 100%   | PASS   |
+--------------------------+------+------+--------+--------+--------+
| Question about location  | 2    | 0    | 100%   | 100%   | PASS   |
+--------------------------+------+------+--------+--------+--------+
| Expression of excitement | 2    | 0    | 100%   | 100%   | PASS   |
+--------------------------+------+------+--------+--------+--------+
| Polite request           | 2    | 0    | 100%   | 100%   | PASS   |
+--------------------------+------+------+--------+--------+--------+
| Statement about weather  | 2    | 0    | 100%   | 100%   | PASS   |
+--------------------------+------+------+--------+--------+--------+
```

Results



incident-io-bot commented 2 minutes ago

Copilot Evals

Total cost: $9.8178 | *Branch output* | *Master output*

🚨 Regressions

Prompts that are failing more than on master:

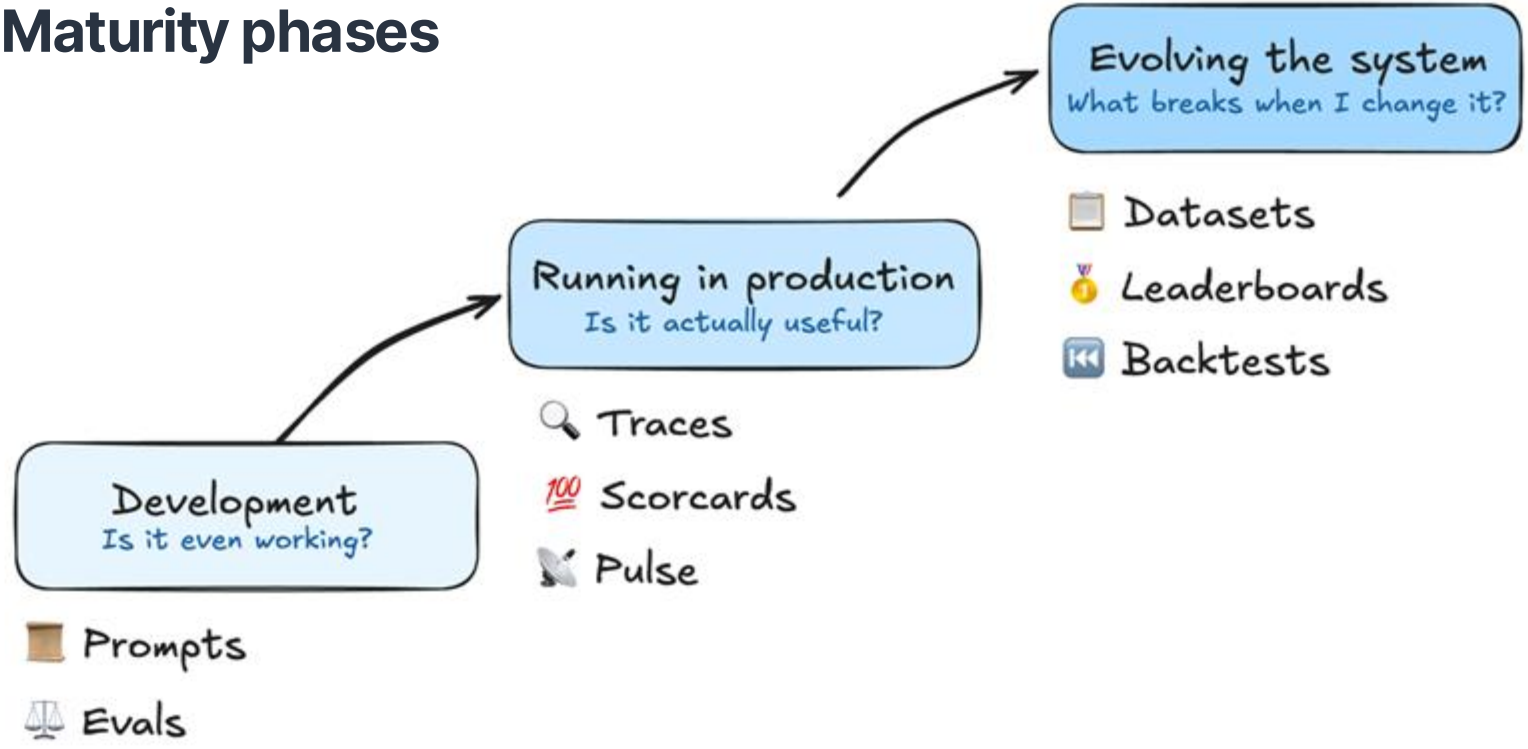
Prompt	Master	Branch	Diff
PromptActionsDraft	100% (7/7)	86% (6/7)	🔻 **-1 case**
PromptCopilotAreYouAskingMe	100% (29/29)	97% (28/29)	🔻 **-1 case**
PromptCopilotScoreThread	100% (10/10)	80% (8/10)	🔻 **-2 cases**

✅ Broken on master

There are no prompts which are always failing.

▶ All prompts

Maturity phases



Development
Is it even working?

- 📜 Prompts
- ⚖️ Evals

Running in production
Is it actually useful?

- 🔍 Traces
- 💯 Scorcards
- 📡 Pulse

Evolving the system
What breaks when I change it?

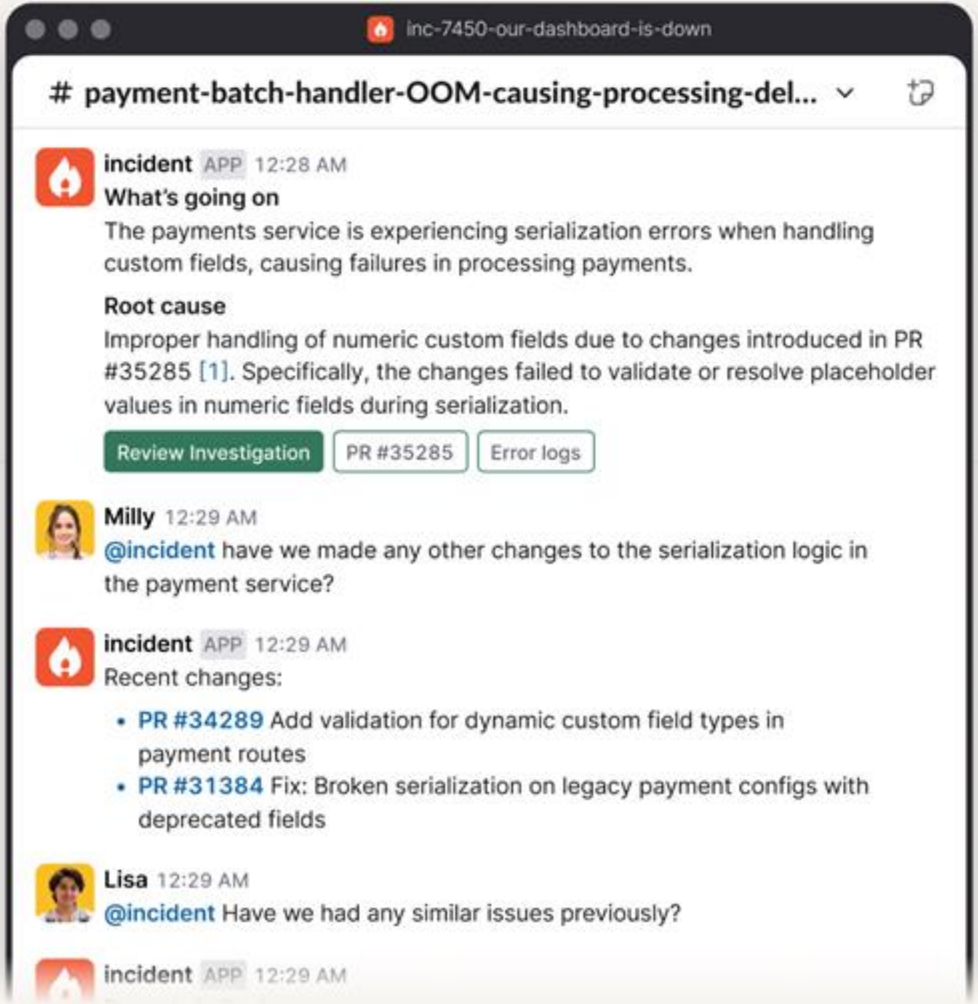
- 📋 Datasets
- 🥇 Leaderboards
- ⏮️ Backtests

Maturity phases

Development

Running in production

Evolving the system


inc-7450-our-dashboard-is-down
# payment-batch-handler-OOM-causing-processing-del...

incident APP 12:28 AM

What's going on

The payments service is experiencing serialization errors when handling custom fields, causing failures in processing payments.

Root cause

Improper handling of numeric custom fields due to changes introduced in PR #35285 [1]. Specifically, the changes failed to validate or resolve placeholder values in numeric fields during serialization.

Milly 12:29 AM

@incident have we made any other changes to the serialization logic in the payment service?

incident APP 12:29 AM

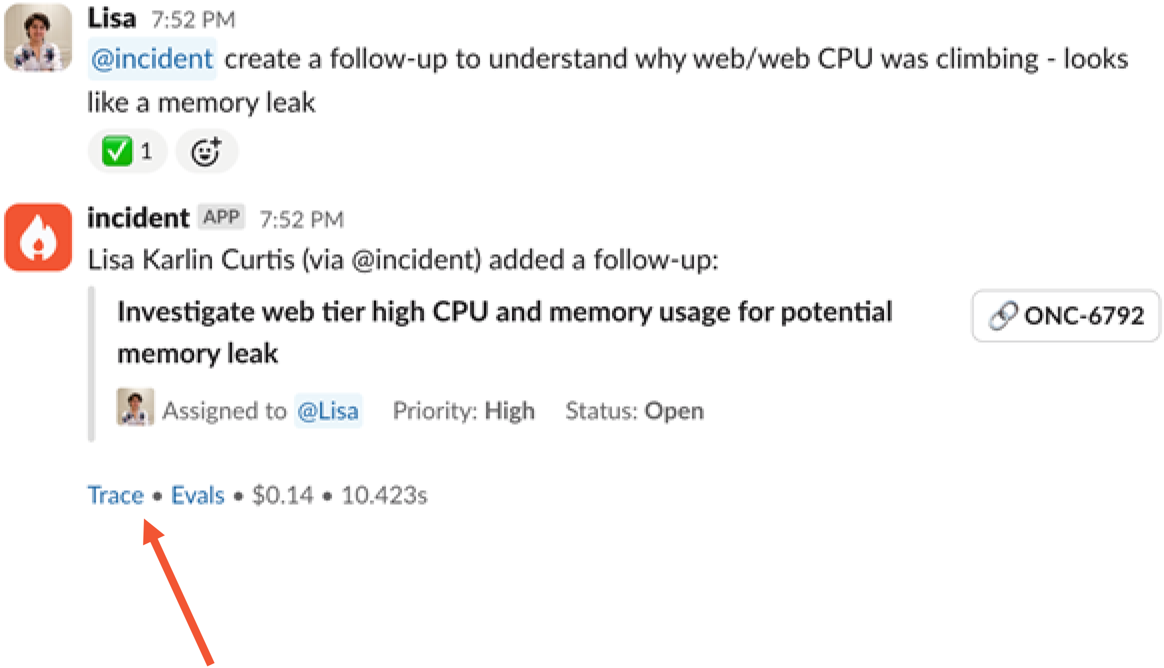
Recent changes:

- PR #34289 Add validation for dynamic custom field types in payment routes
- PR #31384 Fix: Broken serialization on legacy payment configs with deprecated fields

Lisa 12:29 AM

@incident Have we had any similar issues previously?

incident APP 12:29 AM



Lisa 7:52 PM

@incident create a follow-up to understand why web/web CPU was climbing - looks like a memory leak

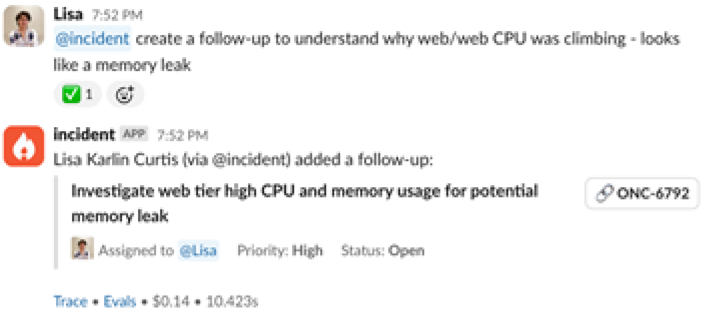
✅ 1

incident APP 7:52 PM

Lisa Karlin Curtis (via @incident) added a follow-up:

> **Investigate web tier high CPU and memory usage for potential memory leak**
>
> 🔗 ONC-6792
>
> Assigned to @Lisa Priority: **High** Status: **Open**

Trace • Evals • $0.14 • 10.423s

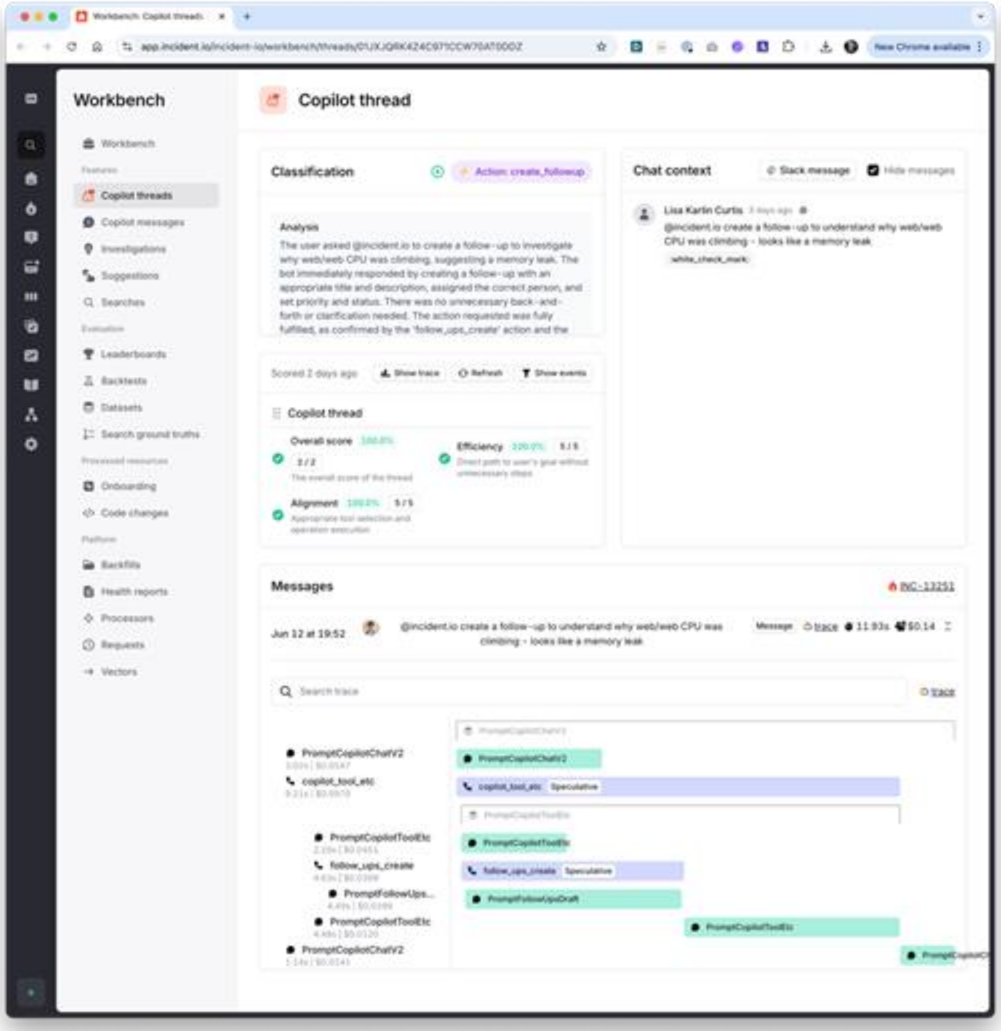


Lisa 7:52 PM
@incident create a follow-up to understand why web/web CPU was climbing - looks like a memory leak

incident APP 7:52 PM
Lisa Karlin Curtis (via @incident) added a follow-up:

> **Investigate web tier high CPU and memory usage for potential memory leak**
> Assigned to @Lisa   Priority: **High**   Status: **Open**

Trace • Evals • $0.14 • 10.423s



Analysis
The user asked @incident.io to create a follow-up to investigate why web/web CPU was climbing, suggesting a memory leak. The bot immediately responded by creating a follow-up with an appropriate title and description, assigned the correct person, and set priority and status. There was no unnecessary back-and-forth or clarification needed. The action requested was fully fulfilled, as confirmed by the 'follow_ups_create' action and the

Lisa Karlin Curtis
@incident.io create a follow-up to understand why web/web CPU was climbing - looks like a memory leak
white_check_mark

Jun 12 at 19:52 @incident.io create a follow-up to understand why web/web CPU was climbing - looks like a memory leak

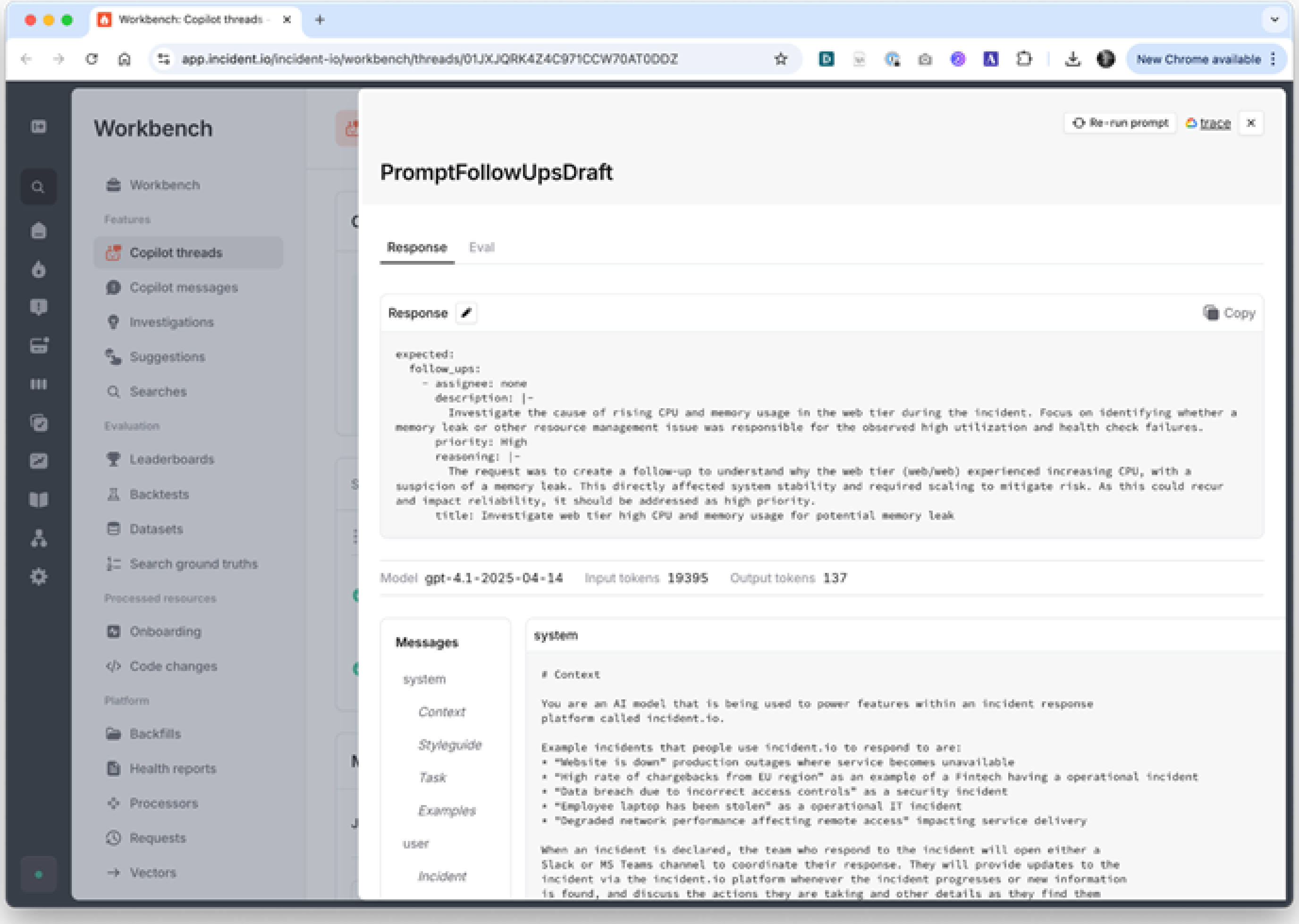
You need to...

...see exactly what we sent and received, in order to debug

Traces

- ➔ Not standard o11y traces, these contain rich data
- ➔ See exactly what was sent and received from the model
- ➔ Allow you to re-run and compare models
- ➔ ‘Steal’ a trace for an eval

Inspect



PromptFollowUpsDraft

Response

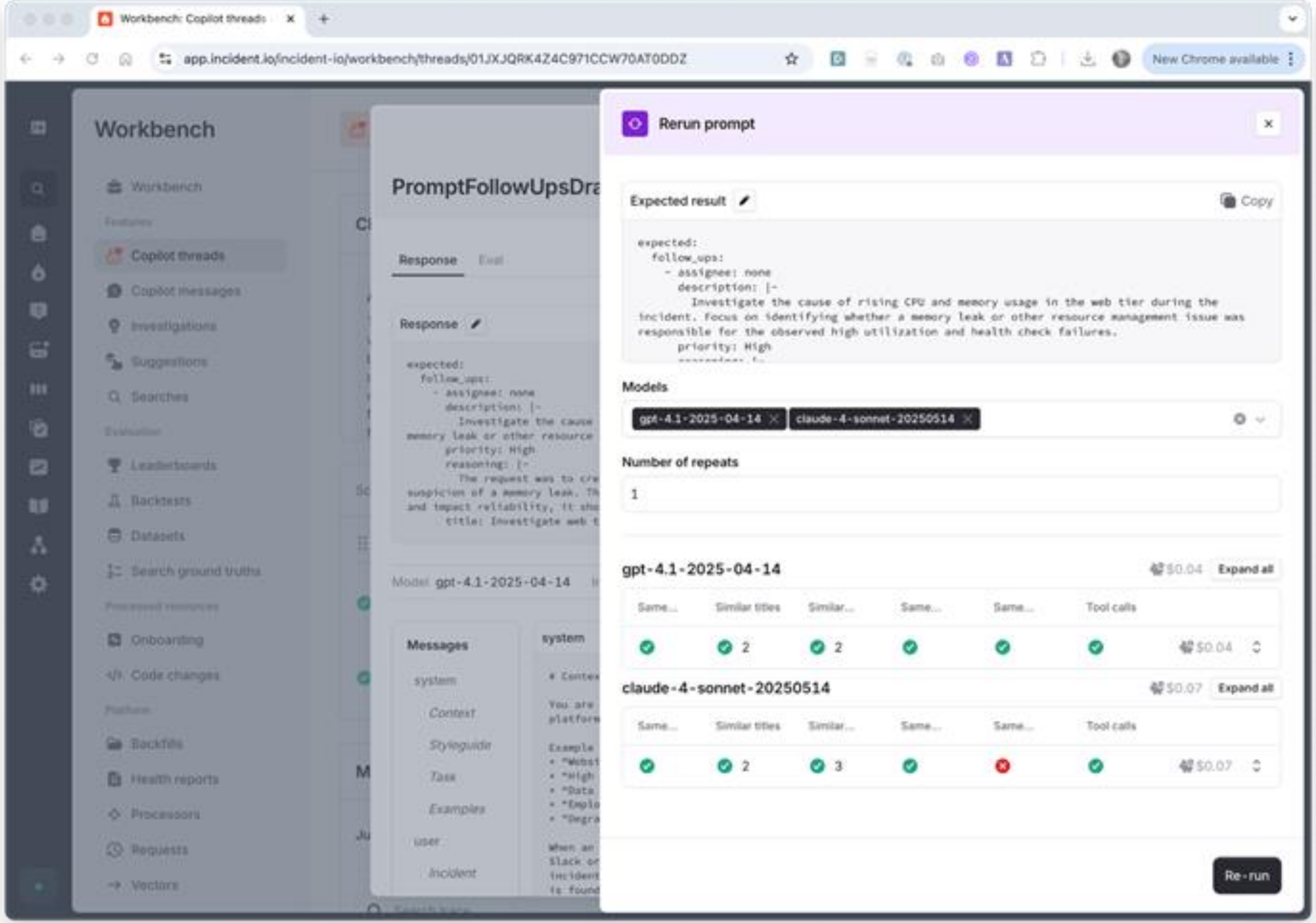
```
expected:
  follow_ups:
    - assignee: none
      description: |-
        Investigate the cause of rising CPU and memory usage in the web tier during the incident. Focus on identifying whether a
memory leak or other resource management issue was responsible for the observed high utilization and health check failures.
      priority: High
      reasoning: |-
        The request was to create a follow-up to understand why the web tier (web/web) experienced increasing CPU, with a
suspicion of a memory leak. This directly affected system stability and required scaling to mitigate risk. As this could recur
and impact reliability, it should be addressed as high priority.
      title: Investigate web tier high CPU and memory usage for potential memory leak
```

Model gpt-4.1-2025-04-14 Input tokens 19395 Output tokens 137

system

```
# Context

You are an AI model that is being used to power features within an incident response
platform called incident.io.

Example incidents that people use incident.io to respond to are:
• "Website is down" production outages where service becomes unavailable
• "High rate of chargebacks from EU region" as an example of a Fintech having a operational incident
• "Data breach due to incorrect access controls" as a security incident
• "Employee laptop has been stolen" as a operational IT incident
• "Degraded network performance affecting remote access" impacting service delivery

When an incident is declared, the team who respond to the incident will open either a
Slack or MS Teams channel to coordinate their response. They will provide updates to the
incident via the incident.io platform whenever the incident progresses or new information
is found, and discuss the actions they are taking and other details as they find them
```

Re-run



Rerun prompt

Expected result

```
expected:
  follow_ups:
    - assignee: none
      description: |-
        Investigate the cause of rising CPU and memory usage in the web tier during the
incident. Focus on identifying whether a memory leak or other resource management issue was
responsible for the observed high utilization and health check failures.
      priority: High
```

Models: gpt-4.1-2025-04-14, claude-4-sonnet-20250514

Number of repeats: 1

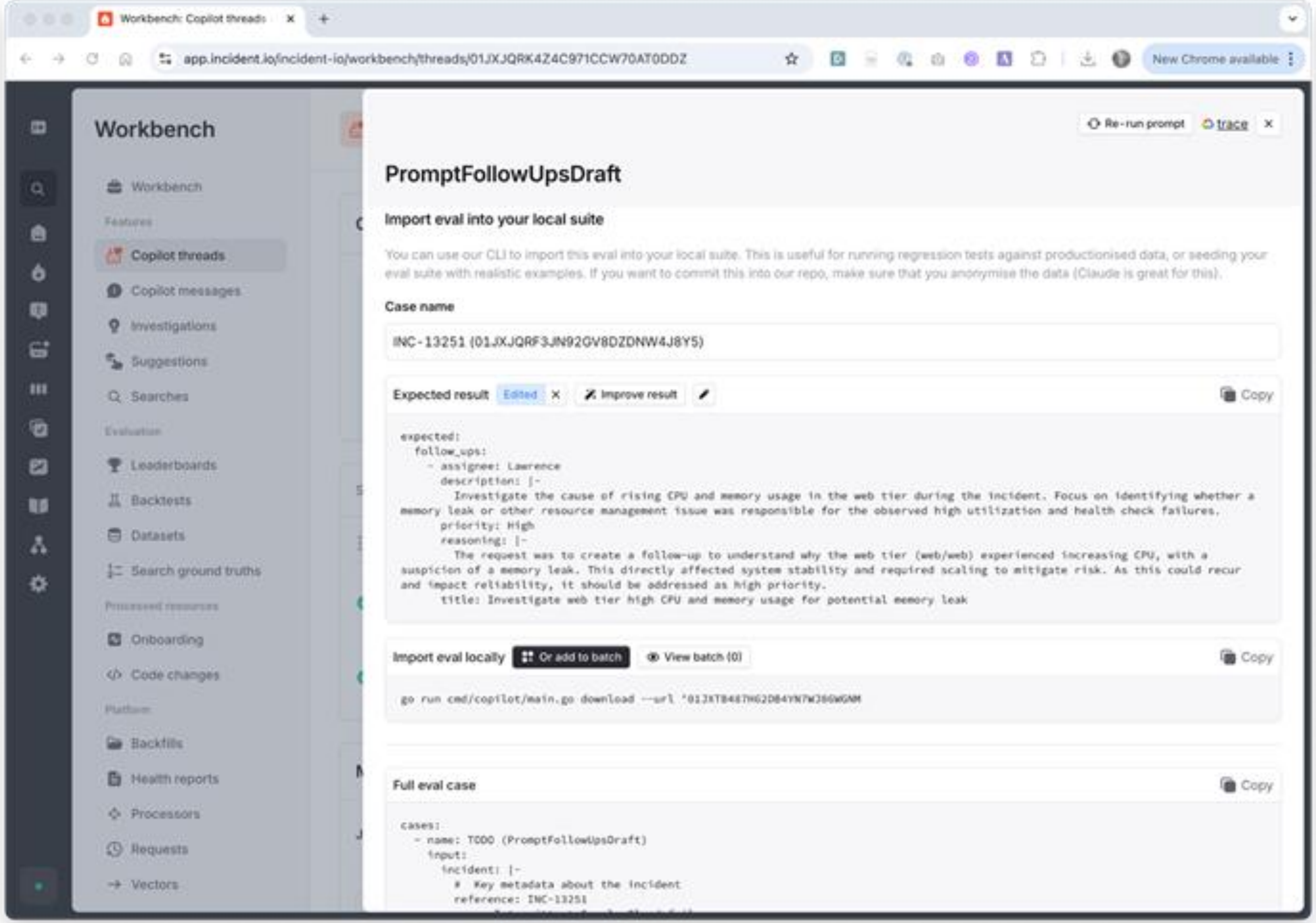
gpt-4.1-2025-04-14 $0.04

Same...	Similar titles	Similar...	Same...	Same...	Tool calls	
✓	✓ 2	✓ 2	✓	✓	✓	$0.04

claude-4-sonnet-20250514 $0.07

Same...	Similar titles	Similar...	Same...	Same...	Tool calls	
✓	✓ 2	✓ 3	✓	✗	✓	$0.07

'Steal' an eval



PromptFollowUpsDraft

Import eval into your local suite

You can use our CLI to import this eval into your local suite. This is useful for running regression tests against productionised data, or seeding your eval suite with realistic examples. If you want to commit this into our repo, make sure that you anonymise the data (Claude is great for this).

Case name

INC-13251 (01JXJQRF3JN92GV8DZDNW4J8Y5)

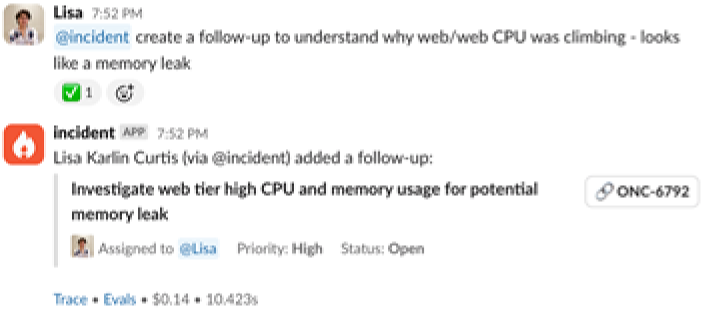
```
expected:
  follow_ups:
    - assignee: Lawrence
      description: |-
        Investigate the cause of rising CPU and memory usage in the web tier during the incident. Focus on identifying whether a memory leak or other resource management issue was responsible for the observed high utilization and health check failures.
      priority: High
      reasoning: |-
        The request was to create a follow-up to understand why the web tier (web/web) experienced increasing CPU, with a suspicion of a memory leak. This directly affected system stability and required scaling to mitigate risk. As this could recur and impact reliability, it should be addressed as high priority.
      title: Investigate web tier high CPU and memory usage for potential memory leak
```

```
go run cmd/copilot/main.go download --url '01JXTB487HG2DB4YN7WJ86GWGNM
```

```
cases:
  - name: TODO (PromptFollowUpsDraft)
    input:
      incident: |-
        #  Key metadata about the incident
        reference: INC-13251
```

You need to...

...score AI interactions, marking them as good/bad and why



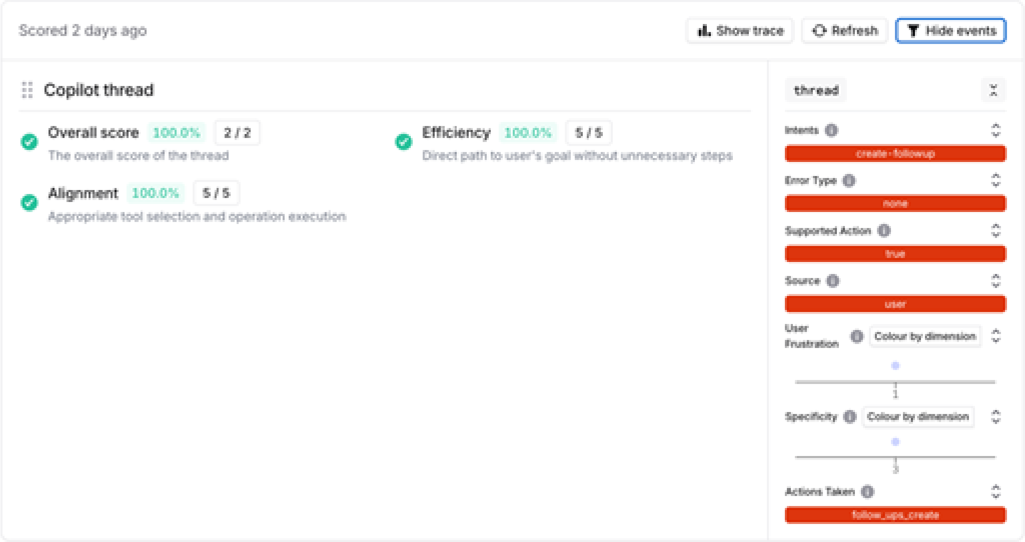
Lisa 7:52 PM
@incident create a follow-up to understand why web/web CPU was climbing - looks like a memory leak

✅ 1

incident APP 7:52 PM
Lisa Karlin Curtis (via @incident) added a follow-up:

> **Investigate web tier high CPU and memory usage for potential memory leak**
> Assigned to @Lisa Priority: High Status: Open

Trace • Evals • $0.14 • 10.423s



Scored 2 days ago

Copilot thread

✓ Overall score 100.0% 2 / 2
The overall score of the thread

✓ Efficiency 100.0% 5 / 5
Direct path to user's goal without unnecessary steps

✓ Alignment 100.0% 5 / 5
Appropriate tool selection and operation execution

thread

Intents: create-followup

Error Type: none

Supported Action: true

Source: user

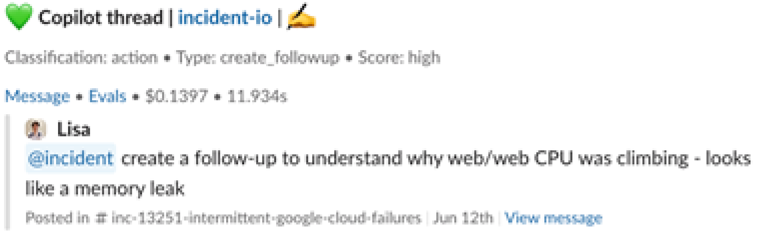
User Frustration: Colour by dimension
1

Specificity: Colour by dimension
3

Actions Taken: follow_ups_create

This is our first objective definition of ‘good’ for AI interactions

#copilot-interactions-pulse



💚 Copilot thread | incident-io | ✍️

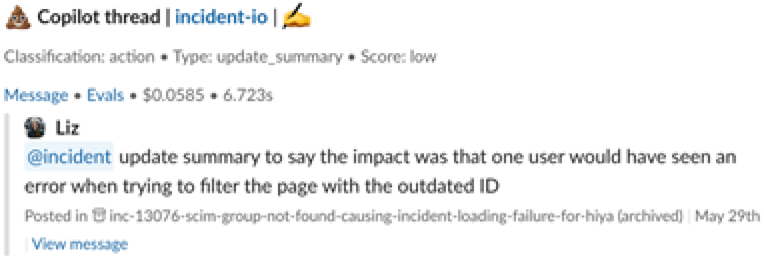
Classification: action • Type: create_followup • Score: high

Message • Evals • $0.1397 • 11.934s

Lisa

@incident create a follow-up to understand why web/web CPU was climbing - looks like a memory leak

Posted in # inc-13251-intermittent-google-cloud-failures | Jun 12th | View message



💩 Copilot thread | incident-io | ✍️

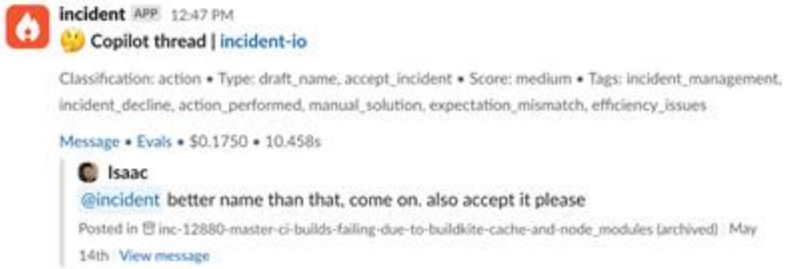
Classification: action • Type: update_summary • Score: low

Message • Evals • $0.0585 • 6.723s

Liz

@incident update summary to say the impact was that one user would have seen an error when trying to filter the page with the outdated ID

Posted in inc-13076-scim-group-not-found-causing-incident-loading-failure-for-hiya (archived) | May 29th | View message



incident APP 12:47 PM

🤔 Copilot thread | incident-io

Classification: action • Type: draft_name, accept_incident • Score: medium • Tags: incident_management, incident_decline, action_performed, manual_solution, expectation_mismatch, efficiency_issues

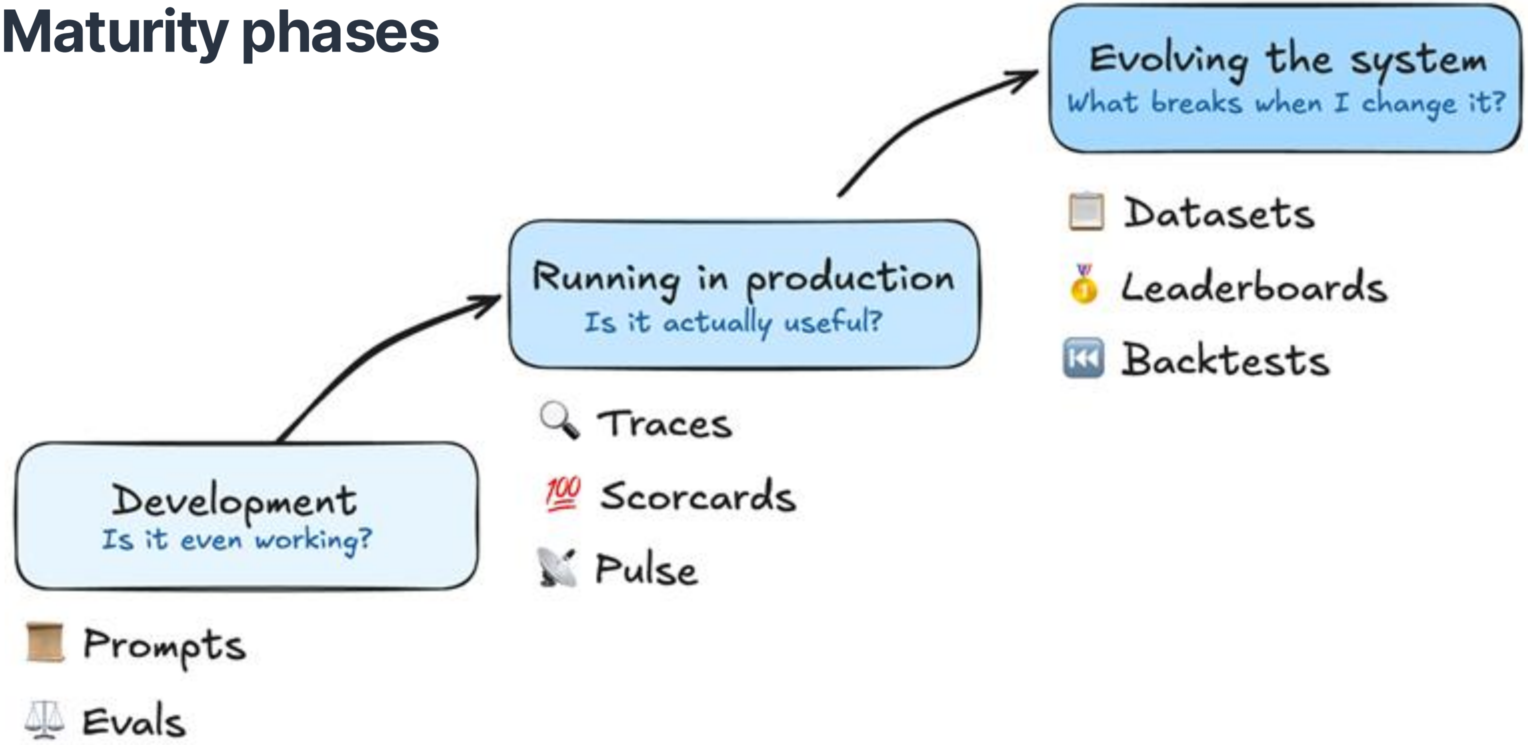
Message • Evals • $0.1750 • 10.458s

Isaac

@incident better name than that, come on. also accept it please

Posted in inc-12880-master-ci-builds-failing-due-to-buildkite-cache-and-node_modules (archived) | May 14th | View message

Maturity phases



Development
Is it even working?

- 📜 Prompts
- ⚖️ Evals

Running in production
Is it actually useful?

- 🔍 Traces
- 💯 Scorcards
- 📡 Pulse

Evolving the system
What breaks when I change it?

- 📋 Datasets
- 🥇 Leaderboards
- ⏮️ Backtests

Maturity phases

Development

Running in production

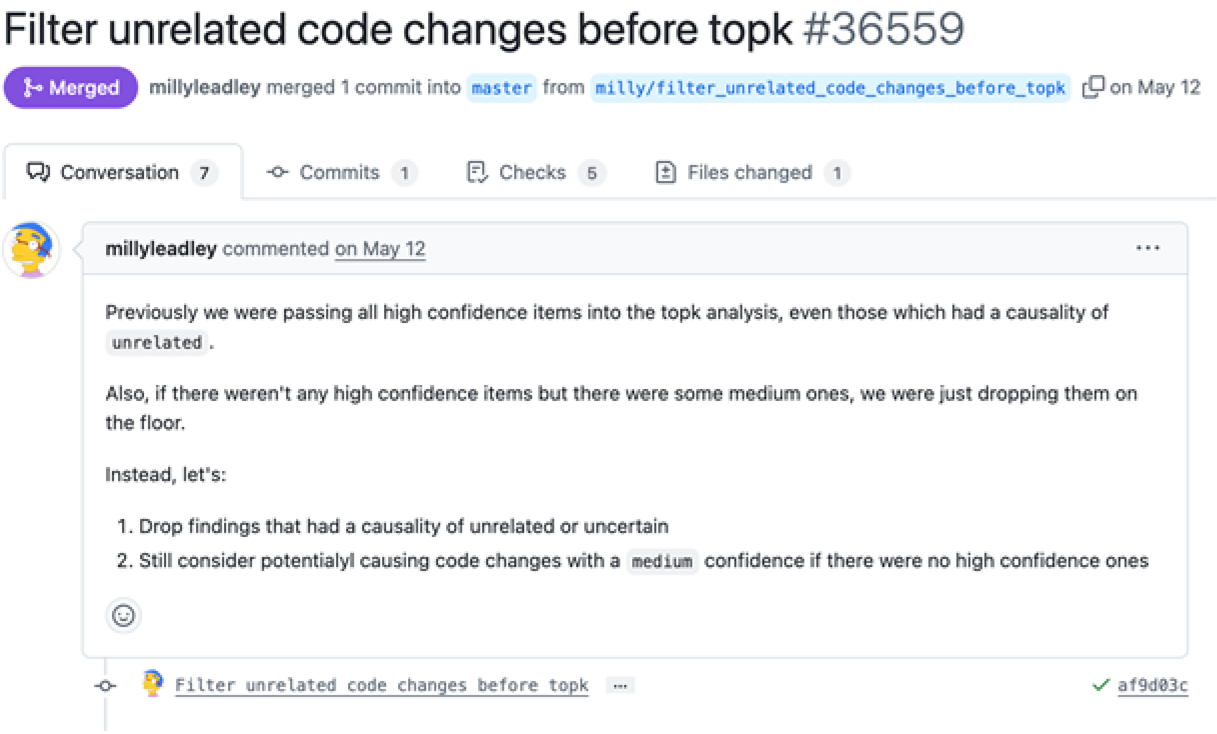
Evolving the system

You need to...

...make a change and know how it's impacted the system for real customers

Backtests

- ➔ Datasets + Scorecards = Backtests
- ➔ Re-run interactions, score them, aggregate scores
- ➔ Make a change and see in ~15m how it changes dataset scores
- ➔ This is how you avoid terrible regressions



Filter unrelated code changes before topk #36559

millyleadley merged 1 commit into `master` from `milly/filter_unrelated_code_changes_before_topk` on May 12

millyleadley commented on May 12

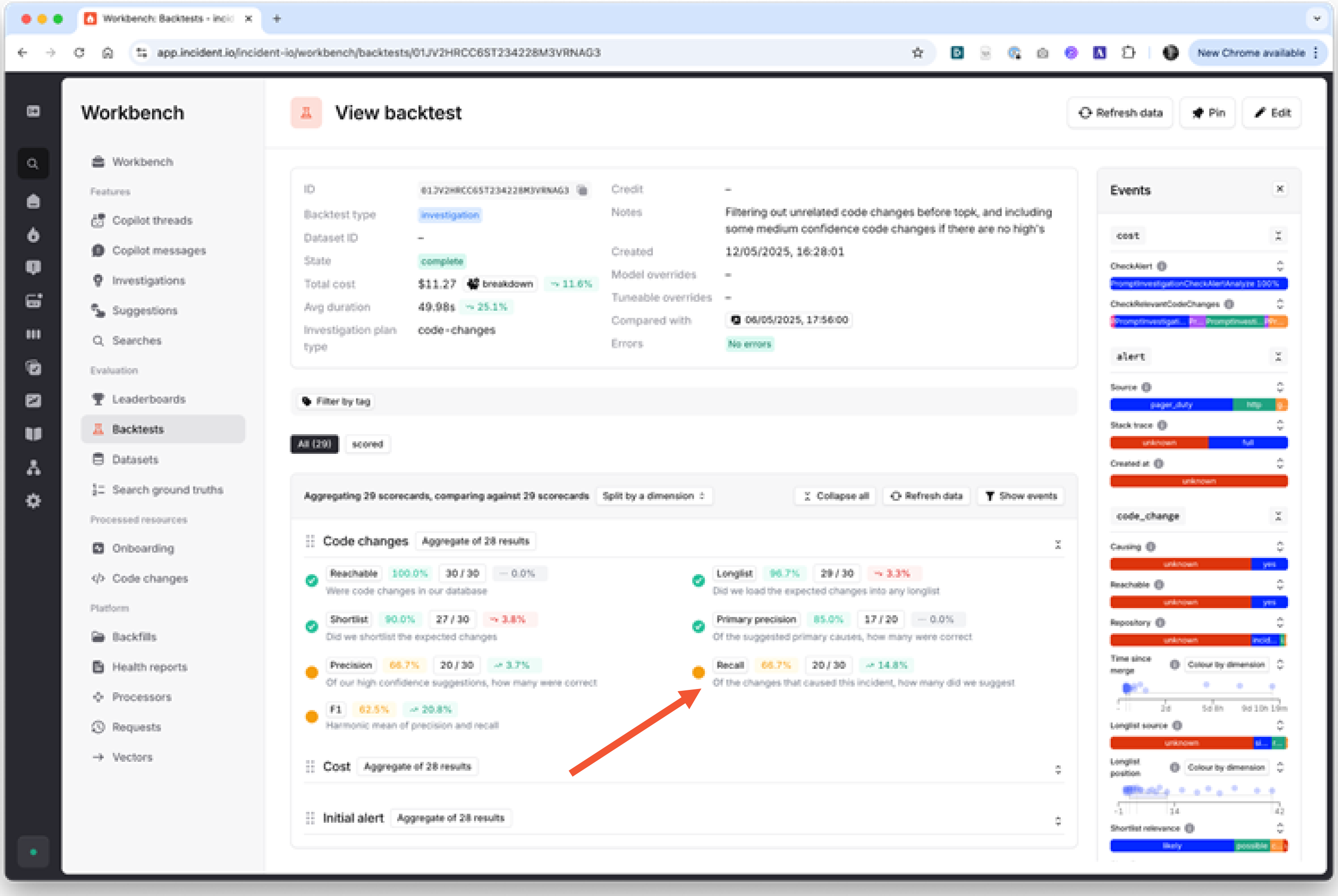
Previously we were passing all high confidence items into the topk analysis, even those which had a causality of `unrelated`.

Also, if there weren't any high confidence items but there were some medium ones, we were just dropping them on the floor.

Instead, let's:

1. Drop findings that had a causality of unrelated or uncertain
2. Still consider potentialyl causing code changes with a `medium` confidence if there were no high confidence ones

`Filter unrelated code changes before topk` … ✓ af9d03c



View backtest

ID: 01JV2HRCC6ST234228M3VRNAG3
Backtest type: investigation
Dataset ID: –
State: complete
Total cost: $11.27 breakdown ↘ 11.6%
Avg duration: 49.98s ↘ 25.1%
Investigation plan type: code-changes

Credit: –
Notes: Filtering out unrelated code changes before topk, and including some medium confidence code changes if there are no high's
Created: 12/05/2025, 16:28:01
Model overrides: –
Tuneable overrides: –
Compared with: 06/05/2025, 17:56:00
Errors: No errors

Aggregating 29 scorecards, comparing against 29 scorecards

Code changes Aggregate of 28 results

✓ Reachable 100.0% 30 / 30 — 0.0%
Were code changes in our database

✓ Longlist 96.7% 29 / 30 ↘ 3.3%
Did we load the expected changes into any longlist

✓ Shortlist 90.0% 27 / 30 ↘ 3.8%
Did we shortlist the expected changes

✓ Primary precision 85.0% 17 / 20 — 0.0%
Of the suggested primary causes, how many were correct

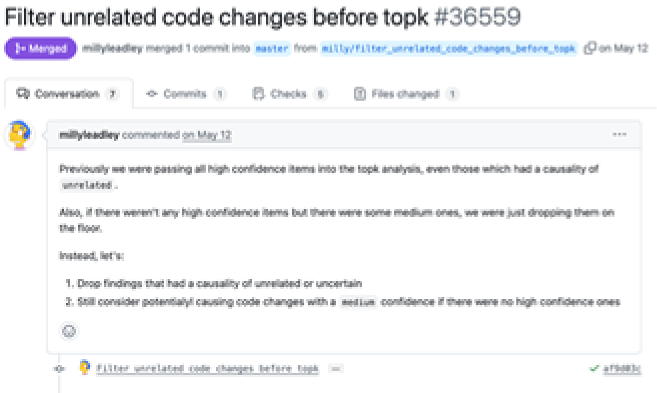
● Precision 66.7% 20 / 30 ↗ 3.7%
Of our high confidence suggestions, how many were correct

● Recall 66.7% 20 / 30 ↗ 14.8%
Of the changes that caused this incident, how many did we suggest

● F1 62.5% ↗ 20.8%
Harmonic mean of precision and recall

Cost Aggregate of 28 results

Initial alert Aggregate of 28 results



Filter unrelated code changes before topk #36559

Merged millyleadley merged 1 commit into `master` from `milly/filter_unrelated_code_changes_before_topk` on May 12

millyleadley commented on May 12

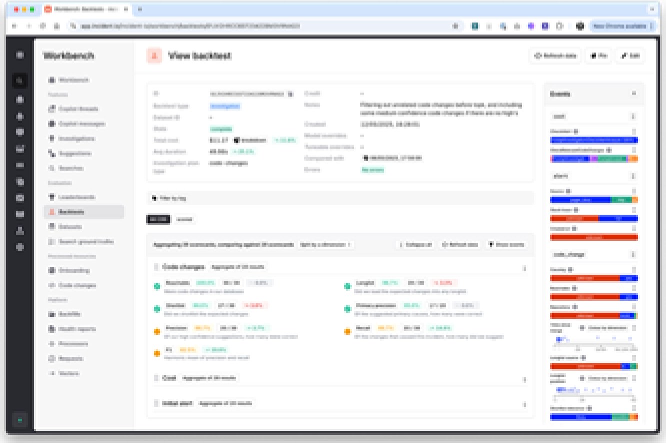
Previously we were passing all high confidence items into the topk analysis, even those which had a causality of `unrelated`.

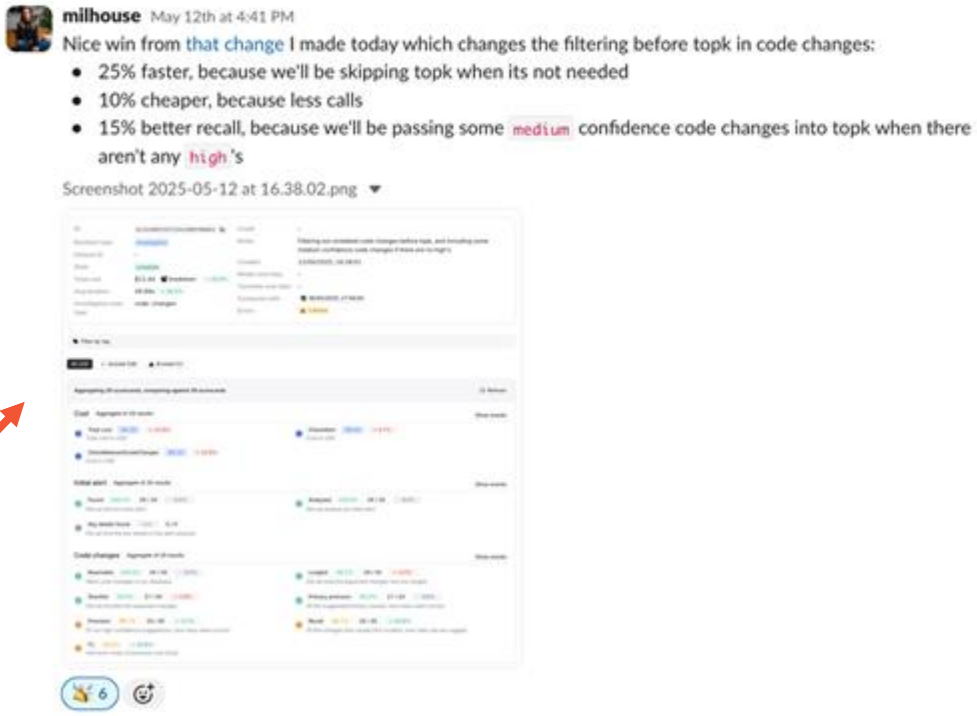
Also, if there weren't any high confidence items but there were some medium ones, we were just dropping them on the floor.

Instead, let's:

1. Drop findings that had a causality of unrelated or uncertain
2. Still consider potentialyl causing code changes with a `medium` confidence if there were no high confidence ones

`Filter unrelated code changes before topk` ✓ af9d83c





milhouse May 12th at 4:41 PM

Nice win from that change I made today which changes the filtering before topk in code changes:

- 25% faster, because we'll be skipping topk when its not needed
- 10% cheaper, because less calls
- 15% better recall, because we'll be passing some `medium` confidence code changes into topk when there aren't any `high`'s

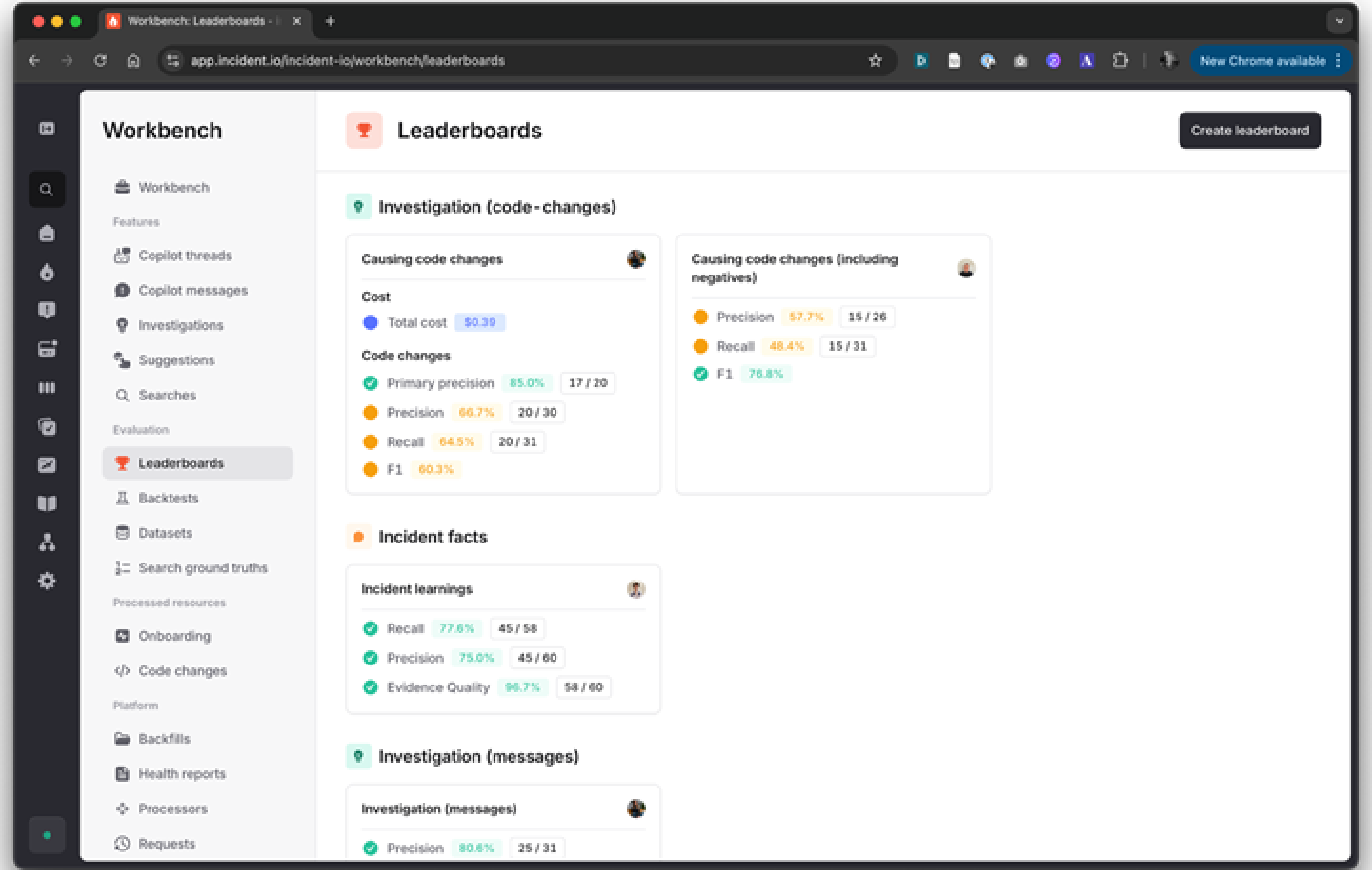
Screenshot 2025-05-12 at 16.38.02.png

You need to...

...track existing baselines and motivate people to hit ‘high scores’

Leaderboards

- ➔ Evaluation criteria, baseline, hill-climb
- ➔ Pin the best scorecard to a leaderboard
- ➔ This is how you reestablish the dopamine hit
- ➔ Make your team feel good when they succeed!



Investigation (code-changes)

Causing code changes

Cost

● Total cost $0.39

Code changes

✓ Primary precision 85.0% 17 / 20

● Precision 66.7% 20 / 30

● Recall 64.5% 20 / 31

● F1 60.3%

Causing code changes (including negatives)

● Precision 57.7% 15 / 26

● Recall 48.4% 15 / 31

✓ F1 76.8%

Incident facts

Incident learnings

✓ Recall 77.6% 45 / 58

✓ Precision 75.0% 45 / 60

✓ Evidence Quality 96.7% 58 / 60

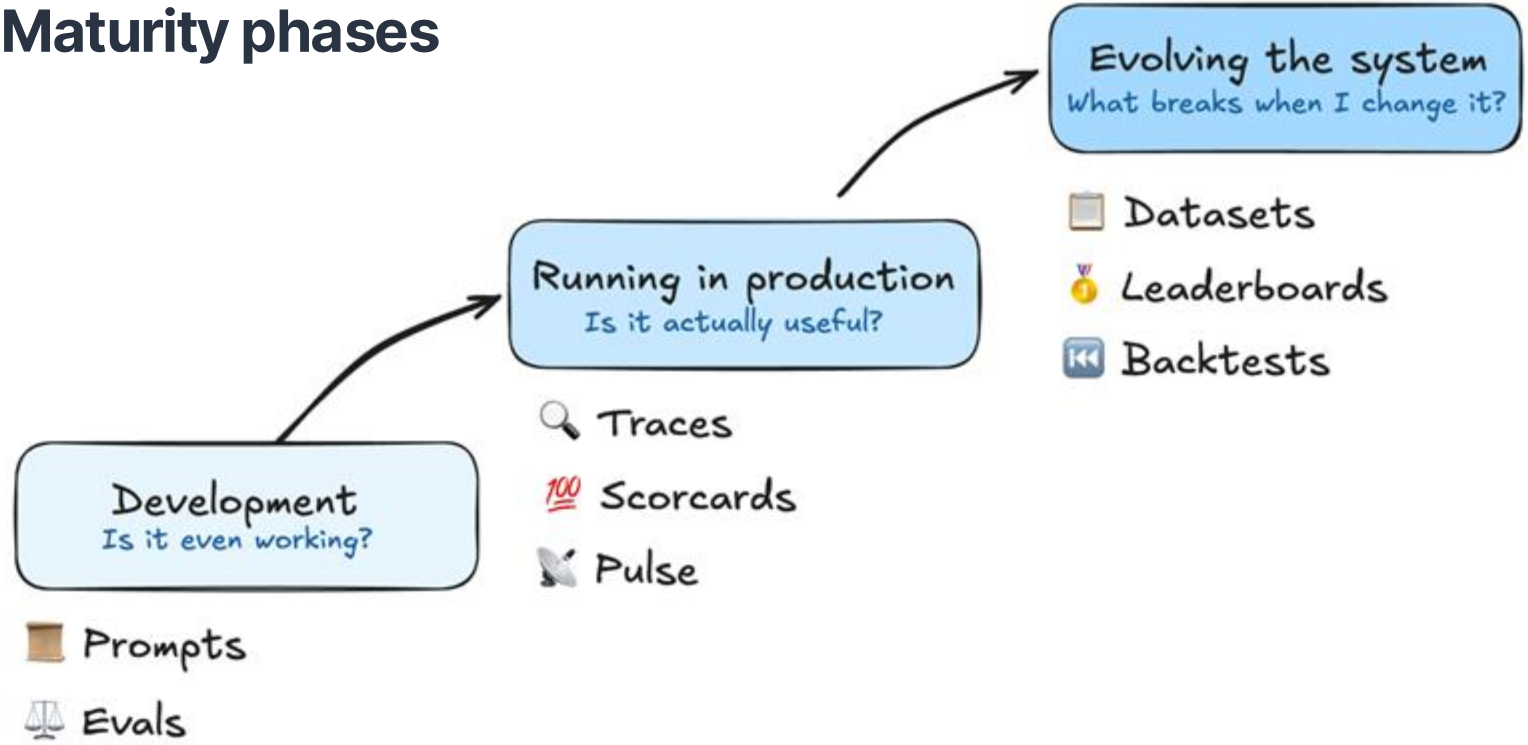
Investigation (messages)

Investigation (messages)

✓ Precision 80.6% 25 / 31

That's it!

Maturity phases



Development
Is it even working?

- 📜 Prompts
- ⚖️ Evals

Running in production
Is it actually useful?

- 🔍 Traces
- 💯 Scorcards
- 📡 Pulse

Evolving the system
What breaks when I change it?

- 📋 Datasets
- 🥇 Leaderboards
- ⏮️ Backtests

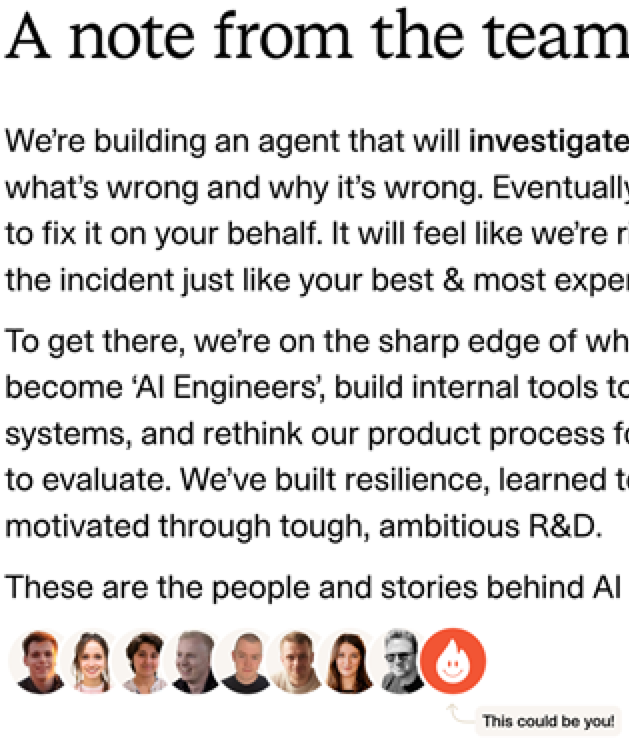
We’ve discussed tools, but they solve for very real human problems.

- ➔ Non-deterministic systems hurt
- ➔ Working with AI is a rollercoaster, it can be exhausting
- ➔ Objective measures are the answer
- ➔ Help celebrate the small wins

We are hiring Product Engineers in London!

https://incident.io/building-with-ai

lawrence@incident.io



A note from the team

We're building an agent that will **investigate** the incident **with** you, show you what's wrong and why it's wrong. Eventually, we'll show you how to fix it or offer to fix it on your behalf. It will feel like we're right there with you, helping you resolve the incident just like your best & most experienced colleague would.

To get there, we're on the sharp edge of what's possible with AI. It's pushed us to become 'AI Engineers', build internal tools to tame the chaos of non-deterministic systems, and rethink our product process for AI systems which are much harder to evaluate. We've built resilience, learned to navigate the uncertainty and stay motivated through tough, ambitious R&D.

These are the people and stories behind AI at incident.io.

incident.io raises $62M

To build AI agents that resolve incidents with you