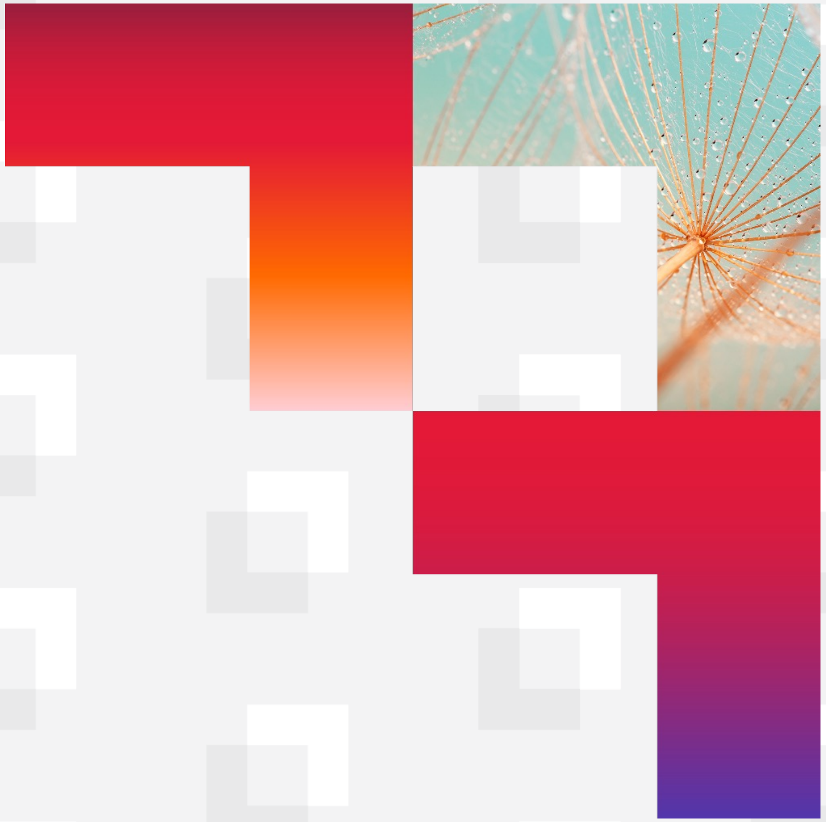
Talks and trainings

Quality engineering and testing

Maaret Pyhäjärvi

CGI



Talks on AI in 2024

Experiences of AI of Today in Testing
Sociotechnical Guardrails for AI-Driven Application Testing
Lessons Learned from Landing a Job Offer with GenAI
Know AI Threats to Focus on Success
Let's Do a Thing and Call it Foo

Talks on AI in 2025

Habit of AI in Quality Engineering
Exploratory Unit Testing with and for GenAI
Code Reviews Have Already Changed
A Herculean Experience
Exploring Test Automation Codebase With GenAI
RAGified Exploratory Notetaking

Trainings on AI in 2024 - 2025

Apply AI of Today on Your Testing (1 day)
Acceptance Factory (1 day)

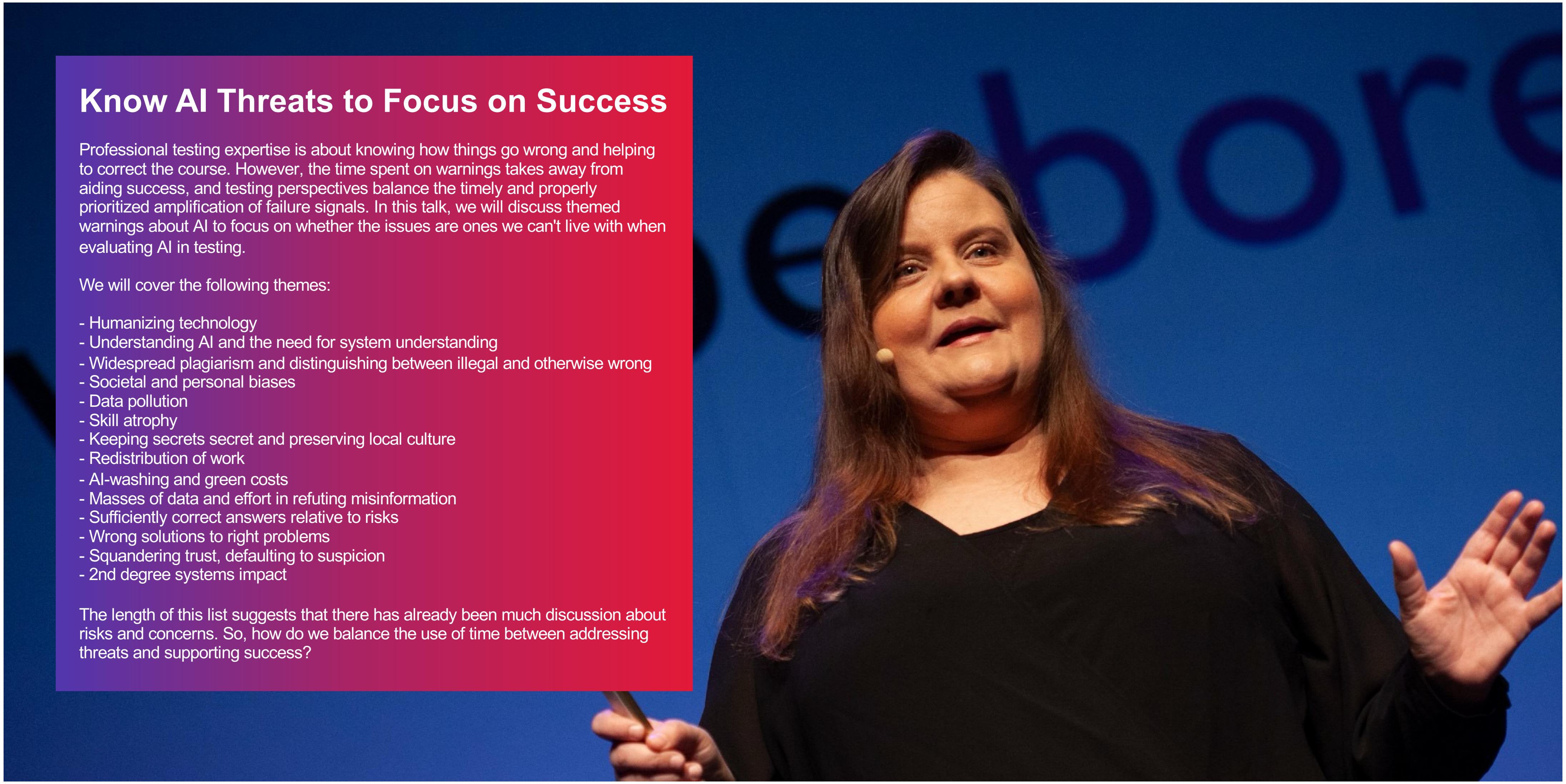
Talks

Sociotechnical Guardrails for AI-Driven Application Testing

With machine learning, large language models and retrieval augmented generation applied on text, voice, images and video, there's plenty of things AI out there. For 'acting humanly' framing to AI, any software doing things humans could do gets bundled in the new wave of AI-driven application testing. Armed with mindset of evaluation and amplifying fail signals, we are educated with great examples of the many ways these technologies fail. We have less education on the guardrails on how these technologies add to results of testing and productivity. We need the warnings to make sense if *we* are ok with *all* of this, yet the time to listen and deliver warnings is time away from figuring out the guardrails to succeed with this. Succeed with what is available to us all today, even if not yet perfectly packaged.

In this talk, we go through three demos, framed with a perspective of someone who cares about results of testing and value of testing perspective in software development. We have people around with these augmented intelligence solutions, and we need a conversation that is framed with the change of testing and curiosity, rather than fear. We need sociotechnical guardrails providing us useful solutions. Given there is a problem here, how do we work with AI so that we recognize and mitigate the problems? How do we move from value of fun to value at our work and discuss use of time in a frame of curiosity, building a future we can and want to be around? How would you use AI for it to be useful to you and your organization?

Time used on something is time away from something else. You'll use time listening to this, and chances are you're able to navigate the waters of AI-driven application testing a little better, today.



Know AI Threats to Focus on Success

Professional testing expertise is about knowing how things go wrong and helping to correct the course. However, the time spent on warnings takes away from aiding success, and testing perspectives balance the timely and properly prioritized amplification of failure signals. In this talk, we will discuss themed warnings about AI to focus on whether the issues are ones we can't live with when evaluating AI in testing.

We will cover the following themes:

- Humanizing technology
- Understanding AI and the need for system understanding
- Widespread plagiarism and distinguishing between illegal and otherwise wrong
- Societal and personal biases
- Data pollution
- Skill atrophy
- Keeping secrets secret and preserving local culture
- Redistribution of work
- AI-washing and green costs
- Masses of data and effort in refuting misinformation
- Sufficiently correct answers relative to risks
- Wrong solutions to right problems
- Squandering trust, defaulting to suspicion
- 2nd degree systems impact

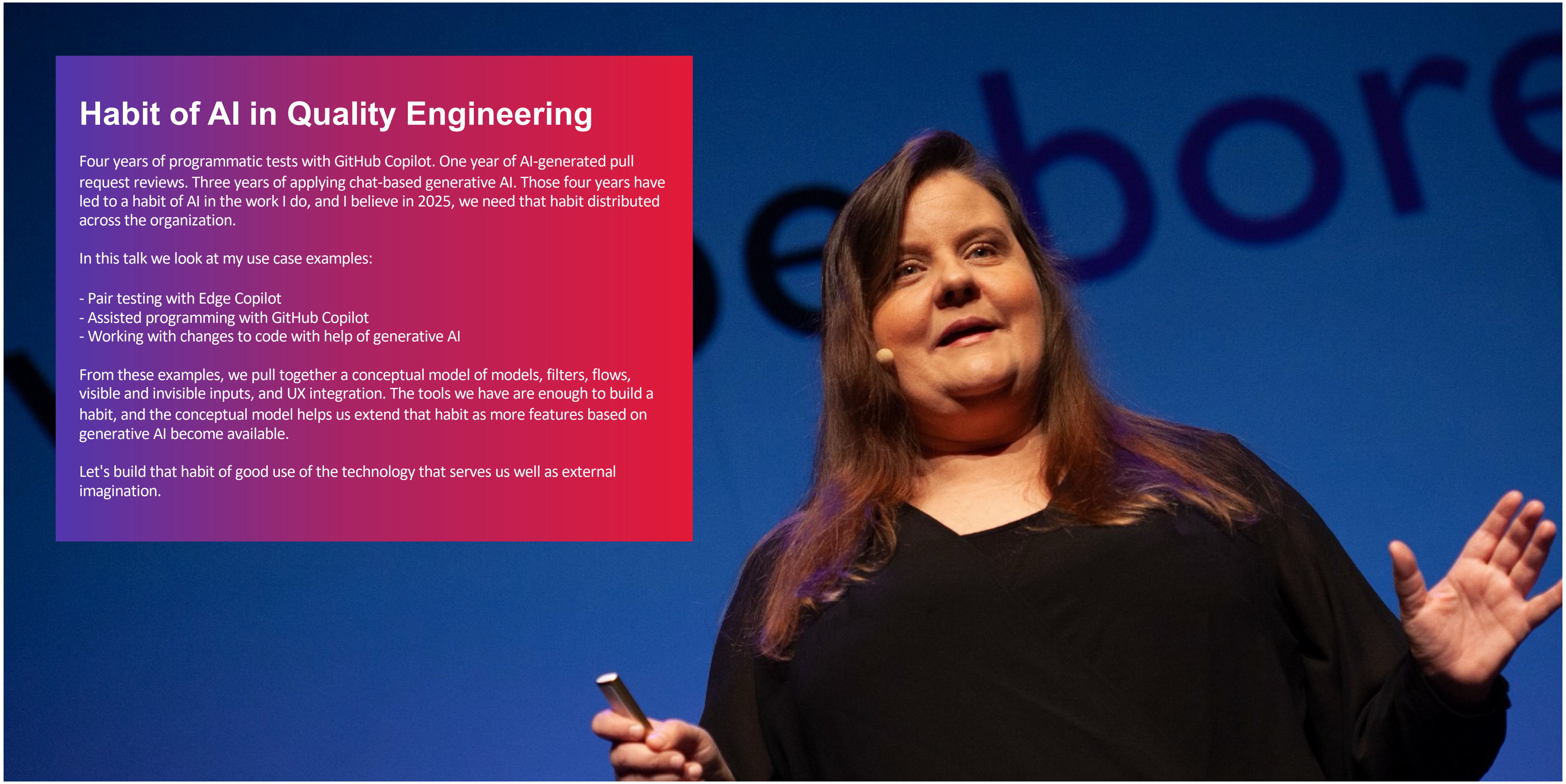
The length of this list suggests that there has already been much discussion about risks and concerns. So, how do we balance the use of time between addressing threats and supporting success?

Lessons Learned from Landing a Job Offer with GenAI

On first weeks of GitHub Copilot availability, I had a pair programming interview for a position I was considering. Armed with python and GitHub Copilot, I landed a job offer and lessons about simple programming problems and their testing in age of AI. In this talk, we start with revisiting the experience of pairing that makes a story with great takeaways on nature of testing. Then we build on that for two years of experiences in benefitting from GitHub Copilot in programmatic testing work.

Key takeaways:

- GenAI code generation allows you to turn a programming problem into a testing problem
- Testing problems are the hard problems
- GenAI code generation in IDE is a useful tool, with 2nd degree impacts to consider



Habit of AI in Quality Engineering

Four years of programmatic tests with GitHub Copilot. One year of AI-generated pull request reviews. Three years of applying chat-based generative AI. Those four years have led to a habit of AI in the work I do, and I believe in 2025, we need that habit distributed across the organization.

In this talk we look at my use case examples:

- Pair testing with Edge Copilot
- Assisted programming with GitHub Copilot
- Working with changes to code with help of generative AI

From these examples, we pull together a conceptual model of models, filters, flows, visible and invisible inputs, and UX integration. The tools we have are enough to build a habit, and the conceptual model helps us extend that habit as more features based on generative AI become available.

Let's build that habit of good use of the technology that serves us well as external imagination.

Trainings for hands-on testing

01 Apply AI of today in your testing

Benefits from AI to testing

- Pairing techniques for AI
- Integrating AI with testing tasks

1 day

02 Test case design

Techniques to better testing

- Test documentation
- Test techniques

2 days

03 Exploratory testing foundations

Explore and document with automation

- Thinking while testing
- Capturing ideas with documentation and automation

1 day

04 Python for testing

Programming for testing

- Supporting testing
- Automating testing

2 days

05 Exploratory testing on computer interfaces

Exploring on code and APIs

- Testing of code in code

1 day

06 Exploratory testing work course

Ideas coverage amplified

- Combining information to test for results

2 days

Other trainings

07 Acceptance testing

User acceptance testing

- Doing testing
- Planning and managing testing

1 day

08 Managing and leading for testing

Framing the right testing

- Planning and managing
- Strategic framing

2 days

09 Social software testing approaches

Paired and grouped work

- Learning to learn, unknown unknowns

1 day

10 Test process improvement

Assessing and improving testing

- Assessment models
- Options for practices

1 day

11 Test automation primer

Framing automation for success

- Selections of tools

1 day

12 Exploratory unit testing

Testing for the programmers

- Unit testing, test-driven development
- What should we test?

1 day

Other trainings

13 Agile testing

Testing practice in agile teams

- Agile process and role of testers in it

2 days

14 First steps to software testing

Getting started as new to testing

- What everyone must know of testing
- Experiencing testing by doing it

½ day

15 Containers and Infrastructure as Code

Test environments

- From preinstalled to ephemeral
- Understanding clients and servers

1 day

16 Architectures and modeling for testing

Context matters

- Product development
- IT integration
- Technology awareness

1 day

17 Exploratory security testing

Functional but vulnerable

- Security as controls
- Security as vulnerabilities

1 day

18 Exploratory performance testing

Working with scale of data and use

- Performance, load, stress and test
- Monitoring for performance
- Tooling options

1 day

Future-Proofing Trainings

01 Acceptance Factory

From Requirements to Acceptance

- Shift Left with Specifying with Examples
- Incremental Acceptance Testing
- Applying AI to create an Acceptance Factory

1 day

Tool specific

19 Robot Framework

5 days

20 Jira and XRay

2 days

21 Azure DevOps

2 days

22 Leapwork

5 days

23 UIPath

5 days

24 ISTQB Foundation Certificate

3 days

Trainings

Apply AI of Today on Your Testing

1-day training

We have our day-to-day work in testing. We have new technologies available, and a lot of information on those. We aspire of AI with computers acting humanly enough to pair with us in both attended and unattended fashion on our testing work, providing value. As creatures of habit, we need to build new habits in applying the technology pair in our daily work.

We have ML, LLM, and RAG. We have GitHub Copilot, ChatGPT and Hugging Face. On this course, we take those, learn to work with those from examples and wrap up by building our very own course AI assistant. Instead of the averaged-out idea that we all think of testing the average way, let's use RAG - Retrieval Augmented Generation - to create a pipeline to more input of instructions. While setting up to build our own, we understand what is available out of box, what sociotechnical guardrails we need and how to change our habits of building attended and unattended use of AI practice of testing.

Course is a hands-on learning course, and exercises are done in ensemble testing format where we rotate roles to contribute as mindshare of the participants. The experiential frame of exercises ensures minimal time for setup and a shared hands-on experience on how to apply AI on testing tasks.

This training day shows you:
- How to thoughtfully use other people's AI tools in your work
- How to break down testing tasks for useful results in pairing with AI
- How to build a habit of using AI tools to enhance your testing activities
- How to build a pipeline from your knowledge to the AI version of you

Sovella nykypäivän tekoälyä testauksessasi

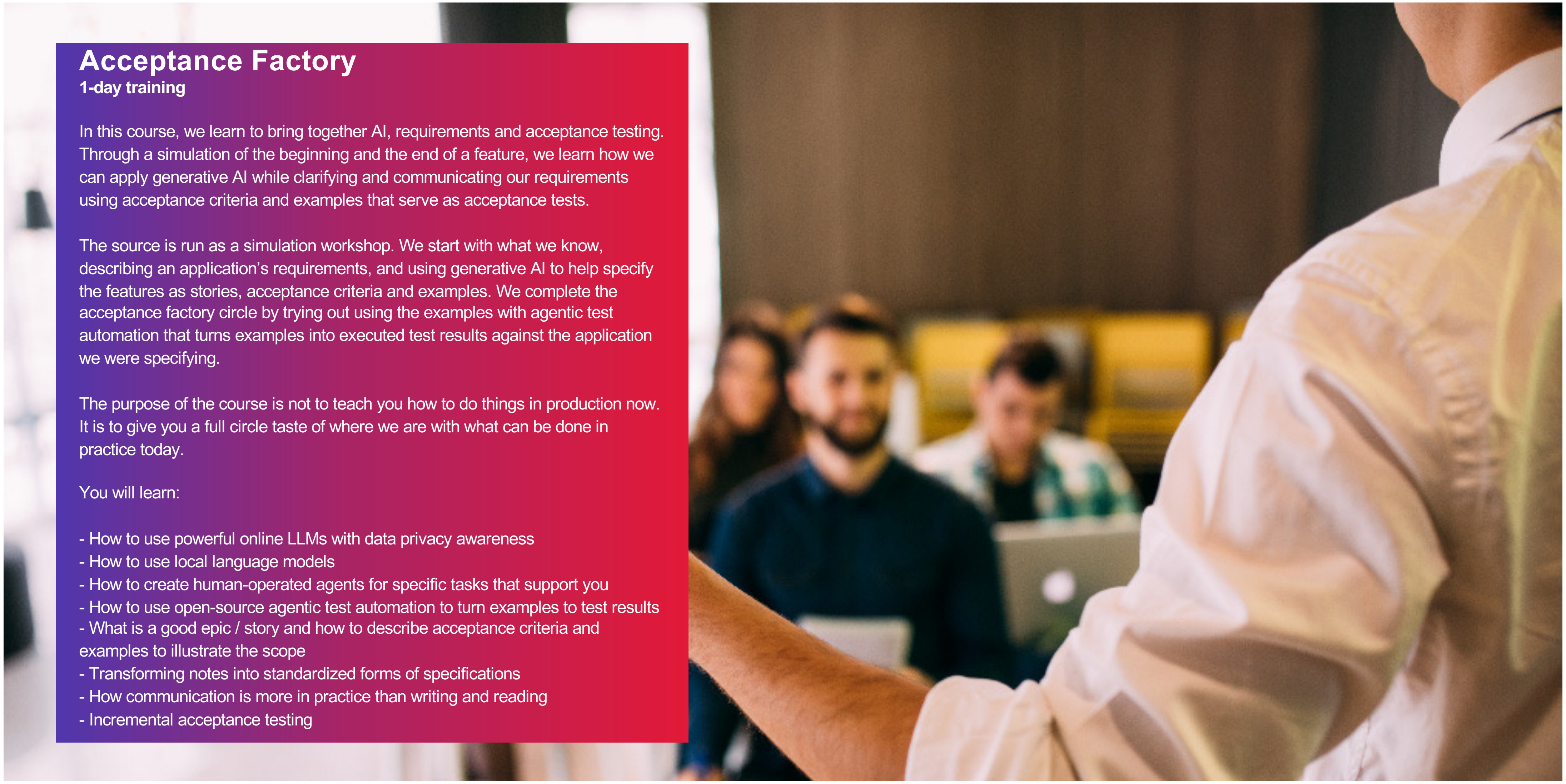
1-pv koulutus

Päivittäisessä testausarjessamme meillä on käytettävissämme uusia teknologioita ja paljon tietoa niistä. Haaveilemme tekoälystä, joka toimisi tarpeeksi inhimillisesti voidakseen työskennellä kanssamme testaustoiminnassa, sekä valvotusti että valvomatta, tuottaen arvoa. Työkalut ovat saatavilla, mutta meillä on opittavaa sekä niiden käytössä että omissa tavoissamme voidaksemme hyödyntää tätä teknologiaparia päivittäisessä työssämme.

On koneoppimista (ML), laajoja kielimalleja (LLM) ja RAG (Retrieval Augmented Generation). On GitHub Copilot, ChatCGP ja Hugging Face. Tällä kurssilla otamme nämä työkalut käyttöön, opimme työskentelemään niiden kanssa esimerkkien kautta ja lopuksi rakennamme oman kurssilaisten tekoälyavustajan. Sen sijaan että avustajamme olisi keskimääräinen kaikesta maailman tiedosta opetettu, annamme sille omaa materiaaliamme ja luomme putken omista teksteistämme avustajaa ohjaamaan.
Rakentaessamme omaa avustajaa, opimme näkemään mitä valmiita ratkaisuja on saatavilla, mitä sosioteknisiä turvarajoja tarvitsemme, ja kuinka muutamme tapojamme hyödyntääksemme tekoälyä sekä sen valvotussa että automatisoiduissa muodossa testauksessa.

Kurssi on käytännönläheinen oppimiskurssi, ja harjoitukset tehdään ryhmätestausmuodossa, jossa osallistujien rooleja kierrätetään. Harjoitusten tukena käytettävä kokemusperäinen rakenne mahdollistaa ajankäytön kohdistumisen yhteiseen käytännön kokemukseen tekoälyn soveltamisesta testaustehtäviin.
Tämä koulutuspäivä opettaa sinulle:

- Kuinka käyttää muiden kehittämiä tekoälytyökaluja harkiten omassa työssäsi
- Kuinka jakaa testaustehtävät osatehtäviin hyödyllisten tulosten saamiseksi tekoälyparin kanssa
- Kuinka rakentaa tapa käyttää tekoälytyökaluja testausaktiviteettiesi parantamiseksi
- Kuinka rakentaa putki omasta tiedostasi tekoälyversioon sinusta itsestäsi



Acceptance Factory

1-day training

In this course, we learn to bring together AI, requirements and acceptance testing. Through a simulation of the beginning and the end of a feature, we learn how we can apply generative AI while clarifying and communicating our requirements using acceptance criteria and examples that serve as acceptance tests.

The source is run as a simulation workshop. We start with what we know, describing an application's requirements, and using generative AI to help specify the features as stories, acceptance criteria and examples. We complete the acceptance factory circle by trying out using the examples with agentic test automation that turns examples into executed test results against the application we were specifying.

The purpose of the course is not to teach you how to do things in production now. It is to give you a full circle taste of where we are with what can be done in practice today.

You will learn:

- How to use powerful online LLMs with data privacy awareness
- How to use local language models
- How to create human-operated agents for specific tasks that support you
- How to use open-source agentic test automation to turn examples to test results
- What is a good epic / story and how to describe acceptance criteria and examples to illustrate the scope
- Transforming notes into standardized forms of specifications
- How communication is more in practice than writing and reading
- Incremental acceptance testing