Generative AI: Should We Say Goodbye to Deterministic Testing?

Dean Wampler, Ph.D.
The AI Alliance and IBM Research

January 24, 2025

thealliance.ai

deanwampler.com/talks

AI Alliance

Outline

- First, about the AI Alliance
- How non-deterministic GenAI affects testing
- What we can do about the challenges
- Adopt a new perspective?

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

Outline

- First, about the AI Alliance
- How non-deterministic GenAI affects testing
- What we can do about the challenges
- Adopt a new perspective?

What’s the connection to FP?
I’ll discuss connections
as we go…

The AI Alliance brings together organizations, people, and resources to accelerate ***open innovation***, ***technology development*** and ***adoption***.

Launched December 5, 2023

Map of Members

Member organizations in the AI Alliance comprise academia, commercial, research and non-profits and span the globe.

Our core beliefs in AI that is open is the tie that binds us, despite our differences.

AI Alliance

The AI Alliance is made up of +140 organizations in +20 countries, and growing

U.S. - Indiana
- University of Notre Dame

U.S. - Utah
- University of Utah

U.S. - Ohio
- Cleveland Clinic

U.S. - Connecticut
- Yale University

U.S. - New Hampshire
- Dartmouth

U.S. - Oklahoma
- University of Oklahoma

France
- Institut Polytechnique de Paris
- Impact AI
- Datacraft

Germany
- EOS GmbH
- TU Munich

Austria
- IDea_Lab of University of Graz

U.S. - California
- UC Berkeley's College of Computing, Data Science, and Society
- Ainekko, Co
- Aitomatic
- AMD
- Anyone AI Inc.
- Anyscale
- Cerebras
- Databricks
- Domino Data Lab
- Fast.ai
- GMI Cloud
- HydroX AI
- Intel
- LangChain
- LlamaIndex
- Meta
- Modular Inc
- neo4j
- Predibase
- Roadzen
- Salesforce
- ServiceNow
- Together AI
- Uber
- Weights & Biases
- Zilliz
- Linux Foundation
- MLCommons
- Partnership on AI

U.S. - Illinois
- University of Illinois Urbana-Champaign
- Center for Advancing Safety of Machine Intelligence (CASMI)

Canada
- Montreal AI Ethics Institute

U.S. - Montana
- Snowflake

U.S. - Pennsylvania
- University of Pennsylvania

Switzerland
- CERN
- Ecole Polytechnique Fédérale de Lausanne
- ETH Zurich

U.S. - Massachusetts
- Northeastern University
- Mass Open Cloud Alliance

UK
- Imperial College of London
- OpenMined
- Stability AI
- OpenUK

Finland
- Silo AI

Bulgaria
- Institute for Computer Science, Artificial Intelligence and Technology

Poland
- Poznan University of Technology: Interdisciplinary Centre for Artificial Intelligence and Cybersecurity

Japan
- Keio University
- Tokyo Institute of Technology
- The University of Tokyo
- Citadel AI
- Fenrir Inc
- Hitachi
- NEC Corporation
- Panasonic Holdings Corporation
- SakunaAI
- SB Intuitions (Softbank subsidiary)
- SONY Group
- Tokyo Electron Limited
- Sakana AI

Senegal
- Kera Health

Israel
- Hebrew University
- Neureality

U.S. - Washington
- Allen Institute for AI

U.S. - Nevada
- Senzing, Inc.

Brazil
- Universidade de São Paulo

U.S. - Delaware
- New Native Inc.

U.S. - Washington D.C.
- National Aeronautics and Space Administration*
- Seed AI

Spain
- ESADE
- MLOps Community
- Barcelona Supercomputing Center

India
- AI4Bharat, IIT Madras
- IIT Bombay
- IIT Jodhpur
- Infosys
- KissanAI
- people+ai
- Sarvam AI
- Wadhwani Institute AI

U.S. - Texas
- University of Texas at Austin
- Anaconda
- Applied Digital
- Dell Technologies
- OpenTeams
- Oracle
- Quansight
- Humane Intelligence
- NumFOCUS

U.S. - New York
- Cornell University
- NYU
- Rensselaer Polytechnic Institute
- St. John's University
- University at Buffalo
- Hugging Face
- IBM
- Lightning AI
- LastMile AI
- Ontocord.AI
- Simons Foundation & Flatiron Institute

Germany
- University of Bayreuth

Vietnam
- FPT Software

U.A.E.
- Mohamed bin Zayed University of Artificial Intelligence
- Core42
- Technology Innovation Institute

Singapore
- A*STAR

Korea
- Kakaocorp

Italy
- International Centre for Theoretical Physics
- International School for Advanced Study

Australia
- Fast.ai

Taiwan
- MediaTek Research

U.S. - Virginia
- National Science Foundation*

U.S. - North Carolina
- NC State University
- Red Hat

Focus Areas & Mission

Represents the investment priorities for the AI Alliance

Member organizations have the choice to take part in one or more of these six focus areas and the agility to shift participation based on their interest and priorities.

1. Skills & Education

Support global AI skills building, education, and exploratory research.

2. Trust & Safety

Create benchmarks, tools, and methodologies to ensure and evaluate high-quality and safe AI.

3. Applications & Tools

Build and advance efficient and capable software frameworks for model builders and developers.

4. HW Enablement

Foster a vibrant AI hardware accelerator ecosystem through SW.

5. Foundation Models & Data

Enable an ecosystem of open foundation models and datasets for diverse modalities.

6. Advocacy

Advocate for regulatory policies that create a healthy open ecosystem for AI.

Focus Areas & Mission

Represents the investment priorities for the AI Alliance

Member organizations have the choice to take part in one or more of these six focus areas and the agility to shift participation based on their interest and priorities.

1. Skills & Education

Support global AI building, educatio exploratory resea

2. Trust & Safety

3. Applications & Tools

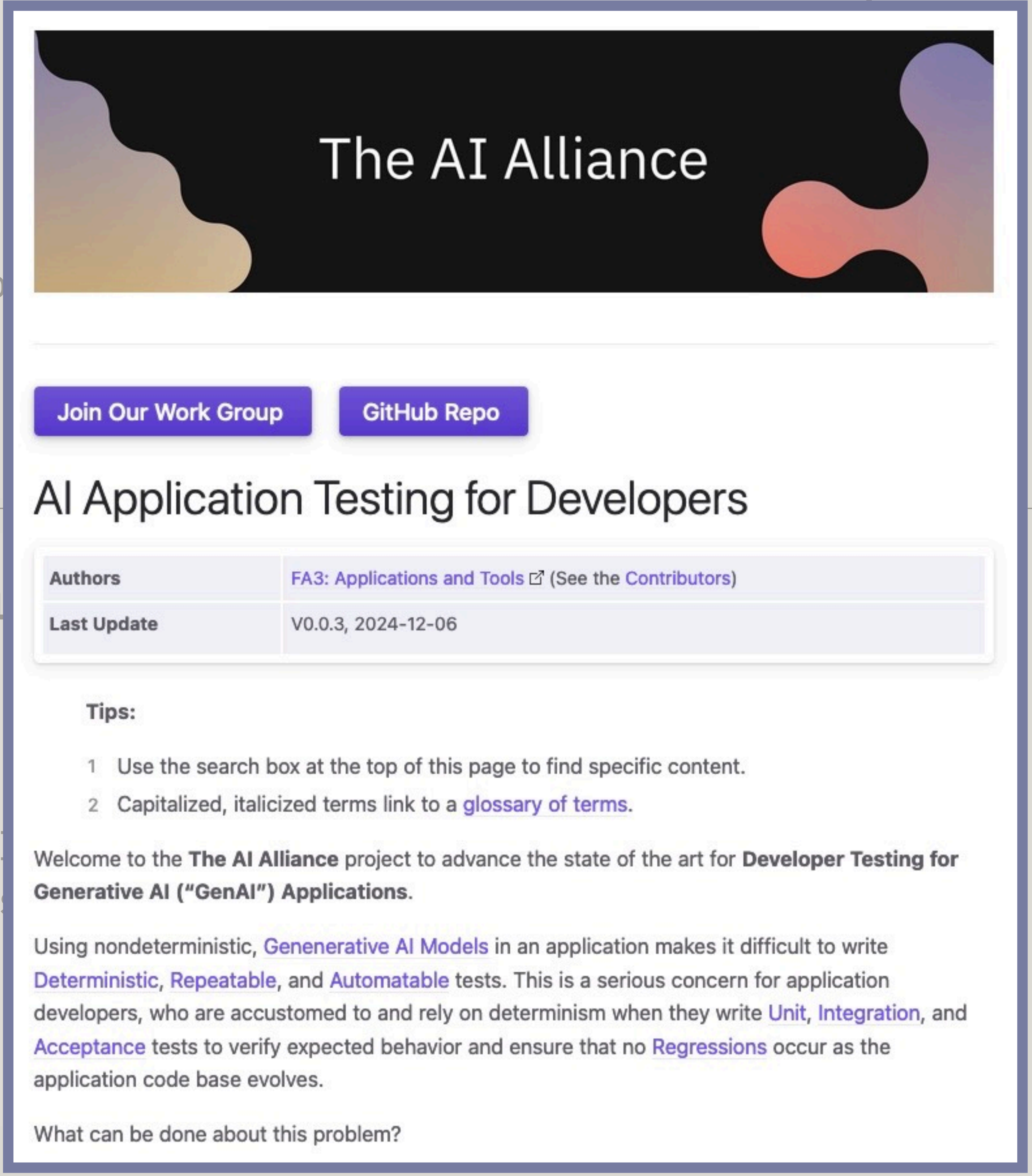
Build and advance efficient and capable software frameworks for model builders and developers.

4. HW Ena

Foster a vibrant A accelerator ecosy through SW.

6. Advocacy

Advocate for regulatory policies that create a healthy open ecosystem for AI.



AI Application Testing for Developers

Authors	FA3: Applications and Tools (See the Contributors)
Last Update	V0.0.3, 2024-12-06

Tips:

1. Use the search box at the top of this page to find specific content.
2. Capitalized, italicized terms link to a glossary of terms.

Welcome to the **The AI Alliance** project to advance the state of the art for **Developer Testing for Generative AI ("GenAI") Applications**.

Using nondeterministic, Genenerative AI Models in an application makes it difficult to write Deterministic, Repeatable, and Automatable tests. This is a serious concern for application developers, who are accustomed to and rely on determinism when they write Unit, Integration, and Acceptance tests to verify expected behavior and ensure that no Regressions occur as the application code base evolves.

What can be done about this problem?

https://the-ai-alliance.github.io/ai-application-testing/

https://bsky.app/profile/aialliance.bsky.social

Join us!

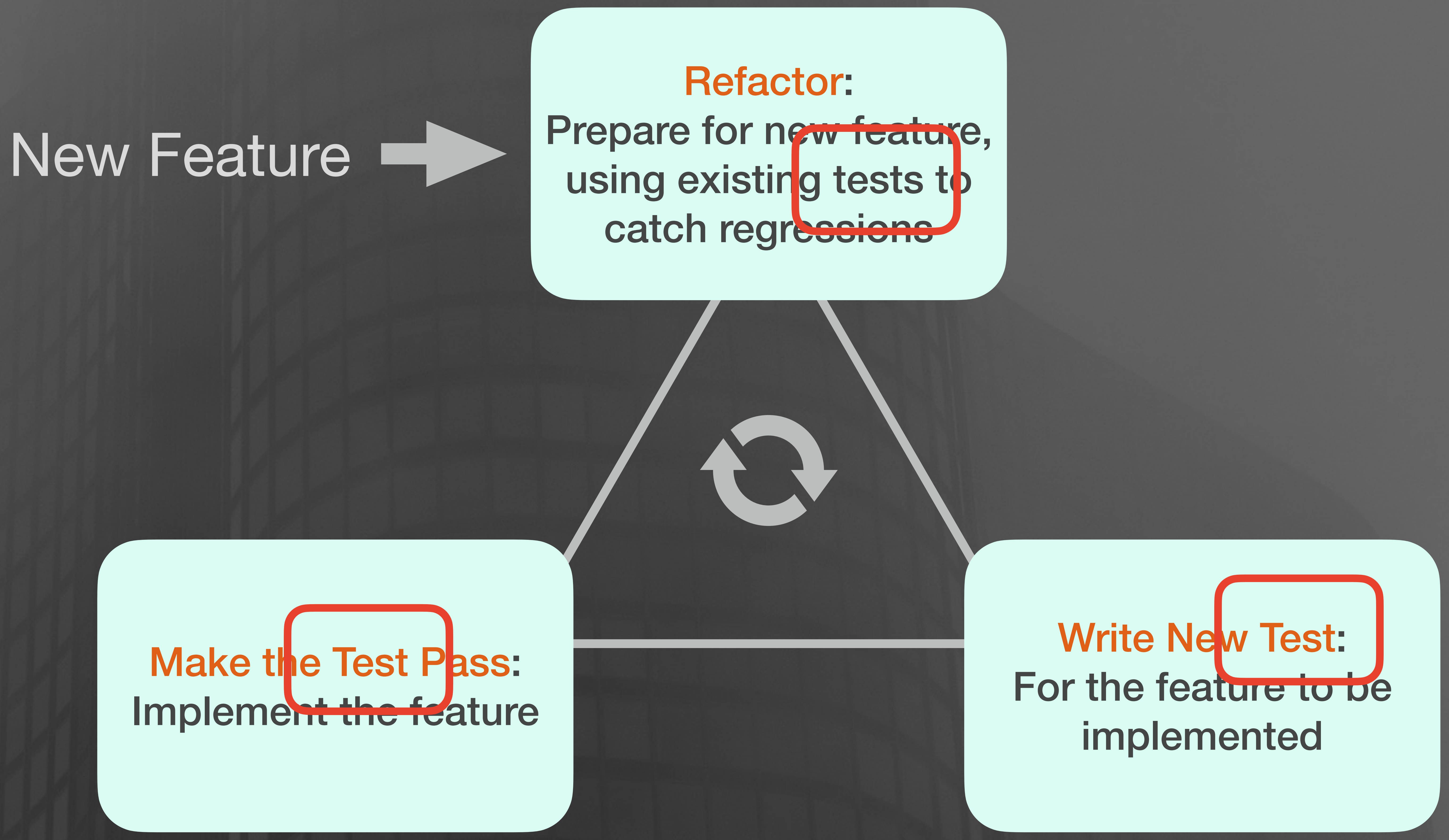
- thealliance.ai

bsky.app/profile/aialliance.bsky.social

linkedin.com/company/the-aialliance/

How non-deterministic GenAI affects testing

Remember the TDD‡ loop?


New Feature
Refactor:
Prepare for new feature, using existing tests to catch regressions
Make the Test Pass:
Implement the feature
Write New Test:
For the feature to be implemented

Testing is integral to this process!

‡ Test-Driven Development

What Do Developers Expect?

Developers expect software to be deterministic[‡]:

- The same input → the same output.
  - e.g., sin(π) = -1
- The output is different? Something is broken!
- Developers rely on determinism to help ensure correctness and reproducibility, and to catch regressions.

[‡] Distributed systems break this clean picture.

What Do Developers Expect?

Develope inistic[‡]:

- The sa
  - e.g.,
- The o roken!
- Devel ensure correc atch regres

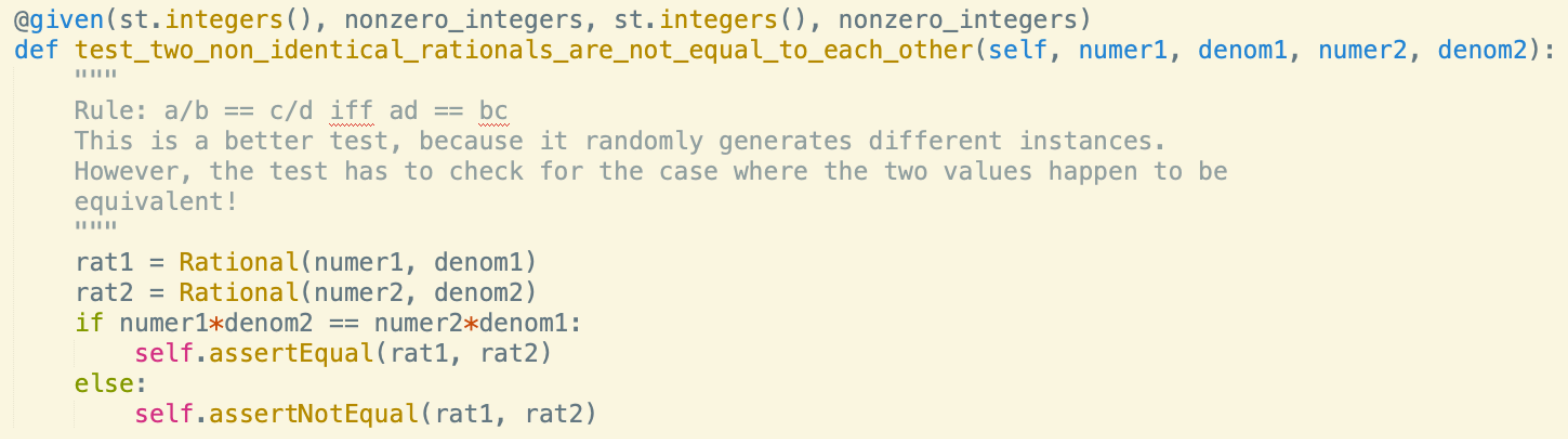
Put another way, the determinism makes it easier to specify the system invariants. What should remain true before and after each step?

[‡] Distributed systems break this clean picture.

What Do Developers Expect?

FP gave us property-based testing:

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



```python
@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)
```

From: https://github.com/deanwampler/tdd-hypothesis-example

What do we get with generative AI?

Generative models are probabilistic[‡]:

- 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

[‡] A tunable "temperature" controls how probabilistic.

What do we get with generative AI?

Generative models are probabilistic‡:

- 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

- Without determinism, how do you write repeatable, reliable tests? Specifically for GenAI,
  - Is that new model actually better or worse than the previous model, in my application?
  - Did any regressions in other behavior occur?

‡ A tunable “temperature” controls how probabilistic.

What do we get with generative AI?

Generat

- The .
  - cha
- With ite repe for GenAI,
  - Is t worse tha
  - Di r occur?

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

‡ A tunable “temperature” controls how probabilistic.

What we can do about the challenges

What we can do about the challenges

- Don’t forget about coupling and cohesion
- Use external tools for verification
- Adapt benchmarks - “unit benchmarks”
- Use an LLM as a judge
- Understand and leverage statistics

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
  - Engine outputs.
  - Use too afety (examp
  - Select t to work best wi

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

Coupling and Cohesion

- Writing
  - Engine outputs.
  - Use too afety (examp
  - Select t to work best wi

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.

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[‡].

[‡] The integration tests that prove features are done.

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 ode?
  - Using makes your ha ne generat ss?
    - (Exan

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.

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.

Adapt benchmarks - “unit benchmarks”

Adapt 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%.

Adapt 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”??

Adapt 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

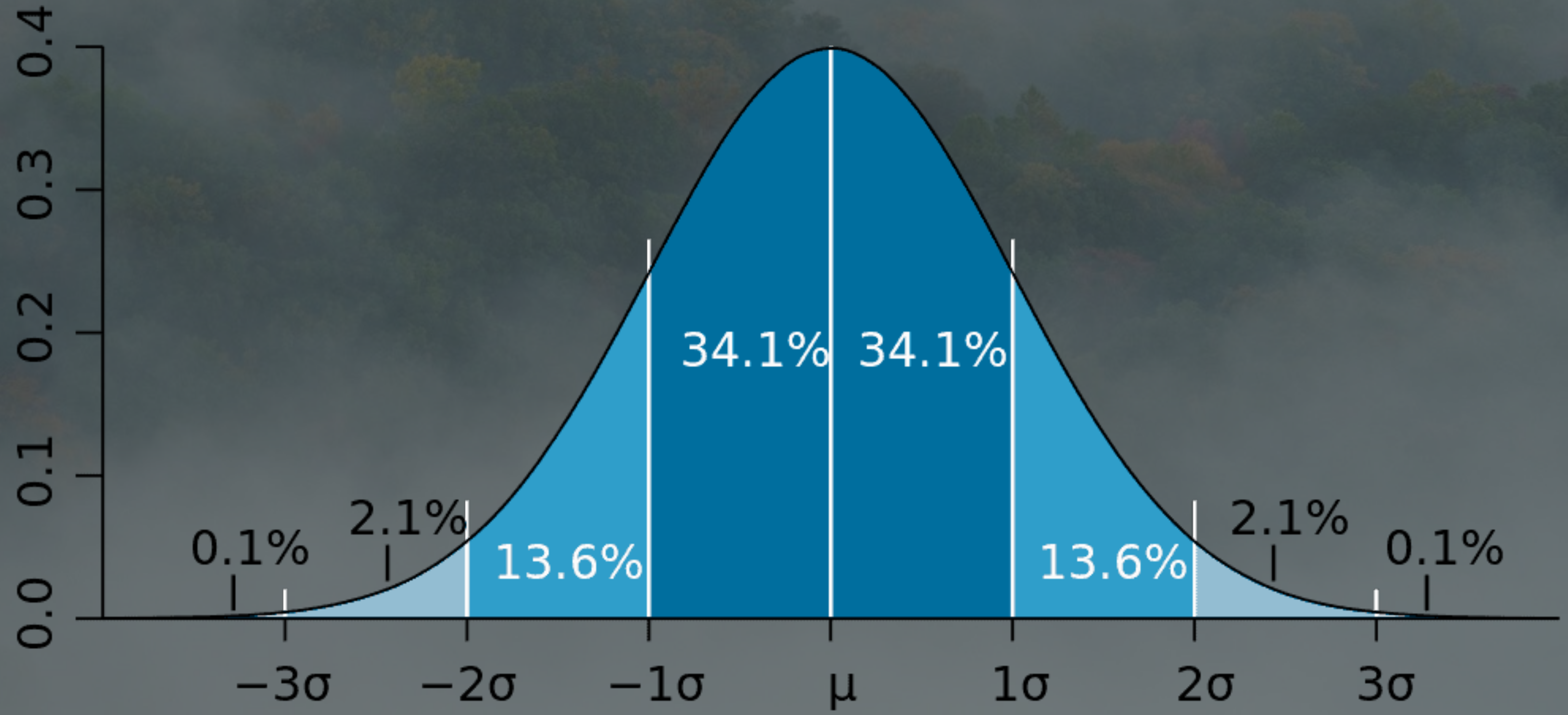
- It can work like this:
  - A test sends a query to the model or app.
  - The query and the response are sent to a larger model with the question, “Is this a good response for this query? Answer yes or no, and if no, provide an explanation.”
  - Fail the test if the answer is no.
  - Use the explanation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

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 Evals: A Statistical Approach to Language Model Evaluations”
    - https://arxiv.org/abs/2411.00640

A new perspective?

A new perspective?

- The Structure of Scientific Revolutions
  - It’s normal to try to bend our current theory to accommodate new data, rather than simply throw out our current theory and start over from the fundamentals.

Should we abandon the idea of deterministic testing, at least for model outputs, in favor of a new approach?

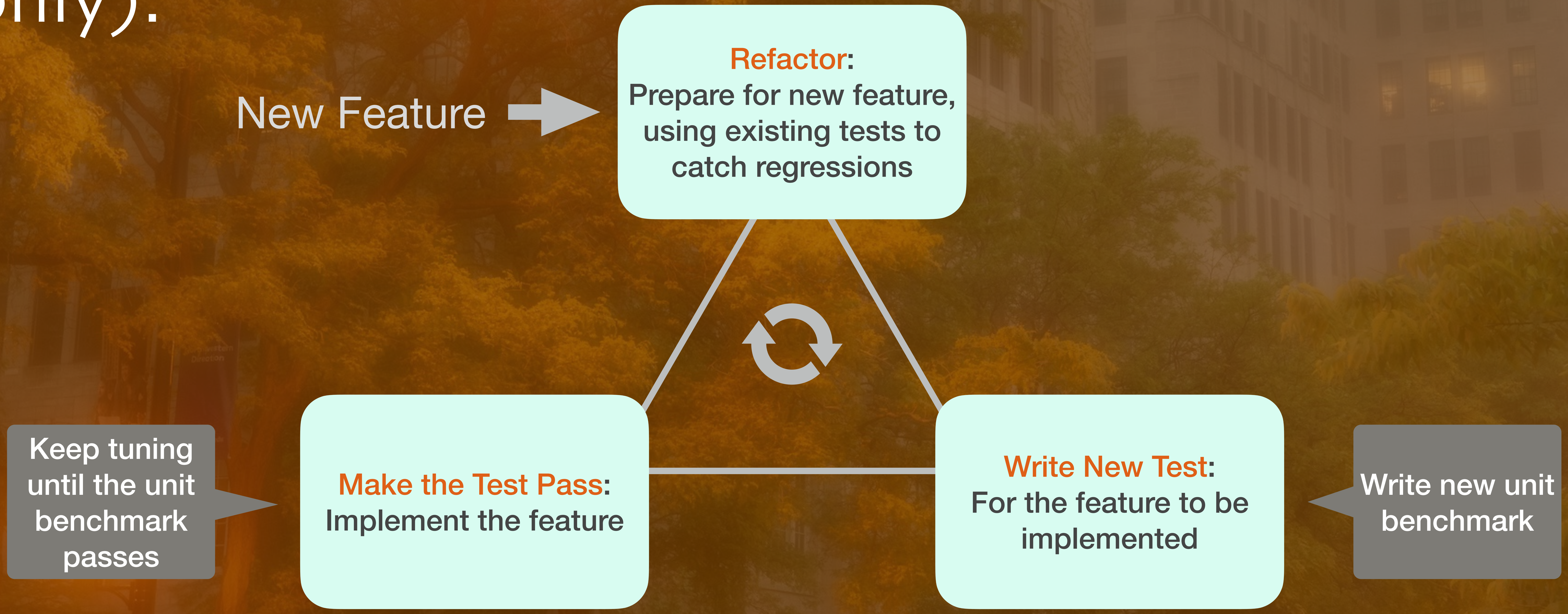
A new perspective?

★What if we switch from verifying desired model behavior to coercing desired behavior, instead?

- We already tune models to improve domain-specific knowledge, chatbot behavior, etc.
- So,
  - Tired: Writing software and testing it.
  - Wired: Tune until satisfactory behavior is achieved.

A new perspective?

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



New Feature

Refactor: Prepare for new feature, using existing tests to catch regressions

Write New Test: For the feature to be implemented

Write new unit benchmark

Make the Test Pass: Implement the feature

Keep tuning until the unit benchmark passes

Thank you!

thealliance.ai

dwampler@thealliance.ai

Mastodon and Bluesky: @deanwampler