

# Testing The Future: How AI Is Reshaping QA





# A few questions

Who of you are developers?

Who of you are testers?

Who of you are both?

# Jens & Praegus





# Tom & Orangebeard



**ORANGEBEARD**





# Testing will no longer be a separate phase, but a natural part of creation.

- In other words, when application code is generated, tests will be generated along with it. This will mean that greenlighting the produced code will actually mean greenlighting passing tests as well.



# Intelligent agents and AI solutions will increasingly drive test automation.

- We are going to see more agents and A.I. solutions that will add test automation alongside produced code. Those that will generate tests along with produced code will see less regressions and will scale much better.



# The craft of testing will evolve, not disappear.

- Testing as we know it will not disappear, but it will be more granular, focused on manual testing of new functionality and code reviews. Instead of huge test suites that will be manually reviewed one by one, testers will mostly test changes, while regression testing will be a domain of automation.



# Test output becomes a quality driver

- LLM's enriched with tool calling and RAG can analyze defects, do RCA or detect defect clustering based on test output, application logging and observability metrics. This can be leveraged to learn which changes carry most risk or predict which tests are most likely to fail next time.



# AI assisted test creation example

```
def is_safe_temperature(readings, threshold):  
    """  
    readings: list of temperature sensor samples  
    threshold: maximum allowed temperature in °C  
    """  
  
    if not readings:  
        return True # assume safe if no data  
    avg = sum(readings[:3]) / 3 # use first 3 samples as quick filter  
    return avg < threshold
```



# Application and test runtime will become a huge challenge.

- The reason for that is, that large language models are trained on a large amounts of code, but they lack information on how the written code actually runs. Prediction: observability tools will be the missing piece in making A.I. produce really reliable code.



# The tester becomes a curator of quality.

- When code is generated using LLM, that generation is guided by a highly skilled tester and reviewed. This will help ensuring the quality of the tests.



# Skills of the future developer & tester.

- Works with adaptive, AI-driven systems
- Validates AI outputs, spots hallucinations, ensures fairness
- Designs prompts, orchestrates AI-driven code (and test) generation
- Translates AI insights into business impact



# Q&A / Closing thoughts

- AI won't replace people, but people who use AI the right way will replace people who don't.