

# Generative AI: Should We Say Goodbye to Deterministic Testing?

Dean Wampler, Ph.D.  
The AI Alliance and IBM  
[dwampler@thealliance.ai](mailto:dwampler@thealliance.ai)

[aialliance.org](http://aialliance.org)

[deanwampler.com/talks](http://deanwampler.com/talks)



# Outline

- First, about the AI Alliance
- How non-deterministic AI affects testing
- So, what should we developers do?
- Thinking about a new perspective

This isn't a "problem solved!" talk. I'll describe the problem and outline potential solutions.

# AI ALLIANCE

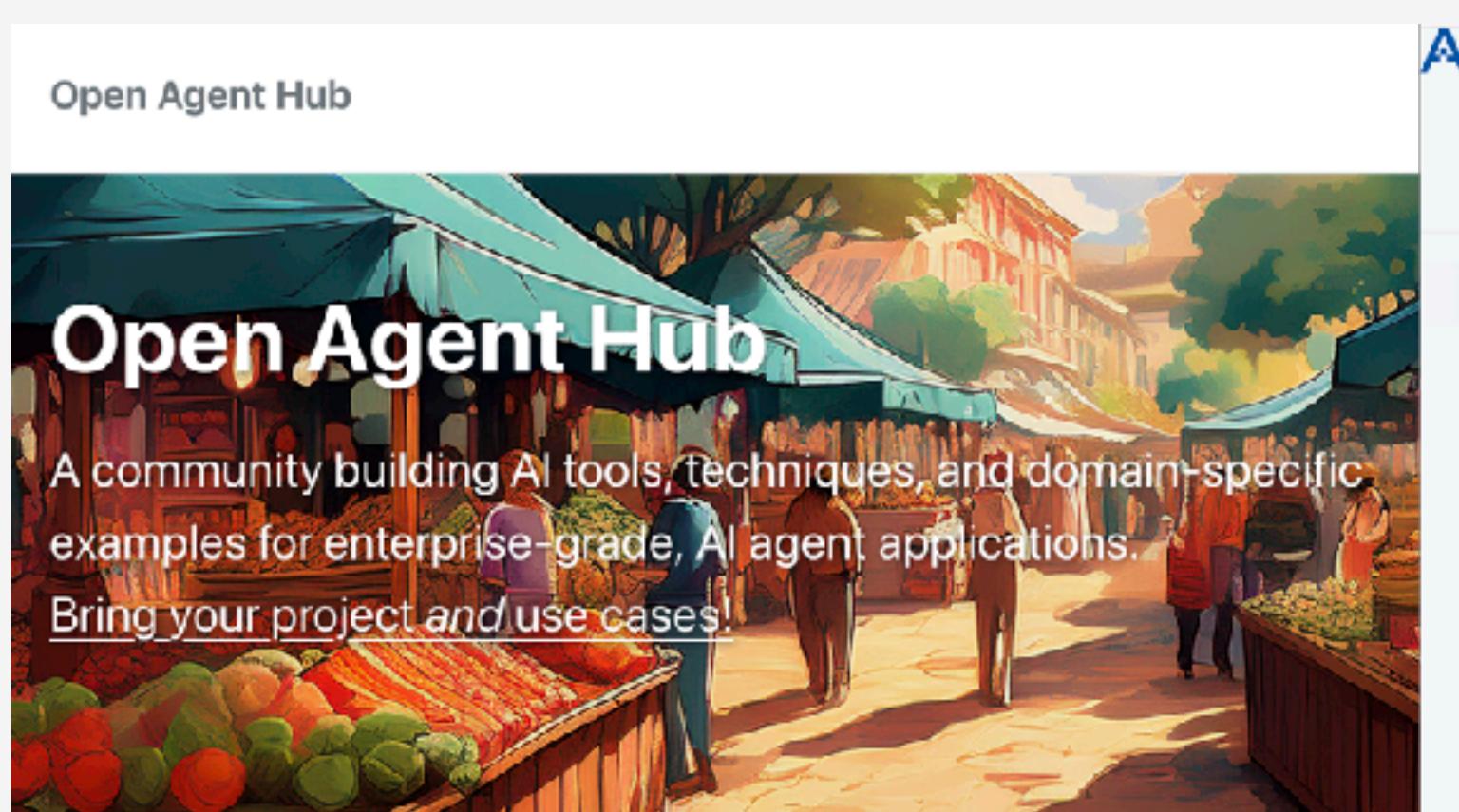
195+ organizations in 25+  
countries accelerating open  
innovation and adoption of AI

These project areas:  
[the-ai-alliance.github.io/](https://the-ai-alliance.github.io/)

# 1. Agents

# Data & Models

# 3. Safety & Governance



ALLIANCE

# Open Trusted Data Initiative

Start Here!

Trustworthiness

Dataset Catalog

Dataset Specification

How We Process Datasets

How to Contribute to OTDI

About Us

References

Q S B O W so Da Data Sub A E

The screenshot shows the homepage of the Open Trusted Data Initiative (OTD). At the top, there are navigation links for "Weights & Biases" and "Human". Below the header, there is a search bar with the placeholder text "Search Open Trusted Data Initiative". Two blue buttons are present: "Browse the Datasets" on the left and "Tell Us About Other Datasets" on the right. The main title "Open Trusted Data Initiative (OTD)" is displayed prominently. A call-to-action message below the title reads: "We are building the world's largest, most diverse catalog of open-sourced datasets for AI. Join us!" In the lower-left area, there is a section titled "Datasets for Languages" with a sub-section about datasets for different human languages. Below this, there are four green buttons representing language categories: "African Languages", "Languages in the Americas", "European Languages", and "Languages in the Middle East".

AI ALLIANCE

AI Alliance  
GitHub  
Organization

---

Home

Open Agent Hub Projects

Open Data and Model

Foundry Projects

Safety, Governance, and  
Education Projects

Contributing

About Us

Microsite Cheat Sheet

Q Search AI Alliance GitHub Organization

The

AI Alliance GitHub Organization Repos AI Alliance Events

# AI Safety, Governance, and Education

Collaborate on the necessary enablers of successful AI applications.

In order for the objectives of the [Open Agent Hub](#) and the [Open Data and Model Foundry](#) to be achieved, fundamental requirements must be met for safety, governance, and the expertise required to use AI technologies effectively.

**AI Safety** encompasses classic cybersecurity, as well as AI-specific concerns, such as suppression of undesirable content and compliance with regulations and social norms. A more general term is **trustworthiness**, which adds concerns about ensuring accuracy (i.e., minimizing hallucinations) and meeting the specific requirements for application use cases, etc. Enterprises won't deploy AI applications into production scenarios if they don't trust them to behave as expected.



 [aialliance.org](http://aialliance.org)

- AI for Vietnam
- CMC Corp
- FPT Software
- GenAI Fund
- SmartOSC
- VNPT-AI
- Vietnam

The *...*

The AI Alliance Community is a non-profit (501(c)(3)) foundation that supports the success of essential projects in the AI community with particular focus on data, models, and agents, including their governance dedicated to the benefit of AI for all of society - not just industry and not just those who can pay to have it.

# Testing Generative AI Agent Applications

[Home](#)[Testing Problems](#)[Architecture and Design for Testing](#)[Testing Strategies and Techniques](#)[Advanced Techniques](#)[The Working Example](#)[References](#)[Contributing](#)[About Us](#)[Join This Project](#)[GitHub Repo](#)

## Testing Generative AI Agent Applications

(Previous Title: Achieving Confidence in Enterprise AI Agent Applications)

*I am an enterprise developer; how do I test my AI agent applications??**I know how to test my traditional software, which is **deterministic** (more or less...), but I don't know how to test generative AI applications, which are uniquely **stochastic**, and therefore **nondeterministic**.*

Welcome to the **The AI Alliance** project to advance the state of the art for **Enterprise Testing of Generative AI Applications**. We are building the knowledge and tools you need to achieve the same testing confidence in your generative AI applications that you have for your traditional applications.

**Note:**

This site isn't about using AI to generate conventional tests (or code). You can find many online resources about that topic. Instead, this site focuses on the problem of how to do testing of any kind when an application contains generative AI components, given the nondeterminism they introduce.

### The Challenge We Face

We enterprise software developers know how to write [Repeatable](#) and [Automatable](#) tests. In particular, we rely on [Determinism](#) when we write tests to verify expected [Behavior](#) and to ensure that no [Regressions](#) occur as our code base evolves. Why is determinism a key ingredient? We know that if we pass the same arguments repeatedly to [most Functions](#) (with some exceptions), we will get the same answer back consistently. This property enables our core testing techniques, which give us [the essential confidence](#) that our applications meet our requirements, that they implement the [Use Cases](#) our customers expect. We are accustomed to unambiguous *pass/fail* answers!

Problems arise when we introduce [Generative AI Models](#), where generated output is inherently [Stochastic](#), meaning the outputs are governed by a probability model, and hence [nondeterministic](#). We can't write the same kinds of tests now, so what alternative approaches should we use instead? The problems are compounded when we have applications built on [Agents](#), each of which will have some stochastic behavior of its own, if it's not [deterministic](#) at all.

In contrast, our AI-expert colleagues (researchers and data scientists) are interested in how well their models perform against particular objectives. For example, a model might be trained to predict stochastic model responses and to assess how well the models perform against particular objectives. For example, a model might be trained to predict the probability of a certain outcome given a set of inputs, and then use that information to make decisions or recommendations.

An AI Alliance project I lead to work on AI testing tools and techniques.

[Projects](#)[Governance, and on Projects](#)[Testing](#)[AI Safety Cheat Sheet](#)

In order for the objectives of the [Open Agent Hub](#) and the [Open Data and Model Foundry](#) to be achieved, fundamental requirements must be met for safety, governance, and the expertise required to use AI technologies effectively.

**AI Safety** encompasses classic cybersecurity, as well as AI-specific concerns, such as suppression of undesirable content and compliance with regulations and social norms. A more general term is **trustworthiness**, which adds concerns about ensuring accuracy (i.e., minimizing hallucinations) and meeting the specific requirements for application use cases, etc. Enterprises won't deploy AI applications into production scenarios if they don't trust them to behave as expected.

**Governance** is an aspect of trustworthiness, specifically the assurances that all end-to-end processes used to create all AI application components are secure, licensed for use, etc. AI models are created with data; they are mostly data themselves. Hence, models, like data, need to be governed.

Finally, **Education** addresses the needs where organizations struggle to learn all the things they need to know in order to use AI safely and effectively. Not only has AI introduced new tools and techniques to software application development, it has fundamentally altered some of the ways software is developed, for example, introducing *stochastic* behaviors as core aspects of application features, where previously *deterministic* behaviors were the norm. Most AI Alliance projects have dual missions: not only to innovate and create, but to educate.

The following projects address these concerns.

[Links](#)[Description](#)[The AI Trust and Safety User Guide](#)

# AI ALLIANCE

Join us!

- [aialliance.org](http://aialliance.org)



[bsky.app/profile/aialliance.bsky.social](https://bsky.app/profile/aialliance.bsky.social)



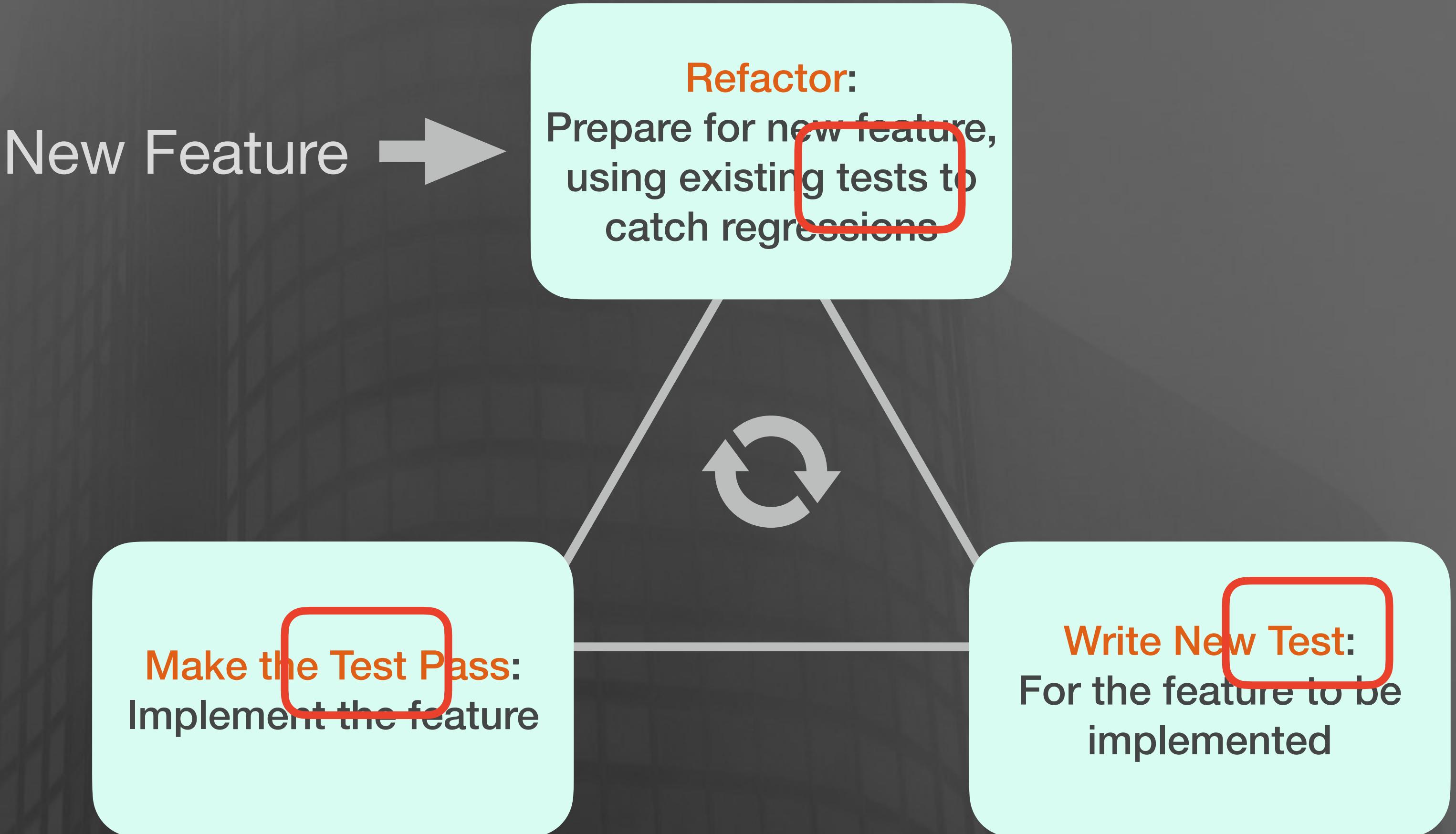
[linkedin.com/company/the-aialliance/](https://linkedin.com/company/the-aialliance/)



AI is non-deterministic.  
How does that affect testability?

# Remember the TDD<sup>‡</sup> loop?

Testing is the foundation of this process!



<sup>‡</sup> Test-Driven Development

# What Do Developers Expect?

Developers expect software to be deterministic<sup>‡</sup>. This helps ensure correctness, and reproducibility enables automation that catches regressions:

- The same input → the same output.
  - e.g.,  $\sin(\pi) = -1$
- The output changes? Something broke!

<sup>‡</sup> Distributed systems break this clean picture.

# What Do Developers Expect?

Developers expect behavior to be  
reproducible.

- The same results
- e.g.,  
the same output
- The order of events

Put another way, the system is deterministic.<sup>‡</sup>  
determinism makes it easier to specify the system invariants.  
What should remain true before and after each step?  
The system must not be broken!

<sup>‡</sup> Distributed systems break this clean picture.

# What Do Developers Expect?

Functional Programming gave us property-based testing:

- E.g., QuickCheck, Hypothesis, ScalaCheck, ...
- Hypothesis example:

```
@given(st.integers(), nonzero_integers, st.integers(), nonzero_integers)
def test_two_non_identical_rationals_are_not_equal_to_each_other(self, numer1, denom1, numer2, denom2):
    """
    Rule: a/b == c/d iff ad == bc
    This is a better test, because it randomly generates different instances.
    However, the test has to check for the case where the two values happen to be
    equivalent!
    """
    rat1 = Rational(numer1, denom1)
    rat2 = Rational(numer2, denom2)
    if numer1*denom2 == numer2*denom1:
        self.assertEqual(rat1, rat2)
    else:
        self.assertNotEqual(rat1, rat2)
```

# What do we get with generative AI?

Generative models are stochastic<sup>‡</sup>:

- The same prompt → **different** output.
- `chatgpt("Write a poem")` → **insanity**

“Insanity is doing  
the same thing  
over and over  
again and  
expecting  
different results.”  
— not Einstein

<sup>‡</sup>Stochastic : described by a random probability distribution, e.g., flipping a coin, rolling dice, measuring the temperature, ...

# What do we get with generative AI?

Generative models are stochastic<sup>‡</sup>:

- The same prompt → **different** output.
  - chatgpt("Write a poem") → **insanity**
- Without determinism, how do you write repeatable, reliable tests for AI apps?
  - Does that new model perform better or worse than the previous model?
  - Did any regressions in behavior occur?

# What do we get with generative AI?

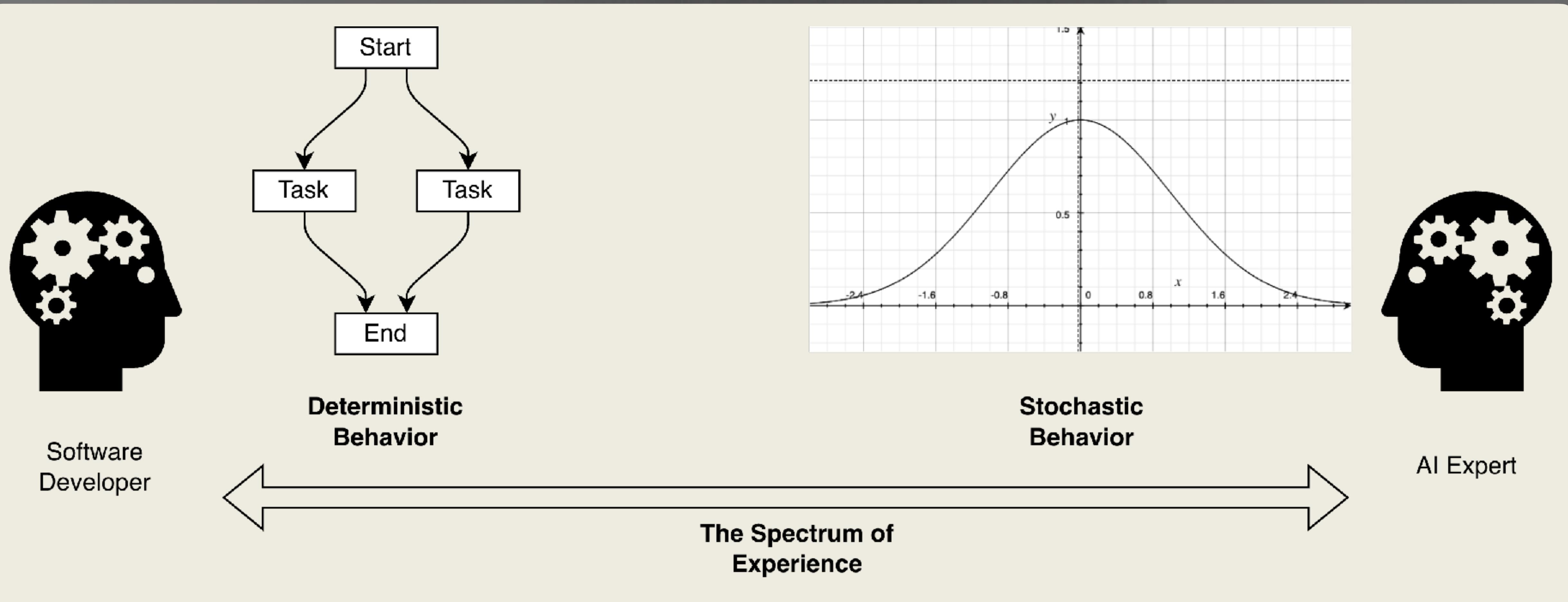
## Generative

- The
- cha
- With
- repe
- Do
- tha
- Did

Put another way, the **invariants** are much less clear and therefore harder to define programmatically and enforce.

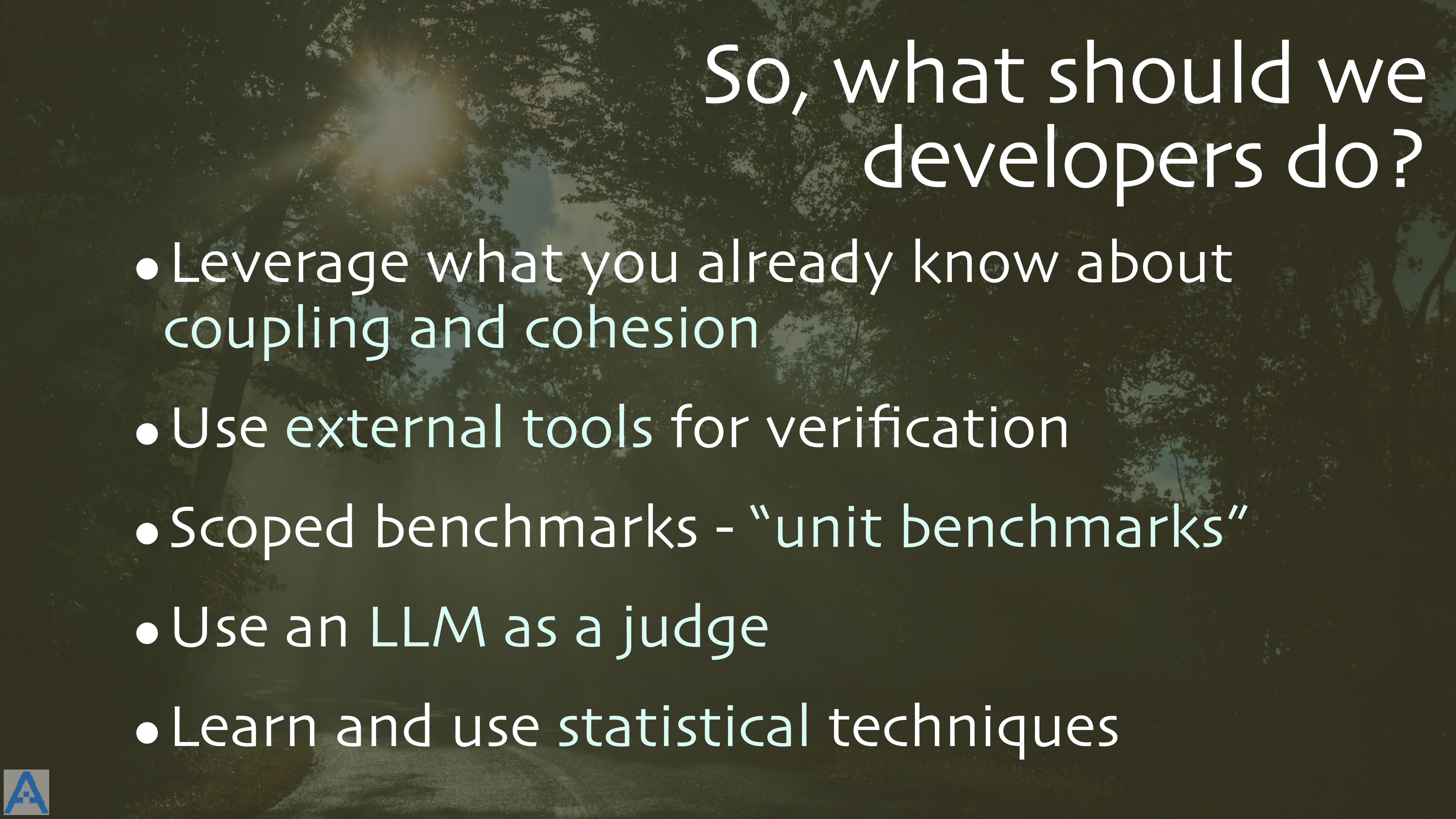
ite  
or worse  
?

# But Data Scientists and AI Experts Are Accustomed to Stochasticity



A photograph of a forest path. Sunlight filters through the dense canopy of tall trees, creating bright rays and shadows on the ground. The path is covered in fallen leaves and appears slightly damp. The overall atmosphere is peaceful and somewhat mysterious.

So, what should we  
developers do?



So, what should we  
developers do?

- Leverage what you already know about coupling and cohesion
- Use external tools for verification
- Scoped benchmarks - “unit benchmarks”
- Use an LLM as a judge
- Learn and use statistical techniques

[JOIN THIS PROJECT](#)[Github Repo](#)

# Testing Generative AI Agent Applications

(Previous Title: Achieving Confidence in Enterprise AI Agent Applications)

*I am an enterprise developer; how do I test my AI agent applications??*

*I know how to test my traditional software, which is **deterministic** (more or less...), but I don't know how to test my AI agent applications, which are uniquely **stochastic**, and therefore **nondeterministic**.*

Welcome to the **The AI Alliance** project to advance **Testing of Generative AI Agent Applications**. We believe you need to achieve the same testing confidence in your AI agent applications as you do in your traditional applications.

## Note:

This site isn't about using AI to generate coverage reports or automated tests. There are many online resources about that topic. Instead, this site is about how to approach the challenges of testing of any kind when an application contains nondeterminism they introduce.

An AI Alliance project I lead to:

- Develop new developer testing tools and techniques adopted from data science.
- Teach developers how to use them.

## The Challenge We Face

We enterprise software developers know how to write **Repeatable**  and **Automatable**  tests.

# Coupling and Cohesion



# Coupling and Cohesion

- The non-deterministic AI model isn't the whole application. (E.g., Agent architectures)
  - Wrap the model in a good API.
  - Use deterministic test doubles for it.
  - Test everything else like you normally do.

# Coupling and Cohesion

- Writing a good API:
  - Engineer your prompts to constrain outputs.
  - Use tools like Pydantic-AI for type safety (example).
  - Select the Gen AI models that seem to work best with your tools.

# Coupling and Cohesion

- Writing
- Engineering
- Use tools  
(examples)
- Select the best way

Thinking about types encourages you to find ways to constrain model queries such that the responses are more closely aligned with your goals.

outputs.  
safety  
to work

# Coupling and Cohesion

- Writing
- Engineering
- Use tools  
(examples)
- Select the best way

Most AI-enabled apps won't be open-ended chatbots, but use AI to resiliently translate between human text and tool APIs, and translate tool-to-tool interactions, so we don't have to do that translation in code ourselves.

outputs.

safety

to work

# Coupling and Cohesion

- However, tried and true C&C techniques don't help us test the model input and output behaviors themselves, nor do they eliminate the non-determinism that is unavoidable in our acceptance tests<sup>‡</sup>.

<sup>‡</sup> The integration tests that prove features are done.

A photograph of a dense forest. The foreground features several tall, thin trees with dark trunks and sparse, silhouetted branches. A small evergreen tree is visible in the lower left. The background is filled with more trees, their trunks creating vertical lines against a hazy, light-colored sky.

Use external tools for verification

# Use external tools for verification

- Are you asking a model to generate code?
- Check it with a parser or compiler
- Scan for security vulnerabilities
- Check for excessive cyclomatic complexity
- Check that only allowed third-party libraries and versions are used.
- ...

# Use external tools for verification

- Are you asking a model to generate code?
- Using TDD? If you ask for code that makes your hand-written tests pass, does the generated code allow the tests to pass?
- (Example)

# Use external tools for verification

- Are you writing tests by hand?
- Using a tool to generate tests?
- (Example)

Currently, I don't think many models are very good at generating powerful tests, but they can do a reasonable job generating code to pass the tests.

ode?  
makes  
ne-  
ss?

# Use external tools for verification

- Are you asking a model to do logic or reasoning?
- Check it with a logic/reasoning engine
- Or use that tool instead to create your logic!

# Use external tools for verification

- Are you asking a model to do planning?
- Check it with a planning engine
- Or use that tool instead to create your plan!

# Use external tools for verification

- Are you asking a model to generate possible chemicals or physical processes?
- Try creating and testing the chemical in a lab.
- Test the physical process with a simulator.
- (Letting AI generate the “idea”, then testing in a simulator may be cheaper than using the simulator to generate possible ideas.)

# Use external tools for verification



One of the reason that Agents are so popular now is the recognition that models can't do everything well (or cheaply). So, complementing models with other tools provides the best results.

# Scoped benchmarks - “unit benchmarks”



# Scoped benchmarks - “unit benchmarks”

- Models are evaluated with benchmarks.
- Use a large number of examples.
- Typically cover a broad topic,
  - e.g., effective Q&A, detect hate speech, detect bias, measure throughput, ...
- Return a single measurement, usually 0-100%.

# Scoped benchmarks - “unit benchmarks”

- Not the same thing as a developer “test”.
- But can we adapt the idea for testing?
  - Use a very narrow scope.
  - Still use a lot of examples for higher confidence.
  - Return a single measurement, usually 0-100%.
    - But at what threshold do you “pass”??

# Scoped benchmarks - “unit benchmarks”

- Example: SQL queries generated from text.
  - Build a Q&A dataset that uses logged queries (expected answers) with appropriate human prompts (the queries).
  - Each unit benchmark might focus on one specific kind of common query.

This is also an example of using a model to translate between text and an “API”.

# Use an LLM as a judge



# Use an LLM as a judge

- You have probably chosen a small model for production, because it costs less to use.
- Use a bigger, smarter for test runs to “judge” responses.
- You’ll call it less often, so the cost won’t be as much of an issue.

Need data for your unit benchmarks? Use a big model to synthesize data!

# Use an LLM as a judge

- It can work like this:
  - A test sends a query to the model or app.
  - The query and the response are sent to a larger model with the question, “Is this a good response for this query? Answer yes or no, and if no, provide an explanation.”
  - Fail the test if the answer is no.
  - Use the explanation to debug.

A wide-angle photograph of a forested hillside. The foreground is shrouded in a thick layer of white fog or mist. The middle ground shows the hillside covered in dense green trees, with some yellow and orange leaves visible, indicating autumn. The background is a darker, more heavily wooded area, also partially obscured by mist.

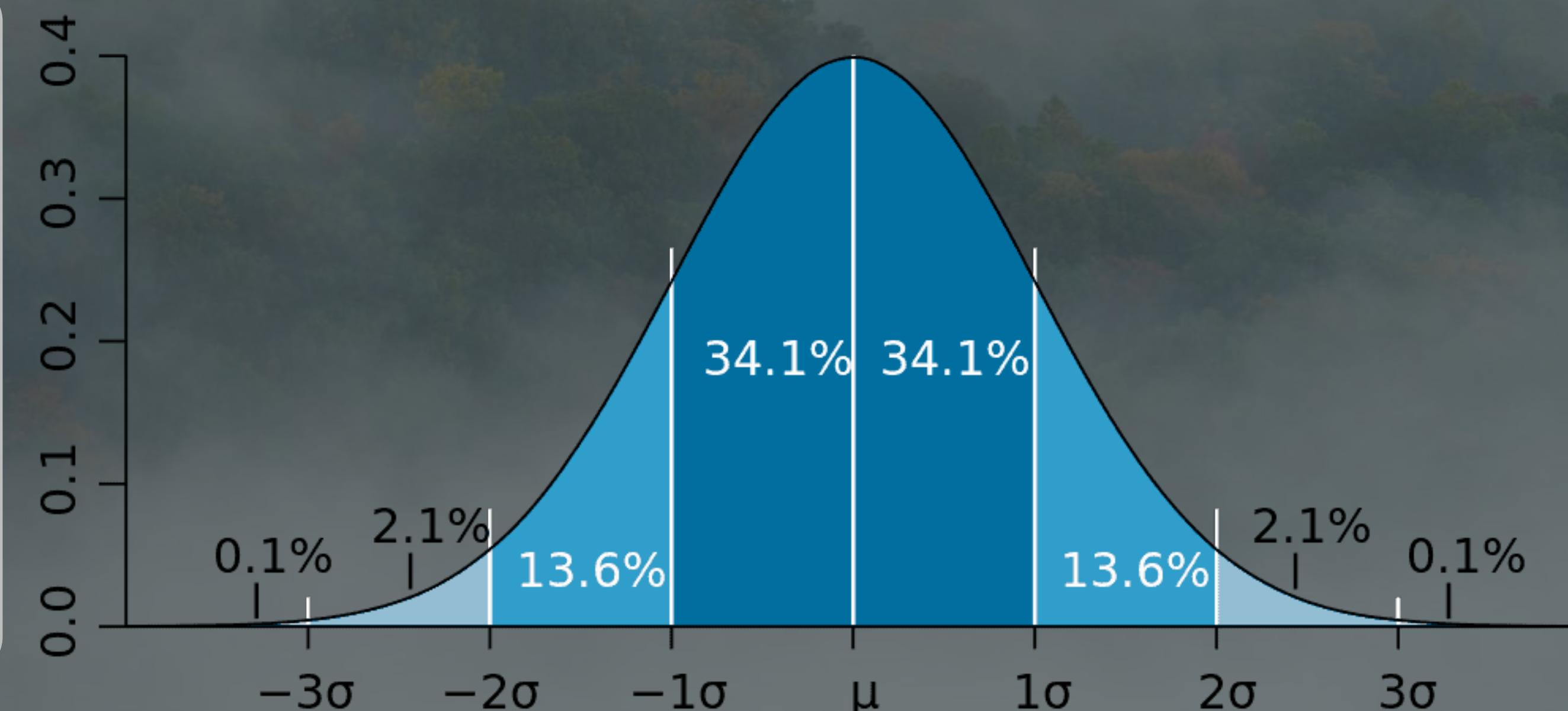
# Understand and leverage statistics

# Understand and leverage statistics

- Scientists are accustomed to using statistics to analyze probabilistic phenomena.
- E.g., a potential discrepancy between theory and experiment must be > five sigma.

"A five-sigma level translates to one chance in 3.5 million that a random fluctuation would yield the result."

Wikipedia



# Understand and leverage statistics

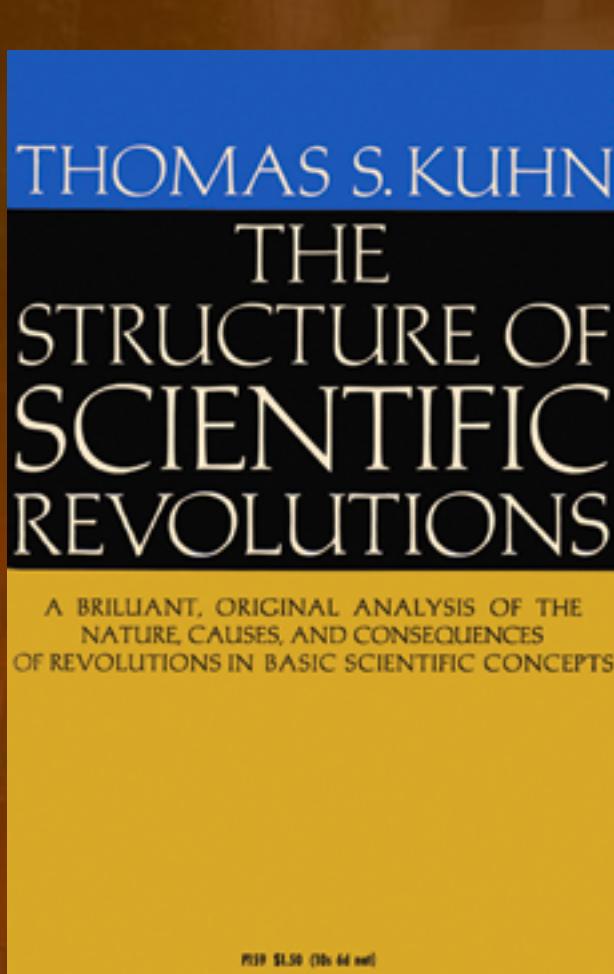
- Classifier models sometimes return a confidence level, i.e., how much they believe they are returning the correct classification.
- “Adding Error Bars to Eval: A Statistical Approach to Language Model Evaluations”
- <https://arxiv.org/abs/2411.00640>



Thinking about a new perspective

# Thinking about a new perspective?

- The Structure of Scientific Revolutions
- We prefer to adapt our current theory to accommodate new data, rather than discard the theory and start over.
- But sometimes, we need to restart from first principles.



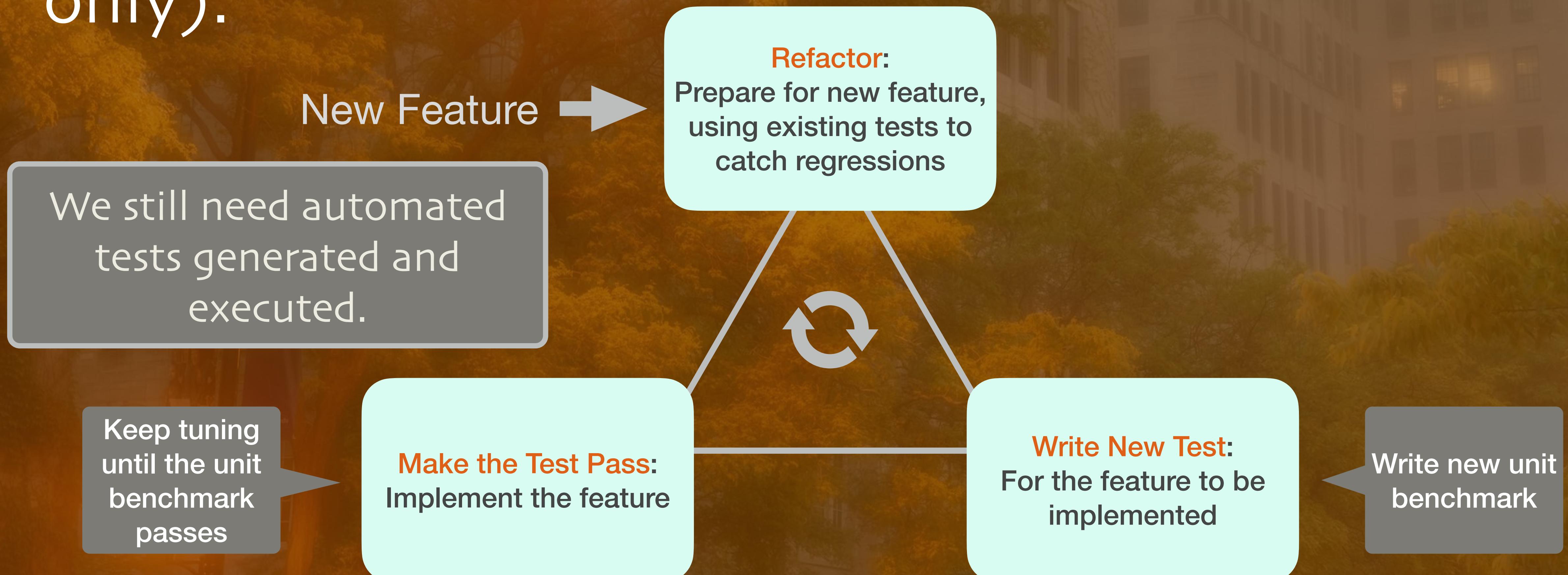
Maybe we have to abandon the idea of deterministic testing, but approach the challenge in a new way.

# From Testing to Continuous Tuning

- ★ What if we switch from testing for desired behavior to tuning for desired behavior?
- We already tune models to improve domain-specific knowledge, chatbot behavior, etc.
- Today: it's only done during model creation.
- Tomorrow: continuously tune incrementally.

# From Testing to Continuous Tuning

- Changes to the TDD cycle (for model behaviors only):



# Thank you!

[thealliance.ai](https://thealliance.ai)

[dwampler@thealliance.ai](mailto:dwampler@thealliance.ai)

Mastodon and Bluesky: @deanwampler

[aialliance.org](https://aialliance.org)

[deanwampler.com/talks](https://deanwampler.com/talks)

