Thoughts from #Som:

1. Use Generative AI to design test cases to test stressed candidate test load in the Certifiable system.(https://www.tricentis.com/products/automate-continuous-testing-tosca/copilot?utm\_source=google&utm\_medium=paidsearch&utm\_campaign=PMax\_High\_AMS\_EN&utm\_term=&gad\_source=1&gclid=CjwKCAiAwaG9BhAREiwAdhv6YwYmFgbaB3CpzwnOAJgXG2GbF192BeQu2s\_hfc3bmVfulNxekSHuQBoCpBgQAvD\_BwE)
2. Use Generative AI to generate example code about candidate registration process in certifiable (https://www.cursor.com/)
3. Add Guardrails ( <https://github.com/NVIDIA/NeMo-Guardrails>)

Question Answering over a set of documents (a.k.a. Retrieval Augmented Generation): Enforce fact-checking and output moderation.

To Prevent sharing of PII through LLM

Domain-specific Assistants (a.k.a. chatbots): Ensure the assistant stays on topic and follows the designed conversational flows. When the Software Architect uses LLM to get question answers and response.

LLM Endpoints: Add guardrails to your custom LLM for safer customer interaction - if the Exam is done via chatbot operation.

1. What are data sources examiner used to Check correctness of the answers?
2. Langfuse for Observability(https://langfuse.com/)

How many candidates taking test per day

How many tests passed

Slow responses in rendering the questions.

Average time to submit the Candidate status

1. LLM Prompt Injection Analysis
2. USE PDCA Cycles-
3. What are sources of Unreliable AI grader:

Model Limitations.

Unintended app use

Information leakage( Answer set)

Reputation Damage of Certifiable

Something Funny

What AI Engineers say vs what they actually mean:

- LLM Engineer

A Python developer who knows how to use the OpenAI API

- RAG

Throwing documents at an LLM and hoping it remembers the important parts

- Prompt Engineering

Writing increasingly complex instructions until the AI does what you want

- Fine-tuning

When basic prompting isn't working but you have a big GPU budget

- AI-First

We added ChatGPT to our product roadmap

- AI Governance

A collection of policies we wrote after something went wrong

- Context Window

The digital equivalent of short-term memory that's never big enough

- Vector Database

Fancy keyword search for your documents that we're paying premium prices for

- AI Ops

Regular DevOps but with more expensive GPU instances

- synthetic data

When you can't birth real data so you ask an AI to make some up