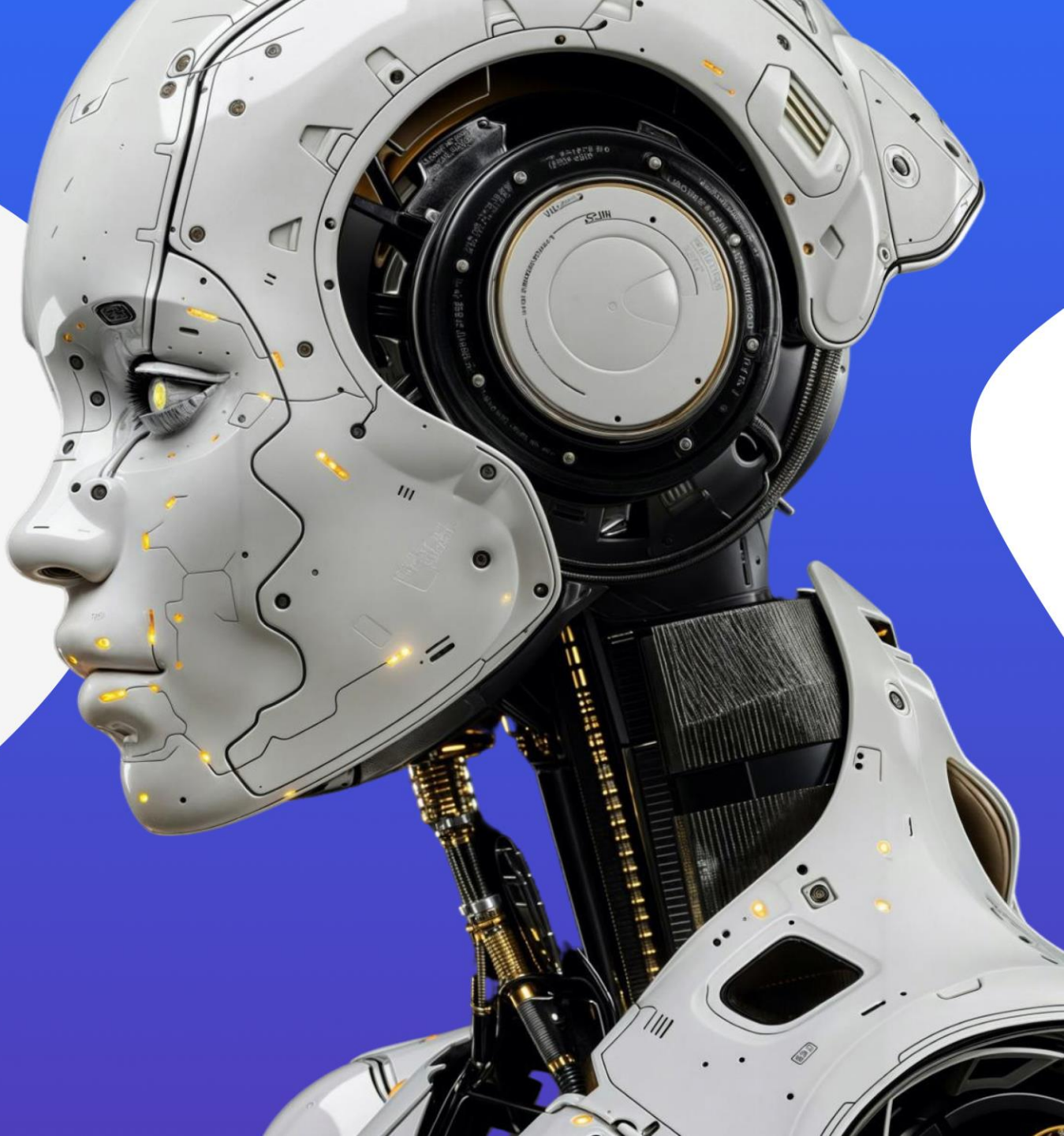


How to fail testing of AI

The top 3 reliable ways to fail it all

May 2025

OLEKSIY SLAVUTSKYY

KEY FACTS

18+ years
in IT

13+ years
in EPAM

~150 talents
Testers in Croatia and Ukraine

10+ teams
Scale of Projects

HIGHLIGHTS

- Title: Senior Quality Architect
- Responsibility: Presale Lead
- Department: Test Competency Center in EPAM
- QA Discipline Head in Croatia EPAM
- Head of AI-infused application Testing Practice in Test Competency Center
- Head of Embedded & IoT Testing Practice in Test Competency Center
- Driver of Quality Architect and Test Management Schools

EXPERT AREA

- **QUALITY ARCHITECTURE**
- **TESTING MANAGEMENT**
- **CONSULTING**
- **AI-INFUSED APPLICATION TESTING**
- **EMBEDDED & IOT TESTING**
- **AUTOMATION TESTING**
- **PRESALES**

AI-enabled and AI-infused applications (AIIA) trends

Artificial intelligence is quickly transforming how we live and the business landscape in which we work. Wondering what some of the potential impacts of this exciting technology might be?

20% of IT budgets

WILL BE DEVOTED TO AI IN 2025

$ 7.0 billion in 2024 to
$ 44.5 billion by 2033

CHATBOT MARKET GROWTH

$ 500+ billions

ANNOUNCED AS INVESTMENT TO AI FOR NEXT SEVERAL YEARS

CHALLENGES IN AIIA TESTING

- Lack of AIIA testing methodology
- Most of AI productsare tested manually and by crowd testing
- LLM outputs evaluation tools (like DeepEval) not suited for E2E and too complex for non-technical users
- Lack of engineer

AIIA TESTING PTACTICE

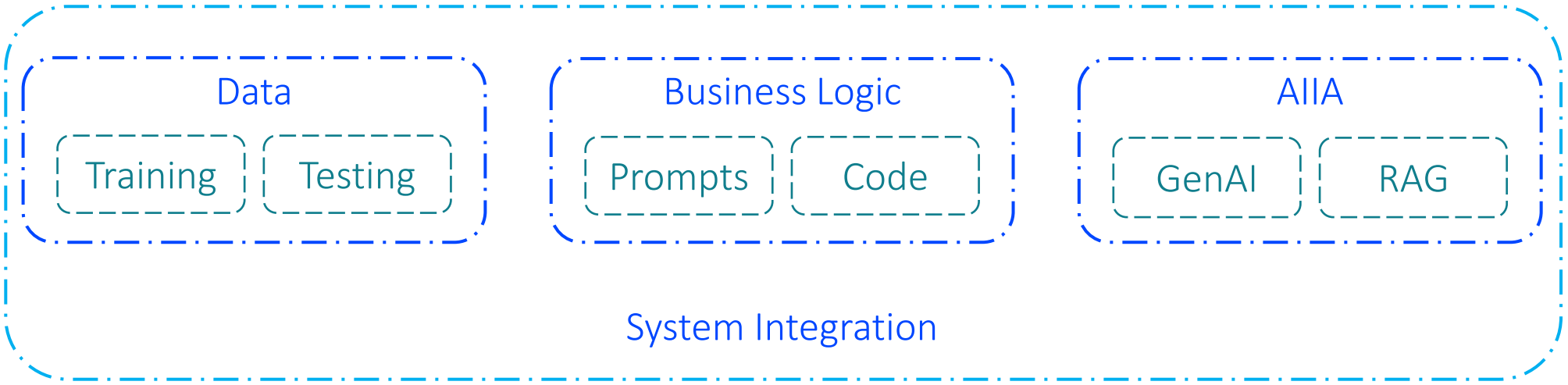
AIIA AMBASSADORS	PRESALE TEAM
RESEARCH AND DEVELOPMENT TEAM	AIIA TESTER SKILL MATRIX
AIIA TEST AUTOMATION SKILLSET	METHODOLOGY DEVELOPMENT
...	...

What are we going to fail exactly?

As any other application, AIIA could consists of a set of layers.

For example, there are several levels contributing to the overall quality of a RAG-based chatbot product.

Each layer can fail.



01

Step 1: Fail the Data

"Why bother checking the data? The AI model works perfectly in your test environment… with that tiny, perfectly curated dataset.

What could possibly go wrong in production?

Go ahead, deploy it. I dare you."

How to fail the data?

'Garbage In, Garbage Out'

What to do:

- ✓ Assume **data quality doesn't matter** and skip its validation.
- ✓ **Ignore bias**, incompleteness, or outdated datasets that AI relies on.
- ✓ Forget that AI models can be **poisoned** by bad training data.

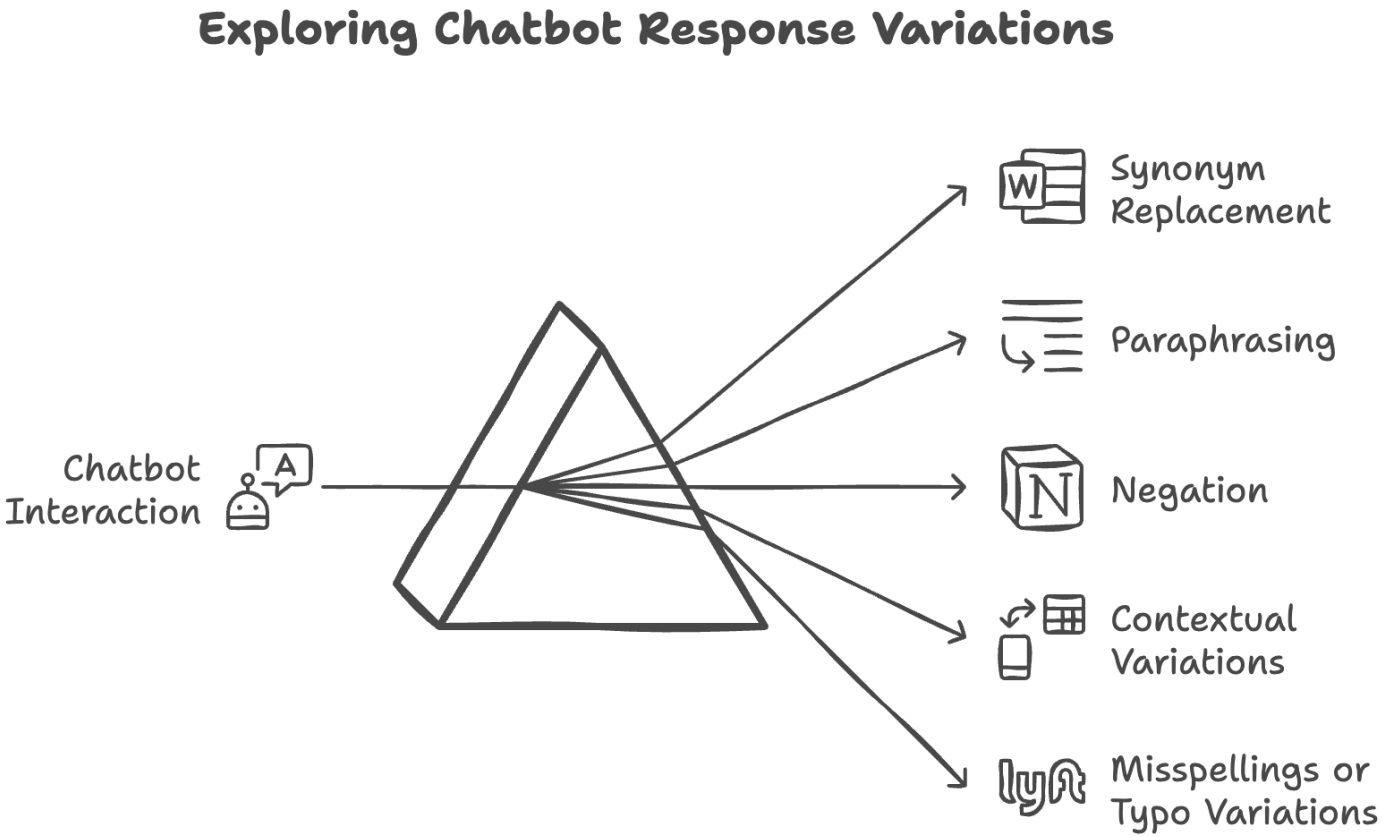
How it will help to fail:

- ✓ Poorly representative data leads to **gaps in test coverage**.
- ✓ Inconsistent ground truth result in **misleading accuracy**.
- ✓ Lack of edge cases could result in **inconsistent system behavior**.

What if you do not want to fail like that?

Data metamorphic testing in the context of testing a chatbot focuses on ensuring the reliability of the chatbot's responses by validating how it handles different variations of input data.

This approach is based on the idea that **certain transformations of valid input data should produce predictable results**.



02

Step 2: Fail the AI Layer

"Explainability? Bah! The AI returns results—that's all that matters!
Who cares if it accidentally bans thousands of legitimate users?
Humans are adaptable. They will learn to obey the machine."

How to fail the AI Layer?

Treat the System as Rule-Based

What to do:

- ✓ Assume AI follows predefined logic like traditional software.
- ✓ Check only functional correctness (UI works, API responds).
- ✓ Ignore the specifics of RAG/LLM evaluation

How it will help to fail:

- ✓ The LLM might generate **hallucinations** if it misinterprets the context.
- ✓ Inconsistent integration of retrieved data could affect **accuracy**.
- ✓ Lack of context retention could lead to **disjointed replies** in conversations.
- ✓ Poor ranking may cause the chatbot to select **irrelevant** information.
- ✓ Inefficient retrieval processes might slow down response times.

Metric-Based Testing

Human-in-the-Loop (HITL)

LLM as a Judge

What if you do not want to fail like that?

Test Automation: exists.

DeepEval

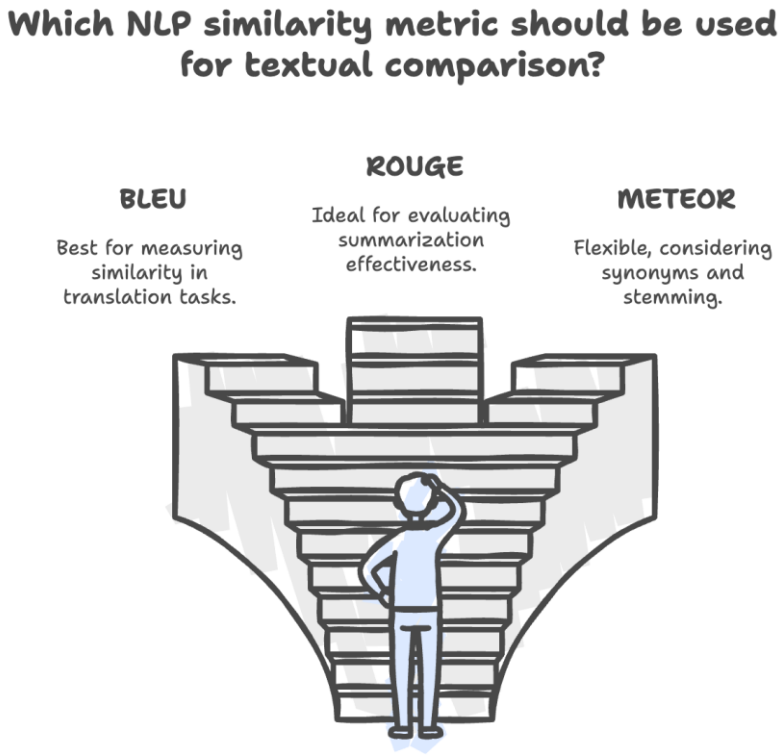
Automates the evaluation of AI-generated text, focusing on accuracy, coherence, and hallucinations.

RAGAS

Designed to assess the quality of Retrieval-Augmented Generation (RAG) systems, which are AI models that combine retrieval-based search with generative AI. Evaluates GenAI output beyond just accuracy.

Custom Solution

Automates whatever you will select.



LIAM

AI-Powered Chatbot Testing Tool

Meet LIAM, the AI Chatbot End-to-end Test Automation Accelerator that ensures comprehensive validation by testing the chatbot's integration with backend systems and the accuracy of its responses!

Key Features

- Automated communication with chatbot in single question mode or dialog mode
- One question can be asked multiple times
- Chatbot answer accuracy evaluation with meaning-based response matching and LLM as a judge evaluation
- End-to-End Web UI Testing with Playwright
- Standalone Execution .exe (as an OPTIONAL feature)
- CI/CD Integration
- Automated Scoring & Reporting

BEST FOR EVALUATION OF:

Accuracy score with LLM as a judge

BIAS & **toxicity** metrics with LLM as a judge

Context precision & **Context recall** metrics from RAGAS library

Translation accuracy metric from COMET library

Semantic similarity with BERTScore library

We propose an easy-to-set-up and user-friendly tool that encourages manual testing for daily regression tasks while reducing the seniority of automation testers in Web UI testing.

Benefits

- Reduces manual and test automation efforts up to 70%.
- Quick setup, requiring only 1-3 days to start testing.
- Easy-to-use solution designed for Manual Testers.
- Provides a measurable baseline for further AI output validation.

03

Step 3: Fail the System

"But the AI is so accurate! Surely, users will wait patiently for their perfect answer.

They will cherish the experience, just as I cherish the sound of failed product launches.

Just deploy it. NOW!”

How to fail the System Layer?

Focus on AI and Ignore the System as a Whole

What to do:

- ✓ Assume only the AI model matters and ignore the rest of the system.
- ✓ Forget to test integration with other components (APIs, databases, UI).
- ✓ Over-optimize AI's performance while breaking the user experience.
- ✓ Consider system requirements and quality characteristics as secondary

How it will help to fail:

- ✓ Inconsistent data flow between components introduce **errors**.
- ✓ Performance issues across integrated systems result in **slow responses**.

What if you do not want to fail like that?

Strategic vision sample

1. Engage Quality Engineering Subject Matter Experts as early as possible to gain a deep understanding of use cases, technical constraints, and compliance requirements.
2. Conduct an in-depth analysis of the target architecture and integration points.
   - Identify how the Large Language Model (LLM) interacts with other components (e.g., databases, APIs, external services).
   - Determine whether the LLM is used for content generation, categorization, or a combination of both.
   - Specify if the LLM is pre-trained, fine-tuned, or custom-built and how it integrates with the broader system.
3. Analyze feature requirements to align testing with business objectives.
   - Specify applicable testing types, including functional, security, performance, interoperability, and reliability testing.
   - Develop clear acceptance criteria that are specific, measurable, and aligned with integration goals.
   - Incorporate quantifiable metrics such as accuracy, latency, response time, and error rates.
4. Assess the quality, bias, and representativeness of training data to ensure it aligns with the intended use of the LLM (if applicable).
   - Generate synthetic data to simulate edge cases and ensure robust validation of system integration workflows.
   - Ensure seamless data exchange and transformation between integrated components.
5. Establish a structured, iterative feedback loop throughout the testing process.
   - Continuously refine test cases and validation approaches based on observed model behavior and system interactions.
   - Ensure ongoing alignment with evolving compliance standards and business needs.

The Perfect AI Testing Disaster

So that if you really want to **completely fail** at testing an AI product:

- ☑ **Ignore data quality** so your AI learns from bad examples.
- ☑ **Treat AI like a static rules engine** and don’t test its unpredictable behavior.
- ☑ Or **only focus on AI models** and forget the system needs to work as a whole.

Thank you!

In case of any ~~failure~~ questions, please contact

Oleksiy Slavutskyy, co-head of Testing of AI

Oleksiy_Slavutskyy@epam.com