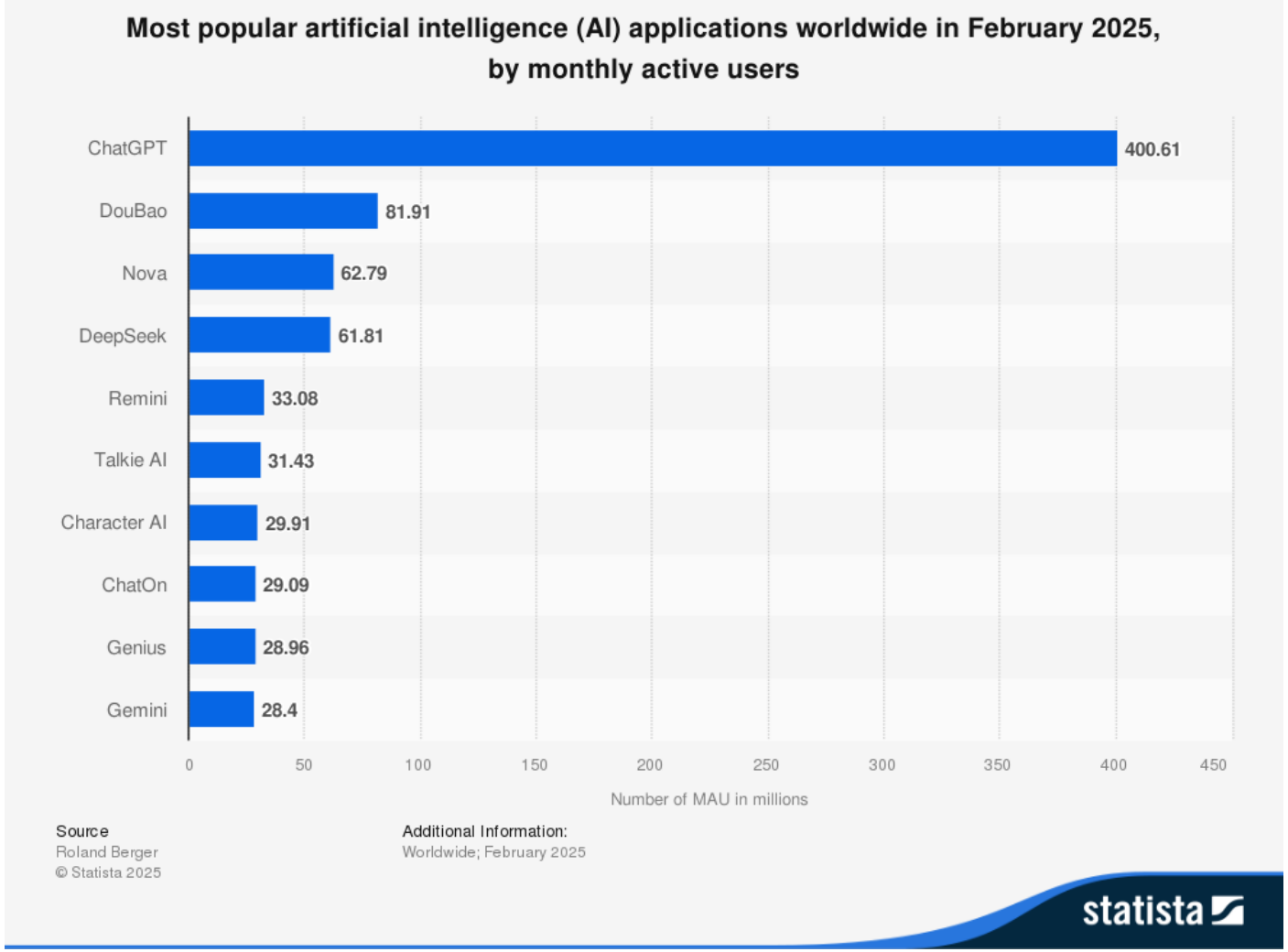
TECHNISCHE UNIVERSITÄT BERLIN

Navigating the growing field of research on AI for software testing

The taxonomy for AI-augmented software testing and an ontology-driven literature survey

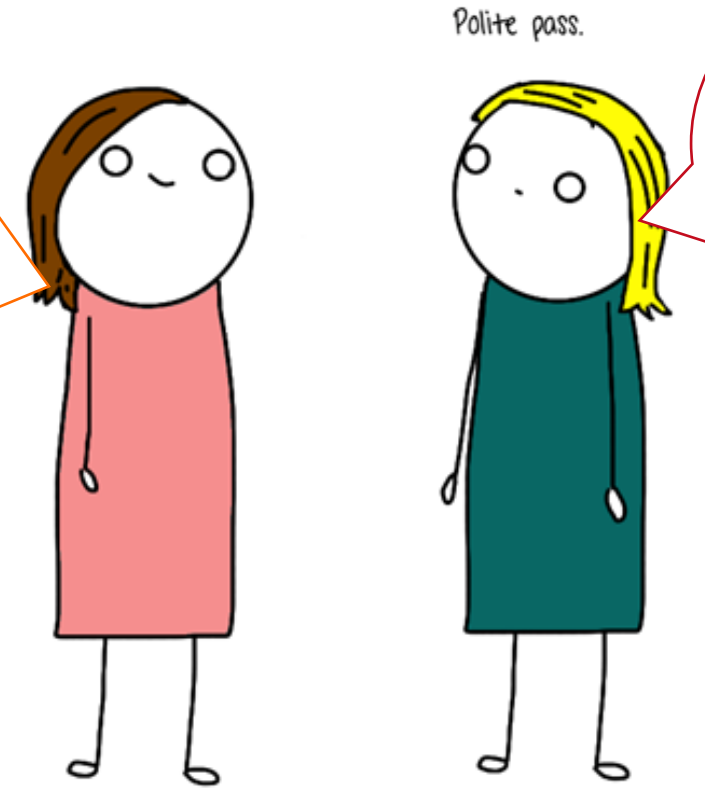
Prof. Dr.-Ing. Ina Schieferdecker | FMICS 2025, Aarhus, Aug 27, 2025

Already enthusiastic about GenAI?



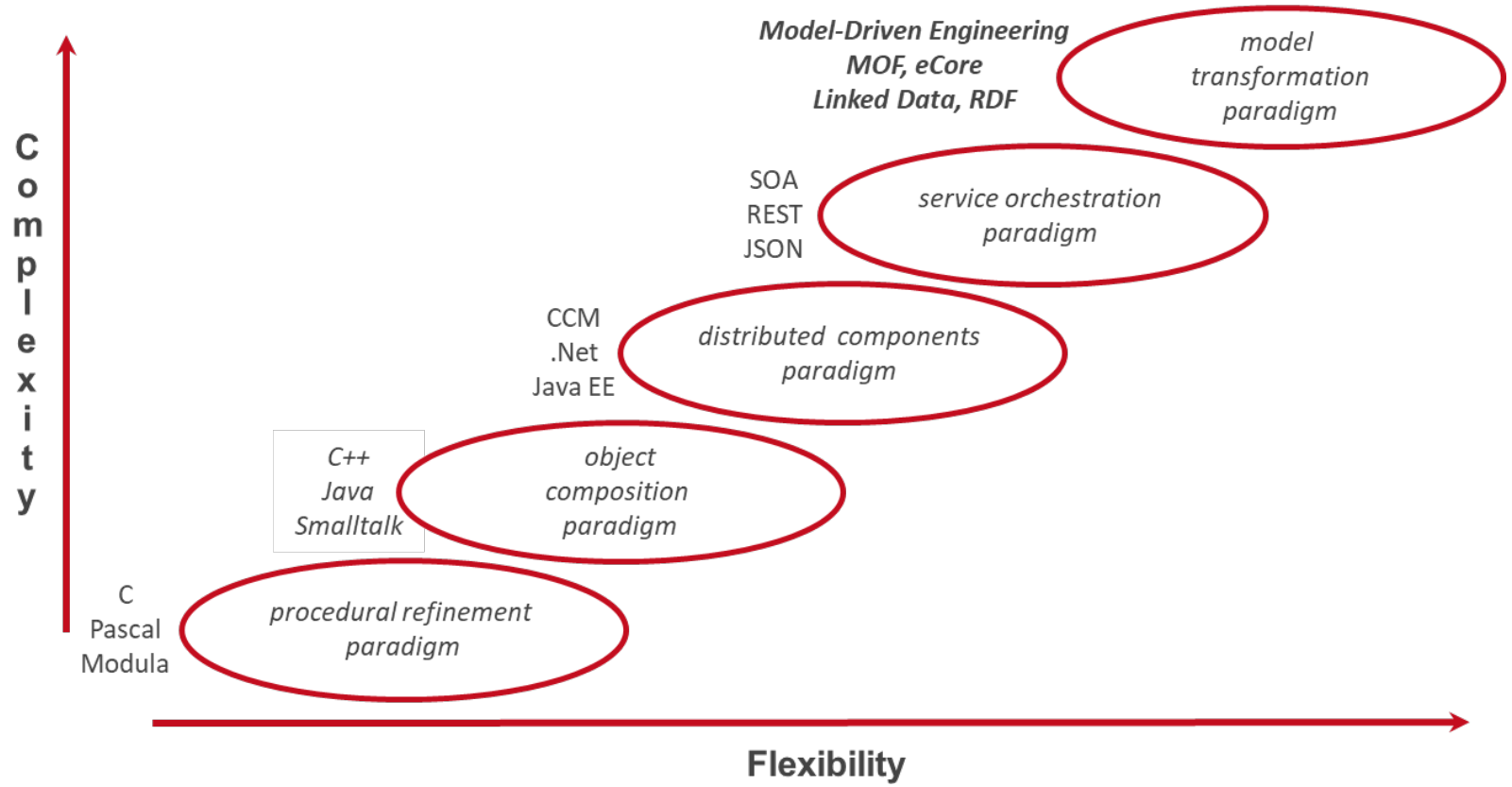
https://www.statista.com/statistics/1609163/top-ai-applications-mau-worldwide/

I am currently getting to know current programming frameworks with great pleasure and am enthusiastic about the possibilities with modern programming with AI.



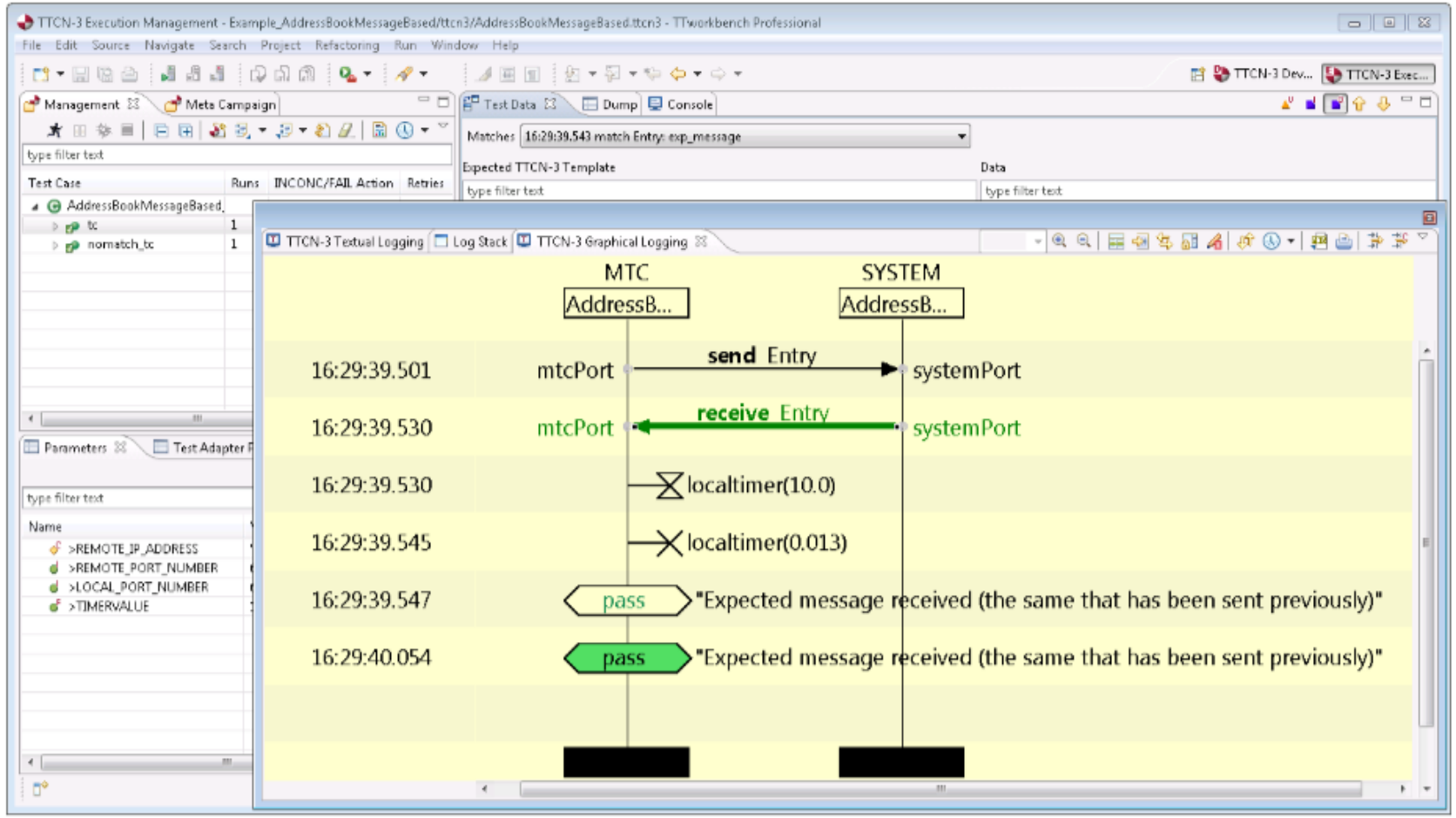
Why is that? Software is now programmed (and tested) by AI.

Software engineering so far



My background: Technology for test execution

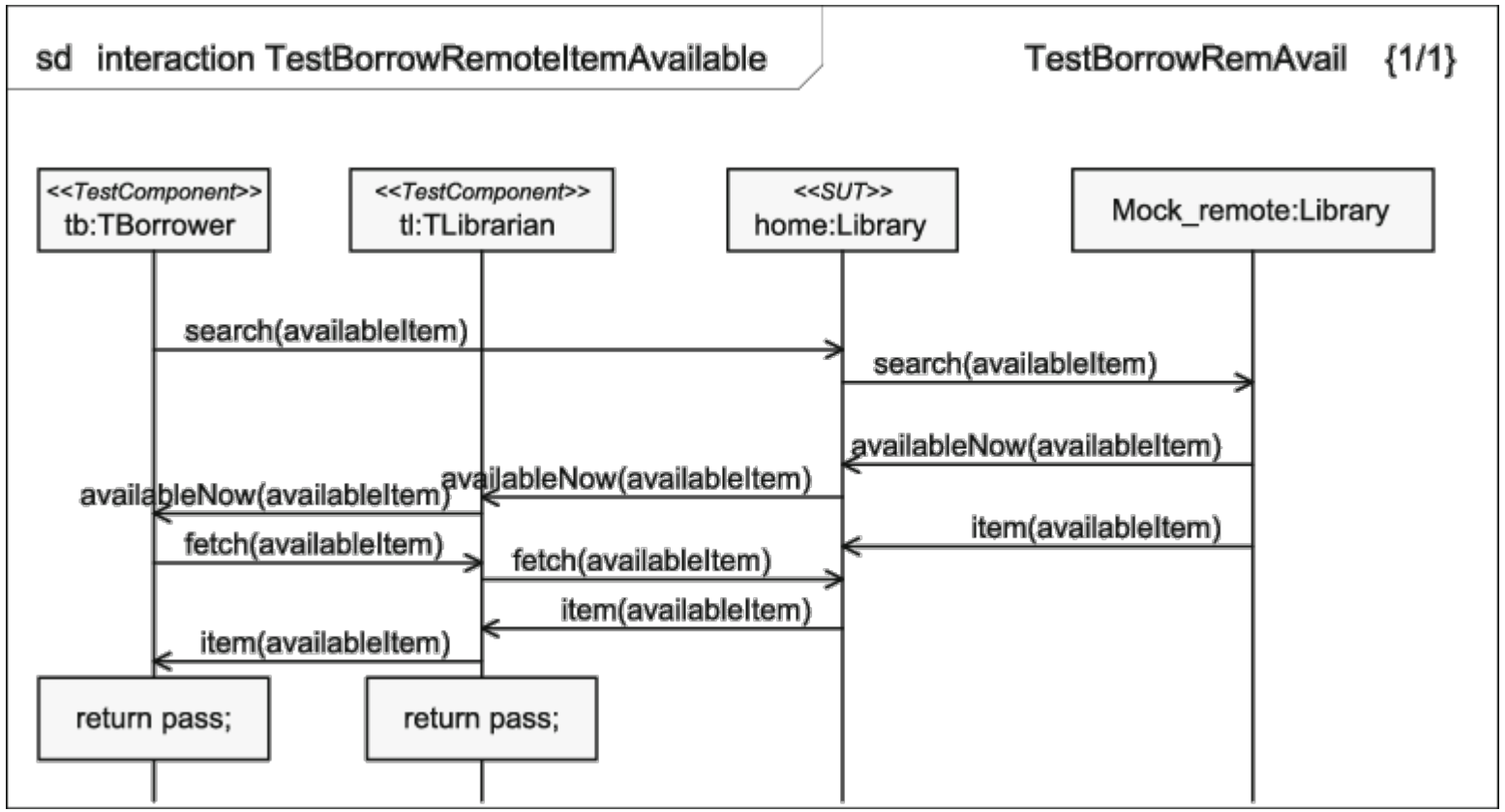
Testing and Test Control Notation



Grabowski, J., Hogrefe, D., Réthy, G., Schieferdecker, I., Wiles, A., & Willcock, C. (2003). An introduction to the testing and test control notation (TTCN-3). Computer Networks, 42(3), 375-403.

My background: Technology for test modelling

UML Testing Profile



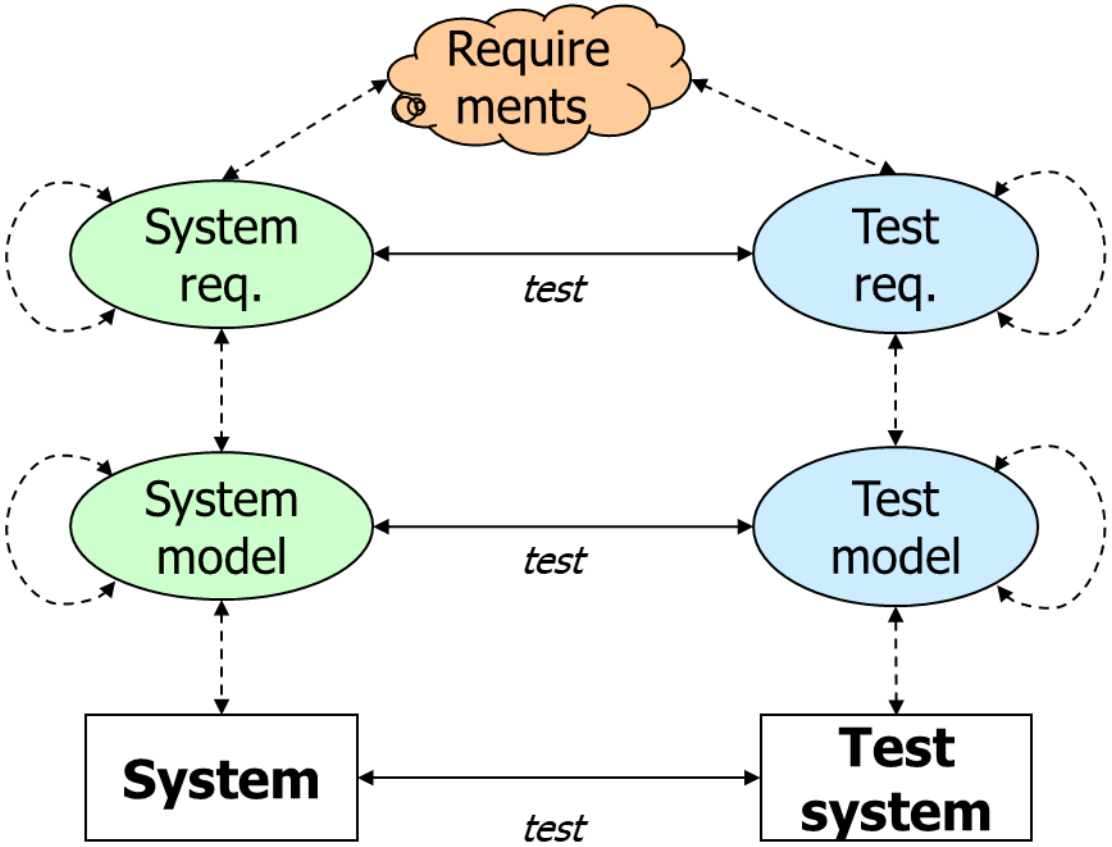
UML TESTING PROFILE

OBJECT MANAGEMENT GROUP®

Baker, P., Dai, Z. R., Grabowski, J., Schieferdecker, I., & Williams, C. (2007). Model-driven testing: Using the UML testing profile. Springer Science & Business Media.

My background: Technology for test generation

Model-based testing v3

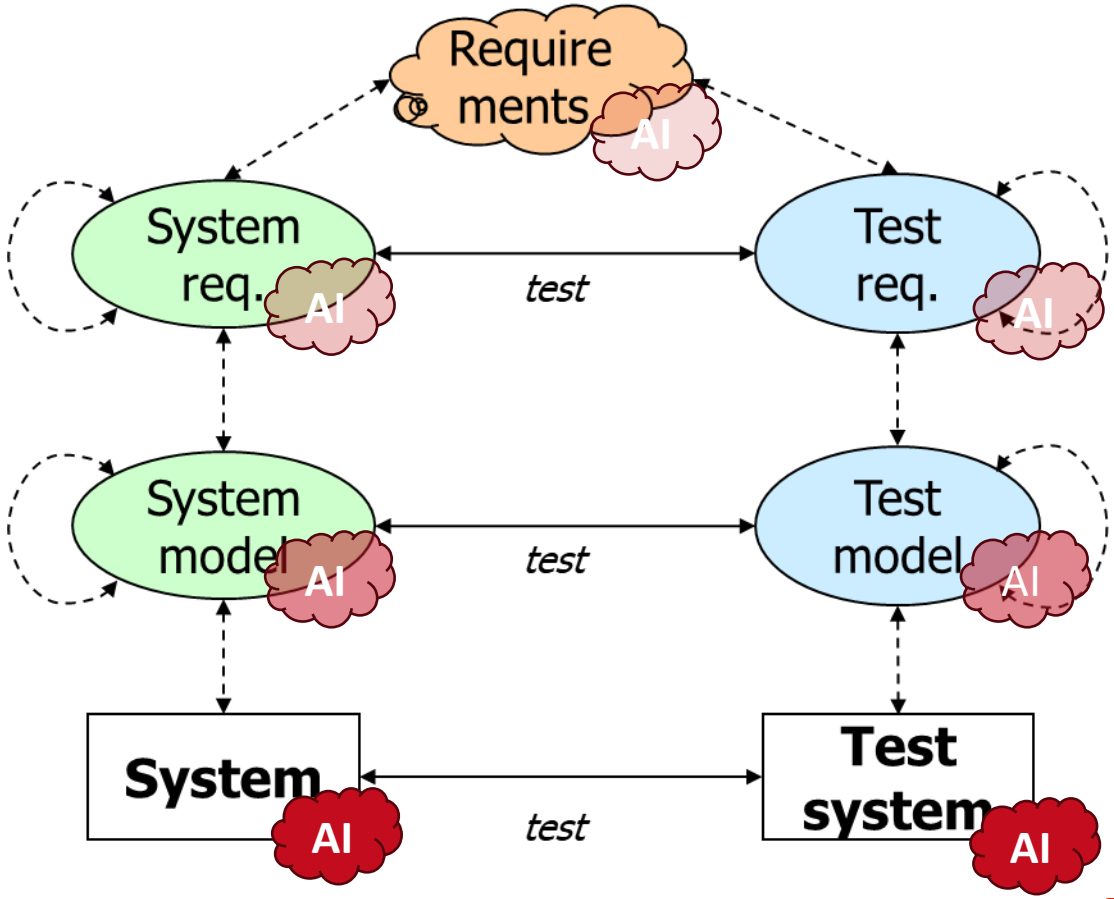


ISTQB Certified Tester Model Based Tester

ISTQB International Software Testing Qualifications Board

Schieferdecker, Ina. "Model-based testing." IEEE software 29.01 (2012): 14-18.

Software testing in the AI era



Schieferdecker, Ina. "Automated Software Testing in the AI Era." Keynote at AST 2024 at ICSE 2024, Lisboa, Portugal, April 2024..

Outline

① Software engineering with AI

② Testing with AI

③ The taxonomy

④ Wrapup

Outline

① Software engineering with AI

② Testing with AI

③ The taxonomy

④ Wrapup

AI application areas

Sep 17, 2024: Sopra Steria Next

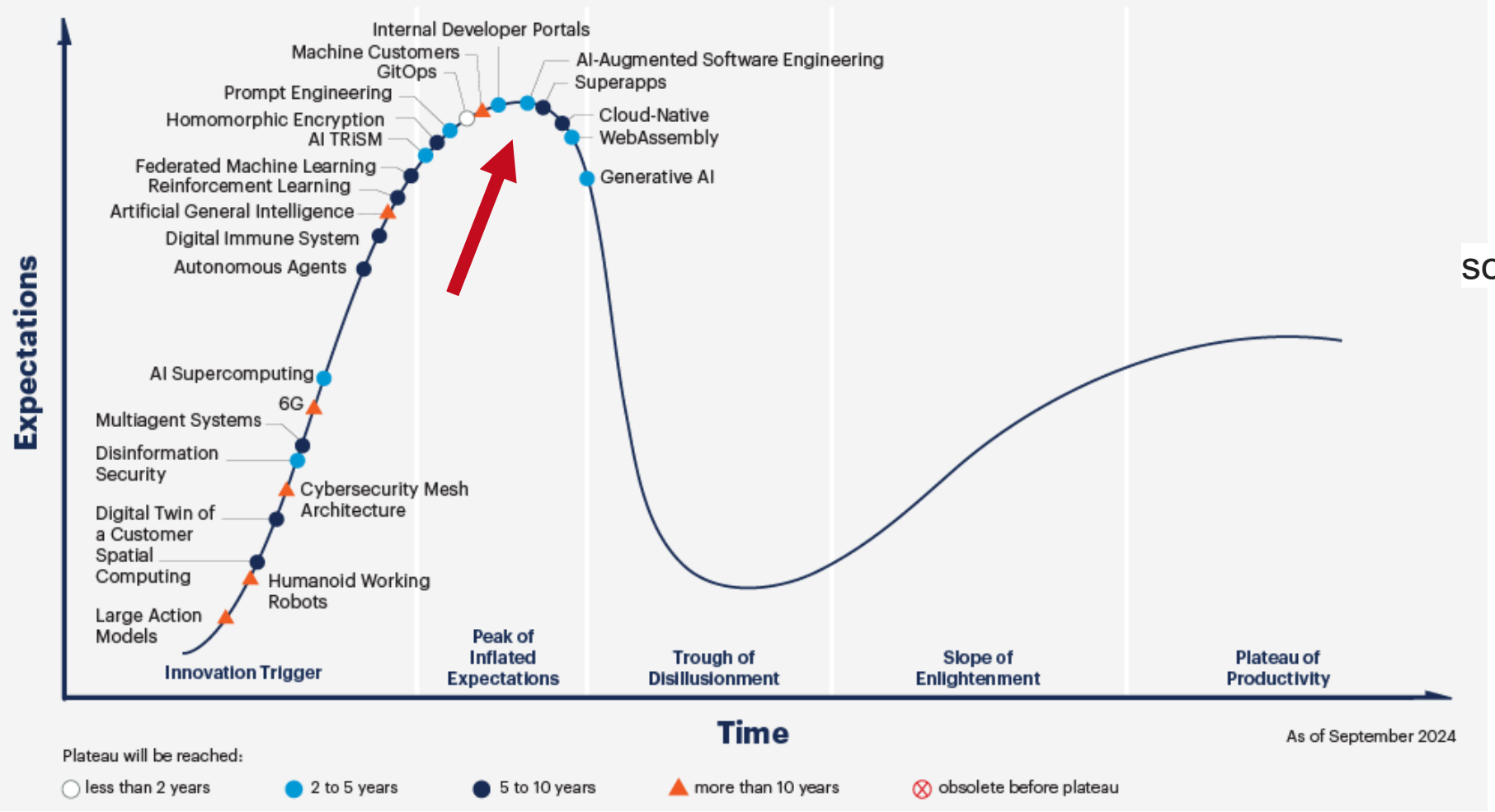
- Market analysis based on four AI categories: AI for Machines, Processes, Humans, Software
- Predicts global AI market will double to $1,270 billion by 2028 with an annual increase of 19 %

AI for *Software*

- Greatest potential together with AI for People
- Developing primarily in the financial services, healthcare, e-commerce and media domain
- Supporting not only programmers, but enabling even non-experts to write complex software

https://www.soprasteria.co.uk/insights/press-releases/details/sopra-steria-next-predicts-ai-market-will-double-to-1-270-billion-by-2028

AI-augmented software engineering, 2024



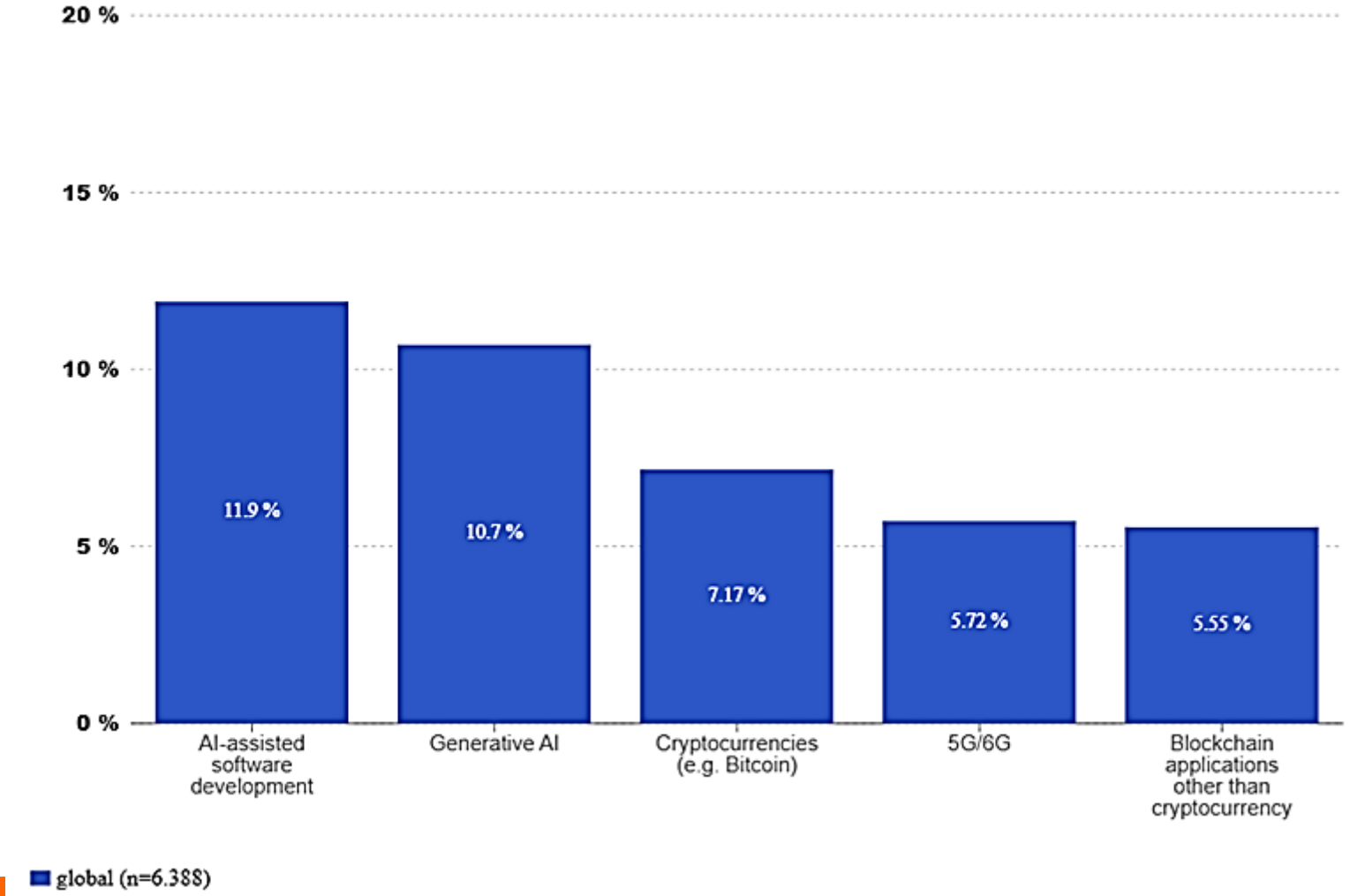
AI practices and platform engineering will reach mainstream adoption in software engineering in two to five years

By 2027, 50% of enterprise software engineers will use machine learning-powered coding tools

https://www.gartner.com/en/articles/hype-cycle-for-emerging-technologies

Areas of software development, 2024

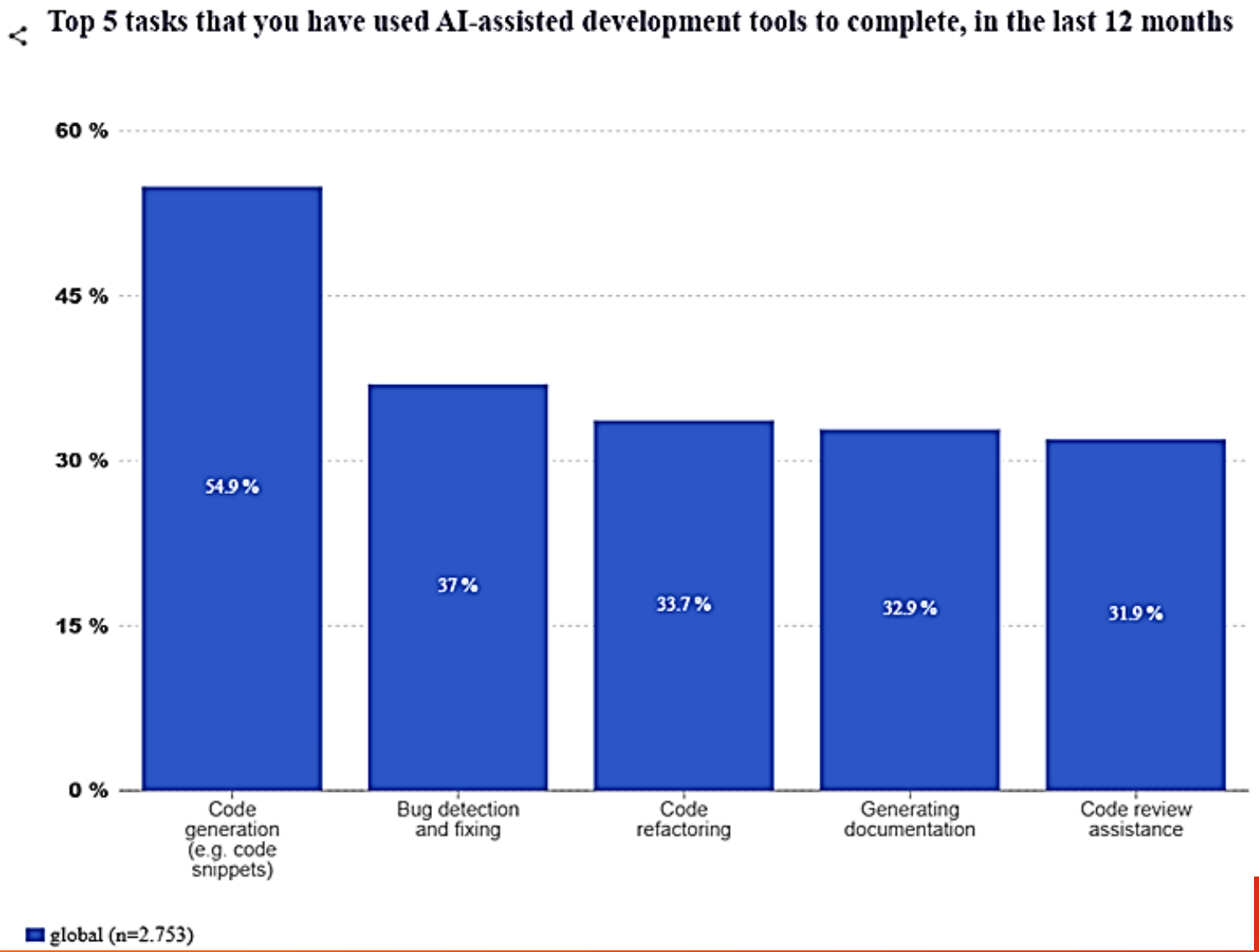
Which of the following areas are developers actively working on?



Developer Nation is a global developer community helping software creators grow throughout their coding journey

https://www.developernation.net/developer-reports/dn26/

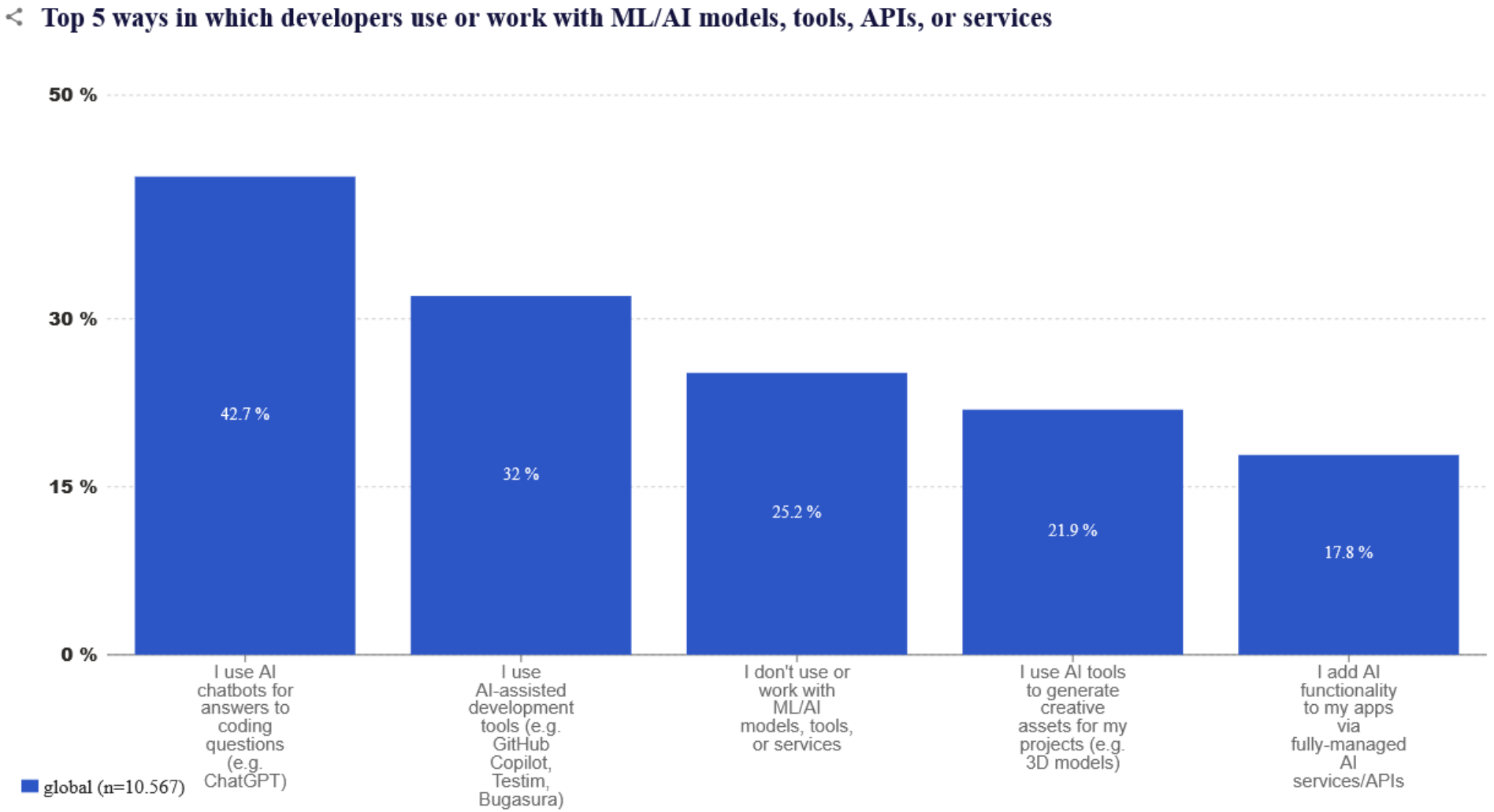
AI-assisted development tools to complete tasks, 2024



One third replied.

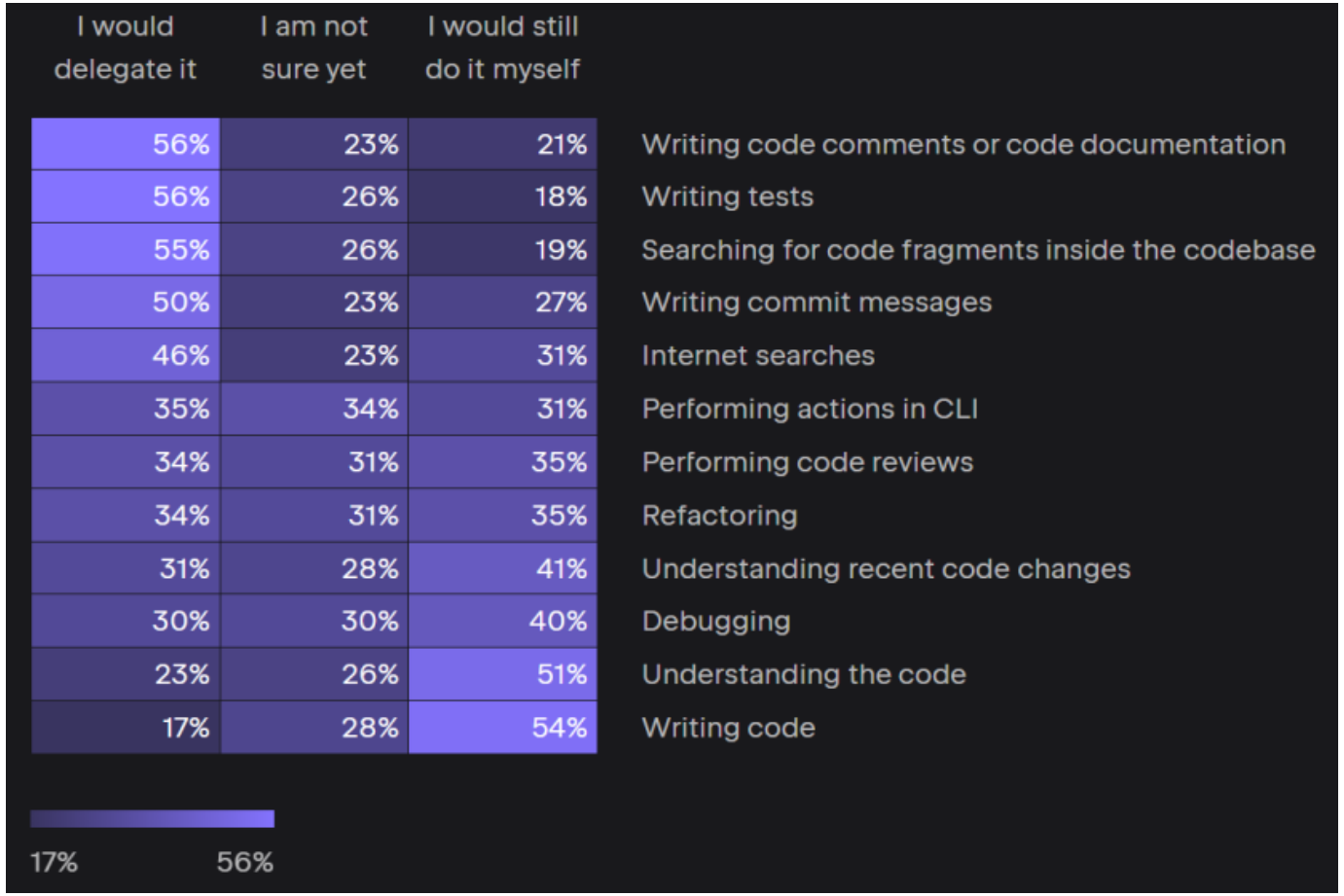
https://www.developernation.net/developer-reports/dn26/

Ways to work with ML/AI models, 2025



https://www.developernation.net/developer-reports/dn29

JetBrains top areas for AI in SDLC, 2023



I would delegate it	I am not sure yet	I would still do it myself	
56%	23%	21%	Writing code comments or code documentation
56%	26%	18%	Writing tests
55%	26%	19%	Searching for code fragments inside the codebase
50%	23%	27%	Writing commit messages
46%	23%	31%	Internet searches
35%	34%	31%	Performing actions in CLI
34%	31%	35%	Performing code reviews
34%	31%	35%	Refactoring
31%	28%	41%	Understanding recent code changes
30%	30%	40%	Debugging
23%	26%	51%	Understanding the code
17%	28%	54%	Writing code

https://www.jetbrains.com/lp/devecosystem-2023/

Outline

① Software engineering with AI

② Testing with AI

③ The taxonomy

④ Wrapup

What is software testing

ISO/ IEC/IEEE 29119-1: Software testing is the process of evaluating a system or its components(s) with the intent to find whether it satisfies the specified requirements (which represents the needs of stakeholders). It also identifies differences between actual and expected results and/or processes to evaluate features of the software item.

ISTQB Glossary v4.6.2: testing - The process within the software development lifecycle that evaluates the quality of a component or system and related work products.

- Software testing is used for the verification (adherence to requirements/specification) and validation (fit for purpose) of software, its components, and of software-based systems.
- Whether the quality of a software is ready / sufficient for deployment is determined by software testing.

Software testing dimensions (an extract)

- Test objects
  - kinds of test objects: application, operating system, (cloud) service, protocol, …
  - levels of test objects: unit, component, integration, system, system of systems
  - domain of test objects: IT, telecom, automotive, automation, avionics, medical, …
- Test types
  - functional, conformance, interoperability
  - security, safety
  - performance, load, stress
  - useability, accessibility
  - acceptance
- Test processes
  - crowd testing, continous testing, …
  - ad hoc – systematic
- Approaches
  - white-box – grey-box – black-box test design
  - static analysis – dynamic test execution
  - single target – back-to-back – A/B-testing
  - requirements-based, specification-based, model-based, product-line-based, risk-oriented test design
  - manual – scripted – automated – hybrid testing
- Test generation methods
  - exploratory
  - data-oriented (e.g. equivalence partitioning, pairwise)
  - behaviour-oriented (e.g. scenario-based, automaton-based, logic-based)
  - AI-based
- Test tools
- …

What software testing is not

- Debugging
  - used to find errors that cause tests to fail
  - close relation in failure / root cause analysis

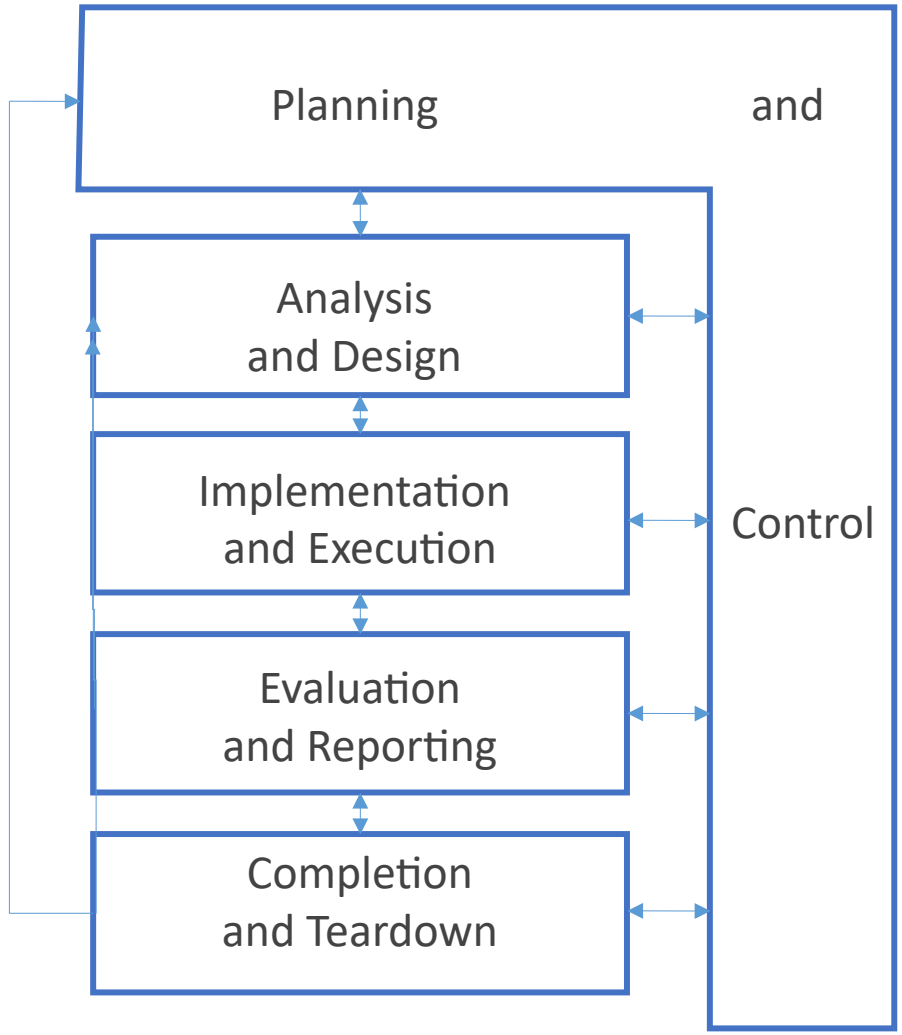
- Formal verification
  - used to prove the correctness / incorrectness of software components
  - close relation with model checking for testing

- Simulation
  - used e.g. to virtually execute / mimick software designs and software usages
  - relation with dynamic test execution

Software test automation

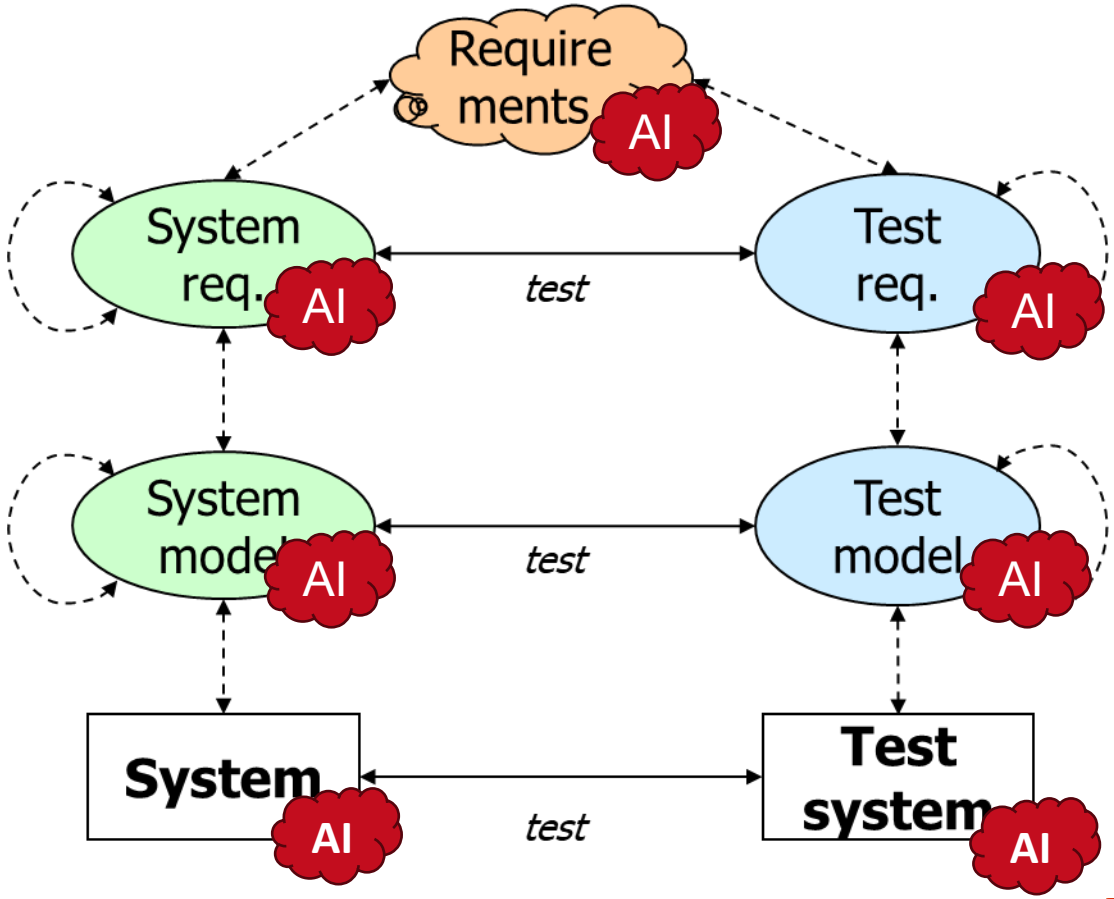
- Automation of
  - Test process / management
  - Test documentation
  - Test optimization
  - Analysis of test outcomes
  - Test execution
  - Test generation
  - Test design

  by tools and frameworks



The Fundamental Test Process

Software testing in the AI era



Revision of Schieferdecker, Ina. "Automated Software Testing in the AI Era." Keynote at AST 2024 at ICSE 2024, Lisboa, Portugal, April 2024..

ChatGPT 5: Is AI able to test itself? – Aug 2025

✅ What AI can do when testing itself

- Automated self-checks: AI models can run validation sets, regression tests, or unit tests to ensure they behave consistently over time.
- Self-monitoring: Some AI systems monitor their own outputs for anomalies or errors (e.g., confidence scoring, consistency checks).
- Fuzz testing & auto-generation of test cases: AI can generate unusual or random inputs to stress-test itself.
- Reinforcement learning: In some cases, AI can learn by trial and error, effectively "testing" different strategies and correcting itself.

❌ **What AI cannot do (at least today)**

- Defining what's correct: AI can check consistency, but it struggles to decide if outputs truly meet user intent or business requirements.
- Understanding context: AI doesn't inherently understand domain-specific goals, ethics, or long-term impacts of its behavior.
- Detecting hidden assumptions: Many bugs come from unclear requirements or human misunderstandings—AI won't notice those without guidance.
- Testing for fairness, bias, or usability: AI can analyze patterns, but human judgment is still essential for deciding what's fair, safe, or user-friendly.

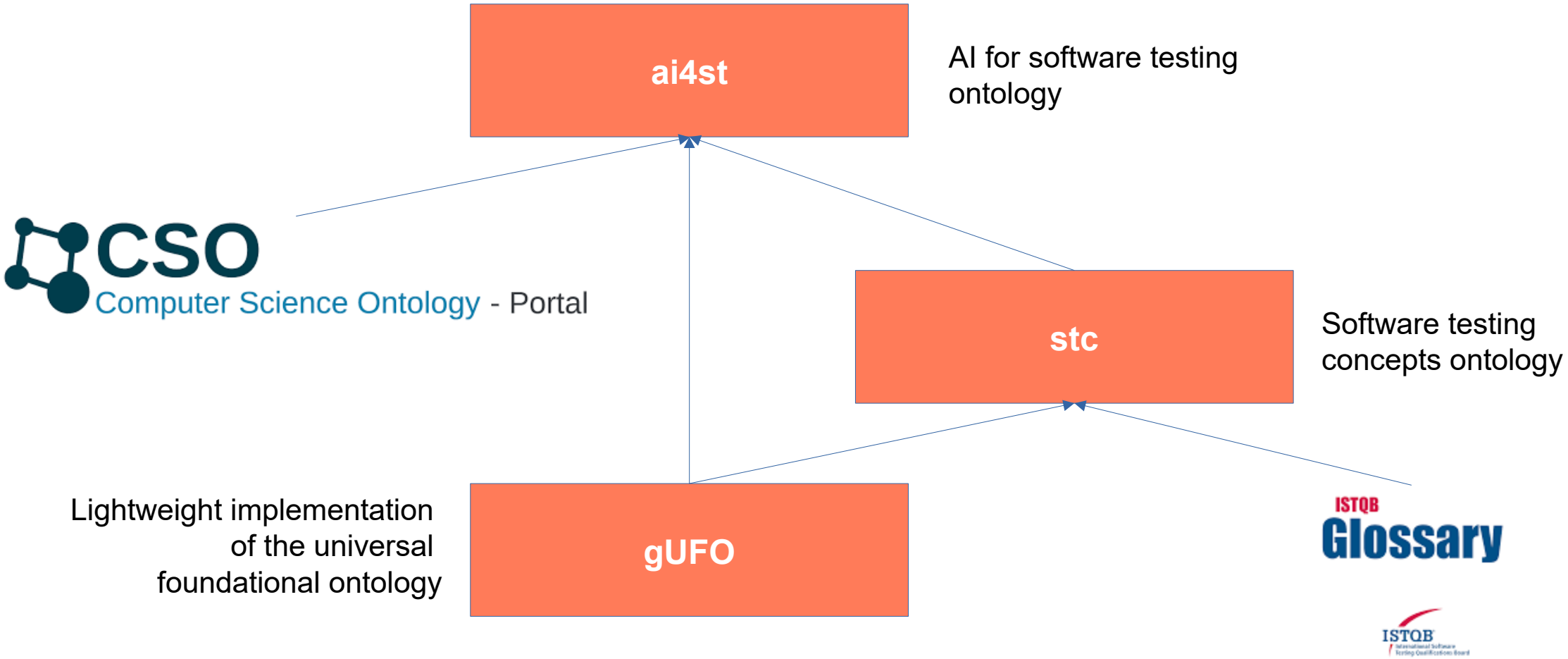
🧩 **So in practice:**

- AI assists in testing (self-checks, automation, generating test data).
- AI cannot replace human testers, because humans bring **context, empathy, critical thinking, and responsibility**.

ChatGPT 5: Human capabilities important for software testing – Aug 2025

Capability	Why It's Important in Software Testing	Importance
Attention to Detail	Detects subtle bugs and inconsistencies.	Must-have
Analytical Thinking	Breaks down complex systems, finds risk areas.	Must-have
Critical Thinking	Ensures requirements and outcomes make sense.	Must-have
Communication Skills	Needed for bug reports, test cases, and teamwork.	Must-have
Logical Reasoning	Supports systematic and structured testing.	Must-have
Persistence	Helps reproduce tricky bugs and verify fixes.	Must-have
Problem-Solving	Useful for diagnosing root causes.	Important
Domain Knowledge	Ensures testing covers real-world scenarios.	Important
Curiosity	Encourages exploration beyond obvious cases.	Important
Time Management	Keeps testing efficient under deadlines.	Important
Collaboration & Teamwork	Smooth cooperation with developers and stakeholders.	Important
Adaptability	Allows quick learning of new tools/tech.	Important
Creativity	Enables out-of-the-box test scenarios.	Nice-to-have
Empathy (User Perspective)	Helps predict user behavior and mistakes.	Nice-to-have
Decision-Making	Supports risk-based prioritization.	Nice-to-have

The *ai4st* ontology



Based on Schieferdecker, Ina K. "Augmenting software engineering with AI and developing it further towards AI-assisted model-driven software engineering" arXiv preprint arXiv:2409.18048 (2024).

Outline

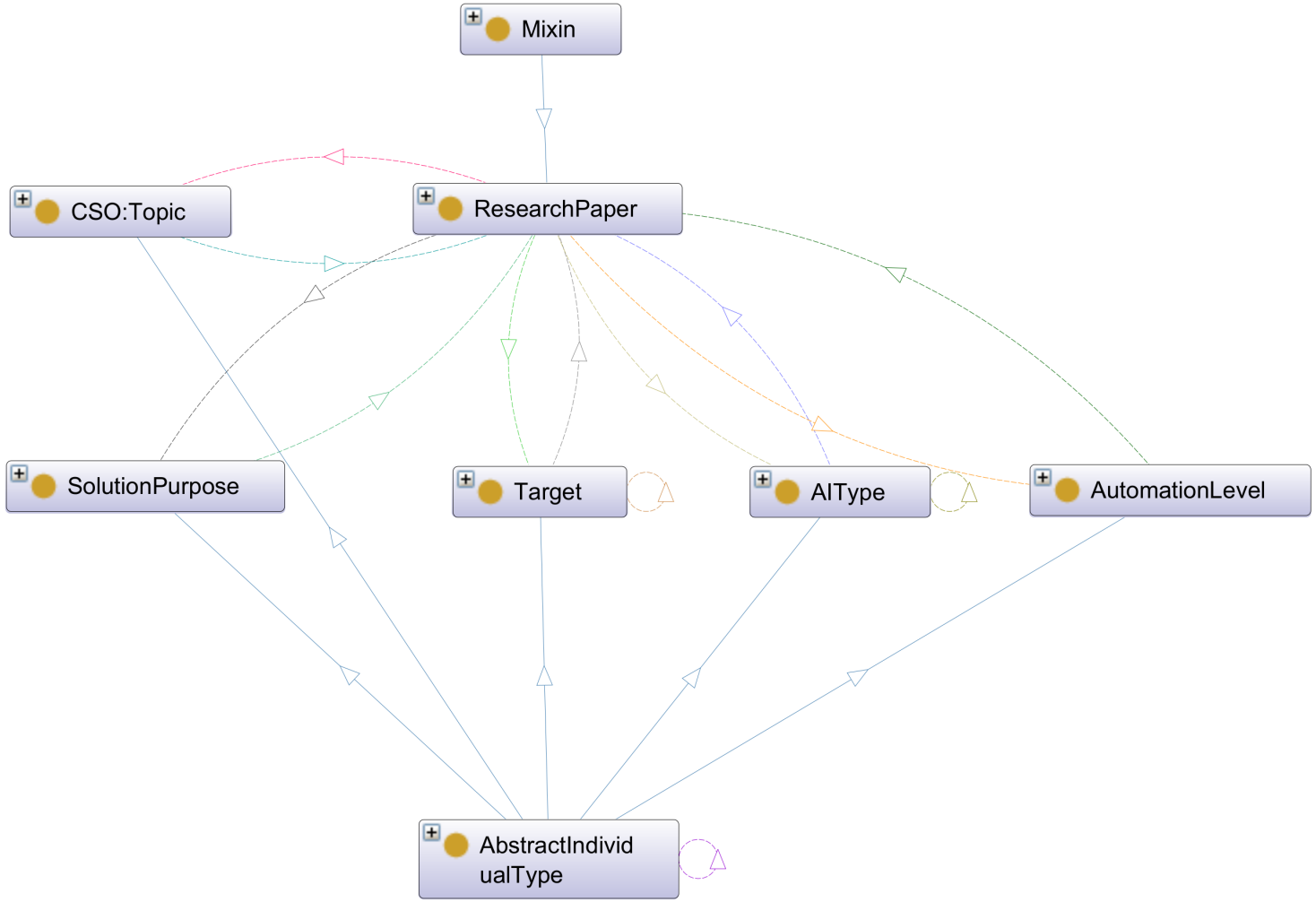
① Software engineering with AI

② Testing with AI

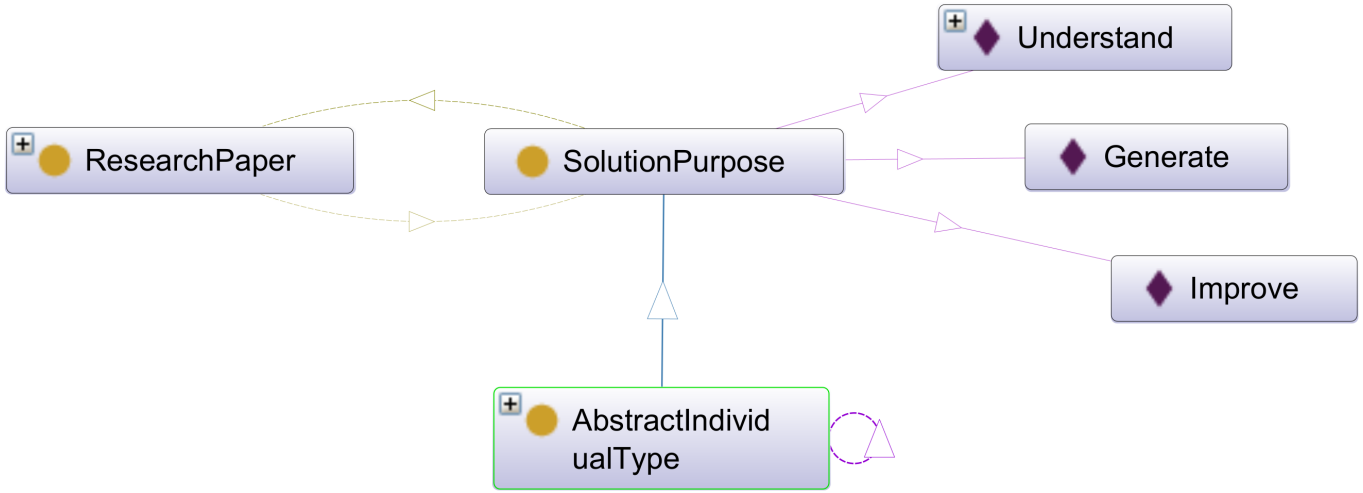
③ The taxonomy

④ Wrapup

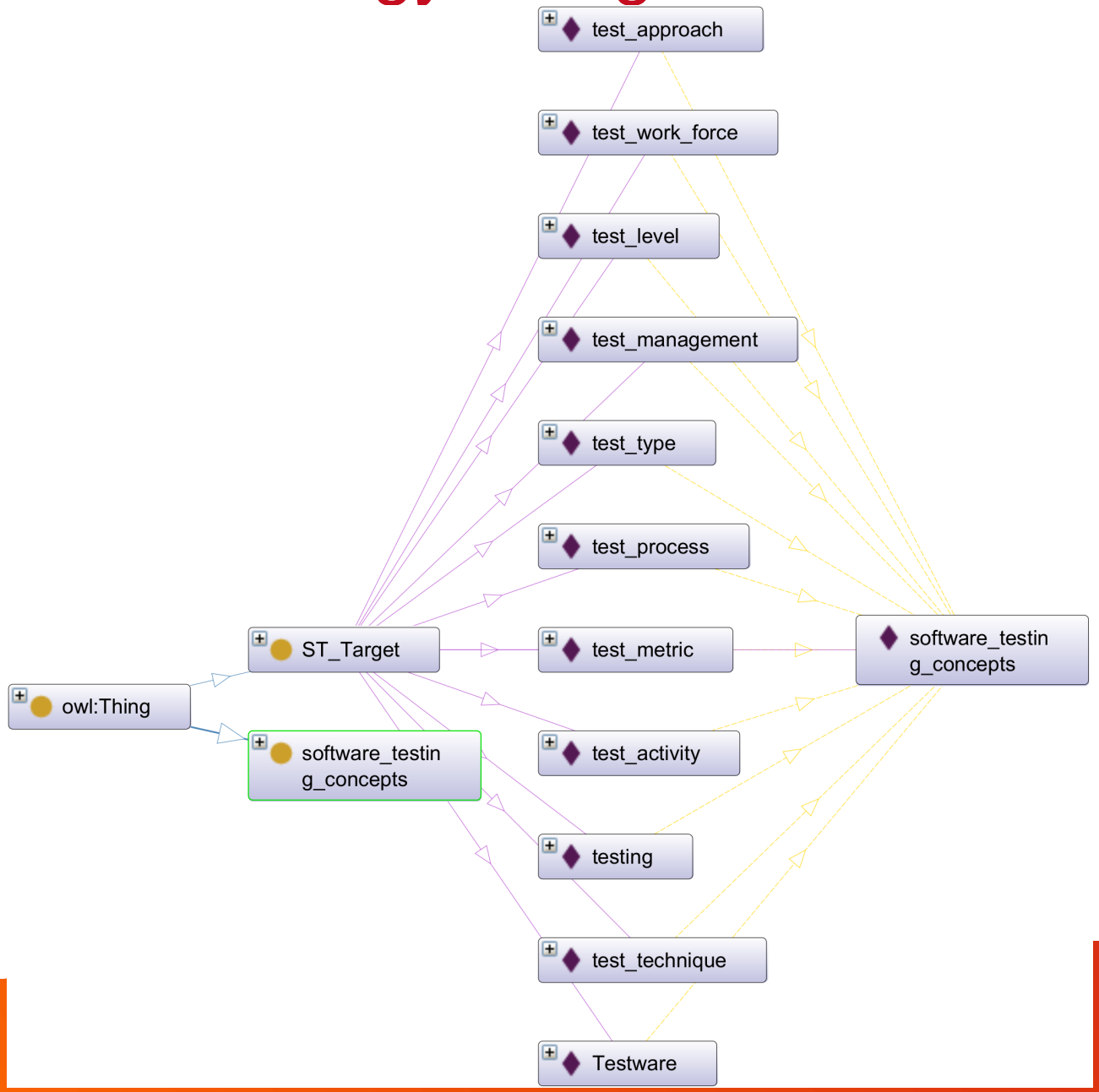
The *ai4st* ontology - overview



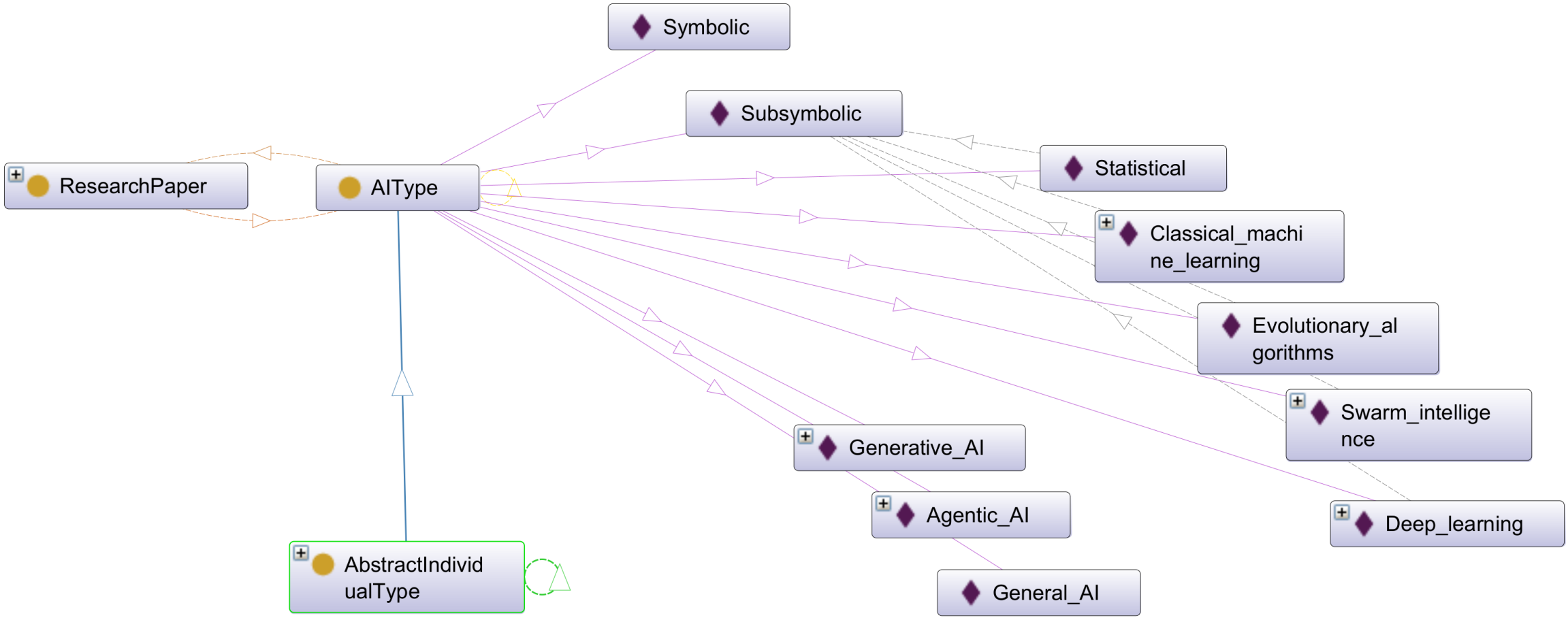
The *ai4st* ontology – purpose dimension



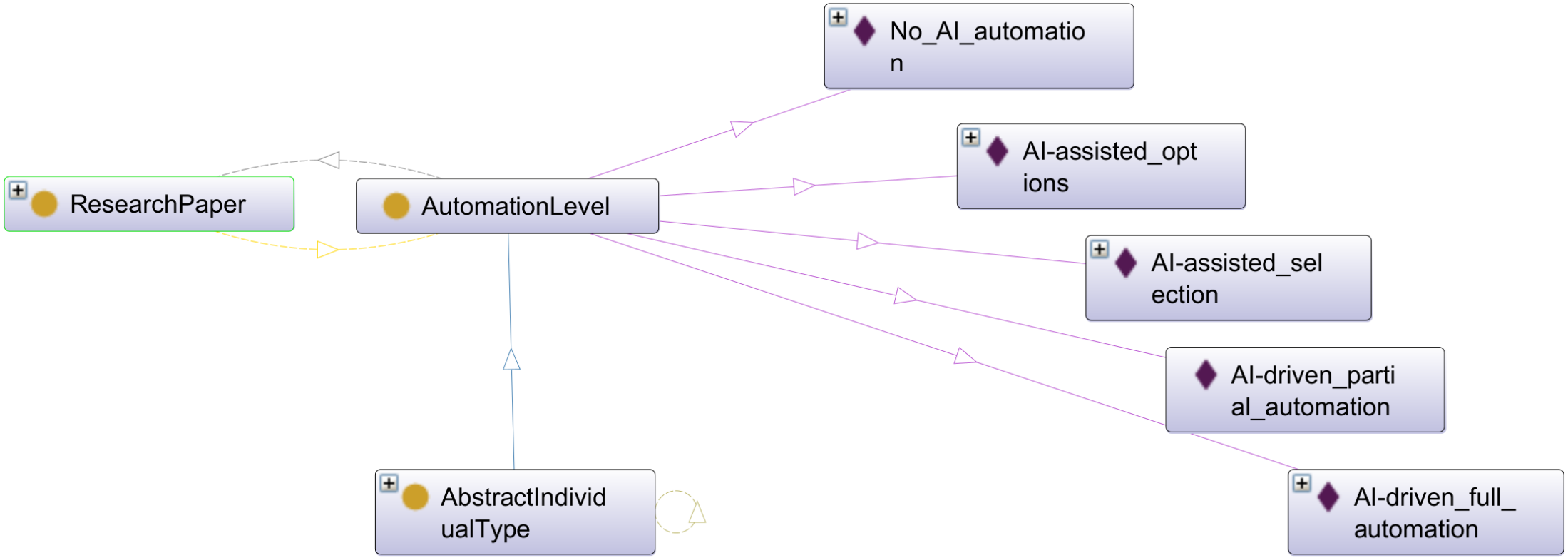
The *ai4st* ontology – target dimension



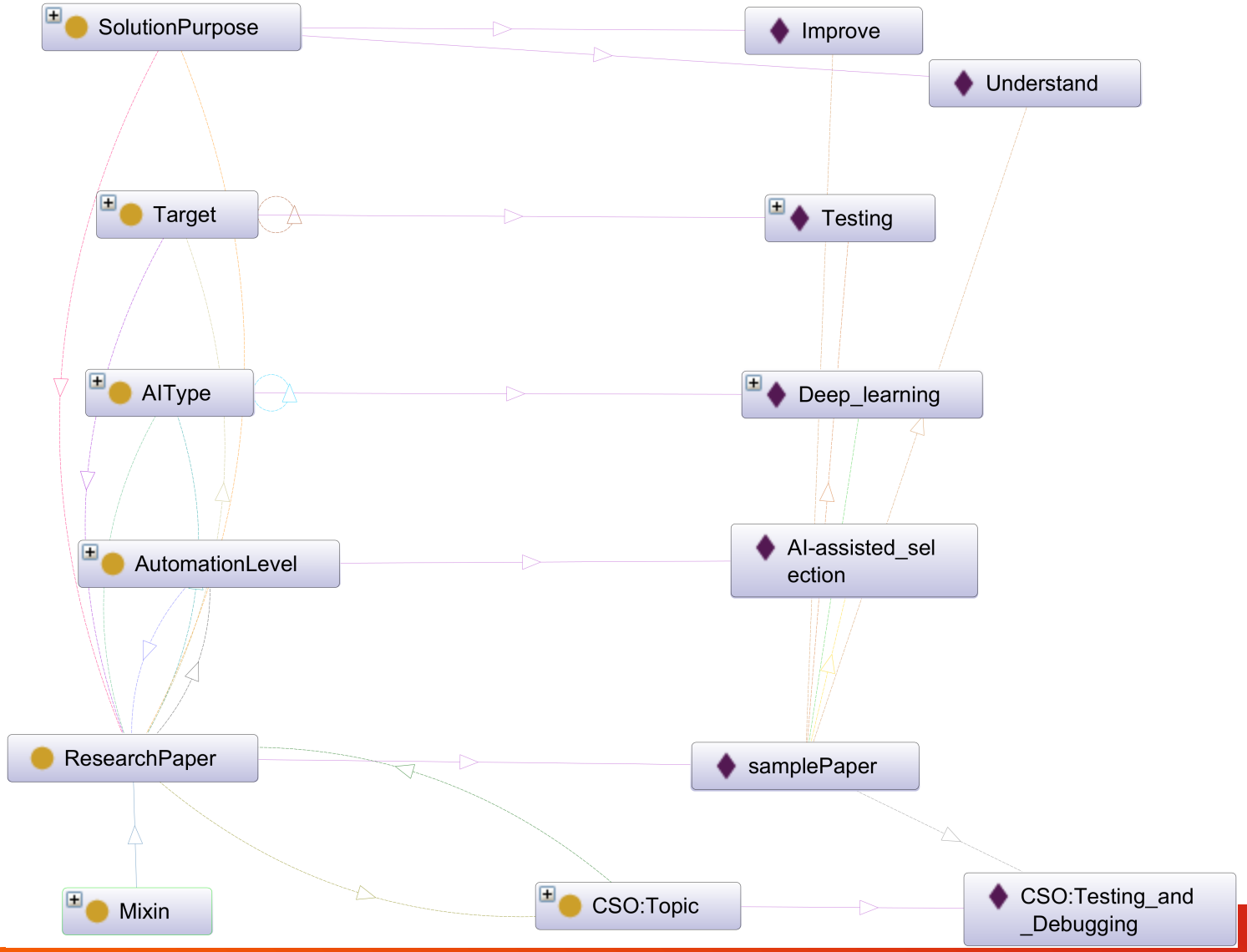
The *ai4st* ontology – AI type dimension



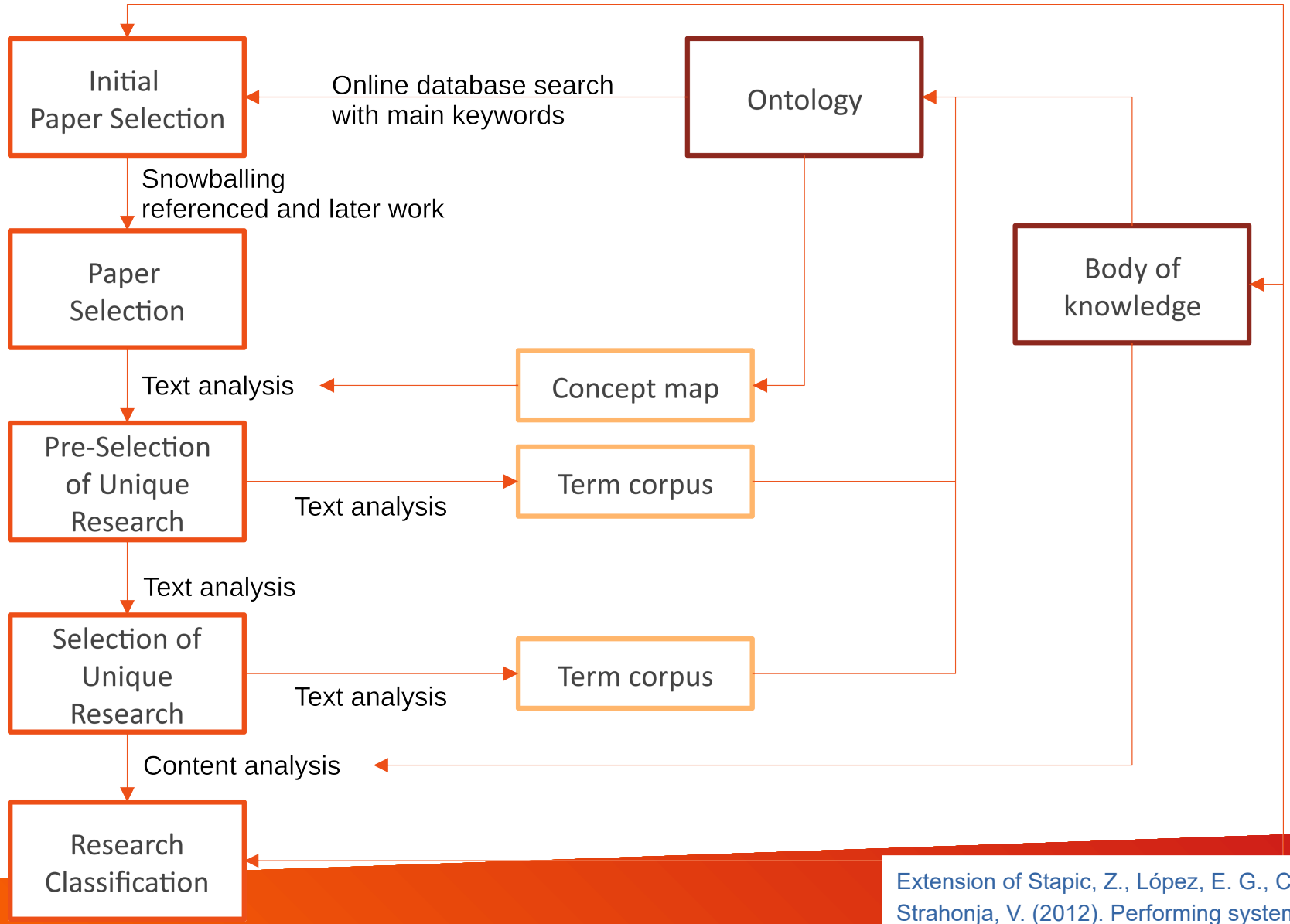
The *ai4st* ontology – automation level dimension



The *ai4st* ontology – sample paper



ai4st systematic literature review



Extension of Stapic, Z., López, E. G., Cabot, A. G., de Marcos Ortega, L., & Strahonja, V. (2012). Performing systematic literature review in software engineering. In Central European conference on information and intelligent systems (p. 441). Faculty of Organization and Informatics Varazdin.

SLR results

- Starting with ICSE 2025 + co-located conferences/workshop and backward snowballing, **1643** papers identified
- **1150** referred to a term in the stc ontology in their abstracts and/or titles, but **735** of these referred to only one term
- Another **1337** papers used variations of the terms in the stc ontology, such as 'unit test' instead of 'unit-level test', but **460** papers referred to only one variation.
- **949** papers using two or more original or alternative terms.
- Of these 949 papers, **656** contained terms from the *ai4st* ontology related to AI.
- Within the 656 papers, **38** relevant **original** research papers for *ai4st* were identified.
- They **covered** all purpose, all main target, all AI type and automation level facets except of AGI and full automation.

SLR results

Text analysis revealed

- 40 new *stc* term candidates:
  - terms such as 'test result' and 'fuzz testing' - included in the ISTQB Glossary, but not yet in *stc*
  - terms such as 'flaky test' and 'genetic testing' - not included in the ISTQB Glossary, but extensively discussed in research
    - Including terms such as 'test bot' and 'bias testing' related to AI for software testing or testing of AI
- 53 new *stc* synonym candidates and
- 26 terms that can be treated as either new term or synonym candidates.

(*stc*: over 200 terms, ISTQB glossary over 600 terms)

38 papers classified in *ai4st*

Ref	Title
[12]	A Method and Experiment to evaluate Deep Neural Networks as Test Oracles for Scientific Software
[40]	A new approach for automatic test case generation from use case diagram using LLMs and prompt engineering
[17]	Acceptance Test Generation with Large Language Models: An Industrial Case Study
[11]	AI in Service of Software Quality: How ChatGPT and Personas Are Transforming Exploratory Testing
[33]	AI-Assisted Test Script Generation for GUI Applications
[39]	AI-Based Enhancement of Test Models in an Industrial Model-Based Testing Tool
[36]	AI-Driven Acceptance Testing: first insights exploring the educational potential for test analysts
[29]	AI-Driven Testing: Unleashing Autonomous Systems for Superior Software Quality Using Generative AI
[37]	AI-driven web API testing
[23]	AI-Powered Multi-Agent Framework for Automated Unit Test Case Generation: Enhancing Software Quality through LLMs
[26]	AIs Understanding of Software Test Architecture
[43]	An Agent-based Architecture for AI-Enhanced Automated Testing for XR Systems
[34]	An Approach To Extract Optimal Test Cases Using AI
[21]	An Approach to GUI Test Scenario Generation Using Machine Learning
[44]	AsserT5: Test Assertion Generation Using a Fine-Tuned Code Language Model
[41]	Automation of Test Skeletons Within Test-Driven Development Projects
[69]	BugBlitz-AI: An Intelligent QA Assistant
[20]	ClozeMaster: Fuzzing Rust Compiler by Harnessing LLMs for Infilling Masked Real Programs
[66]	Deep Multiple Assertions Generation

38 papers classified in *ai4st*

...continued from previous page

[6]	Development of Cloud and Artificial Intelligence based Software Testing Platform (ChArIoT)
[22]	Generative AI for Software Test Modelling with a focus on ERP Software
[60]	Ethical AI-Powered Regression Test Selection
[28]	Getting pwn.d by AI: Penetration Testing with Large Language Models
[72]	GUI-Based Software Testing: An Automated Approach Using GPT-4 and Selenium WebDriver
[8]	Leveraging Large Language Models for Usability Testing: a Preliminary Study
[18]	LLM-Based Labelling of Recorded Automated GUI-Based Test Cases
[70]	New Approaches to Automated Software Testing Based on Artificial Intelligence
[42]	On the Effectiveness of LLMs for Manual Test Verifications
[19]	Owl Eye: An AI-Driven Visual Testing Tool
[35]	Reducing Workload in Using AI-based API REST Test Generation
[10]	Reinforcing Penetration Testing Using AI
[24]	RIVER 2.0: an open-source testing framework using AI techniques
[1]	TCP-Net++: Test Case Prioritization Using End-to-End Deep Neural Networks Deployment Analysis and Enhancements
[30]	TOGLL: Correct and Strong Test Oracle Generation with LLMs
[54]	Using an agent-based approach for robust automated testing of computer games
[12]	Using Large Language Models to Generate Concise and Understandable Test Case Summaries
[45]	Visual Test Framework: Enhancing Software Test Automation with Visual Artificial Intelligence and Behavioral Driven Development
[27]	WIP: Assessing the Effectiveness of ChatGPT in Preparatory Testing Activities

Test case design with AI so far

- Xiao *et al.* “Software testing by active learning for commercial games” to generate tests for commercial games
  - learner receives as input samples of input/output pairs extracted from the game engine
  - as output, it yields a model of the game’s expected behavior
- Mariani *et al.* “Automatic testing of GUI-based applications” [illegible] ally
  - learner receives as input an initial test suite, GUI a[illegible]
  - as output, it produces a behavioral model from[illegible]
- Hoi *et al.* “Guided GUI testing of android apps with[illegible] sequences of test inputs for Android apps
  - learner receives as input sequences of actions [illegible]
  - as output, it gives a model representing the GUI of the a[illegible]
- Aarts *et al.* “Improving active mealy machine learning for protocol conformance testing”
  - learner receives sequences of input/output pairs
  - as output, it yields a Mealy machine model representing the behavior of the SUT

Models from system executions are learned from which tests are being generated (MBT v1)

Uptake of AI-based methods in software testing tools – Nov 2024

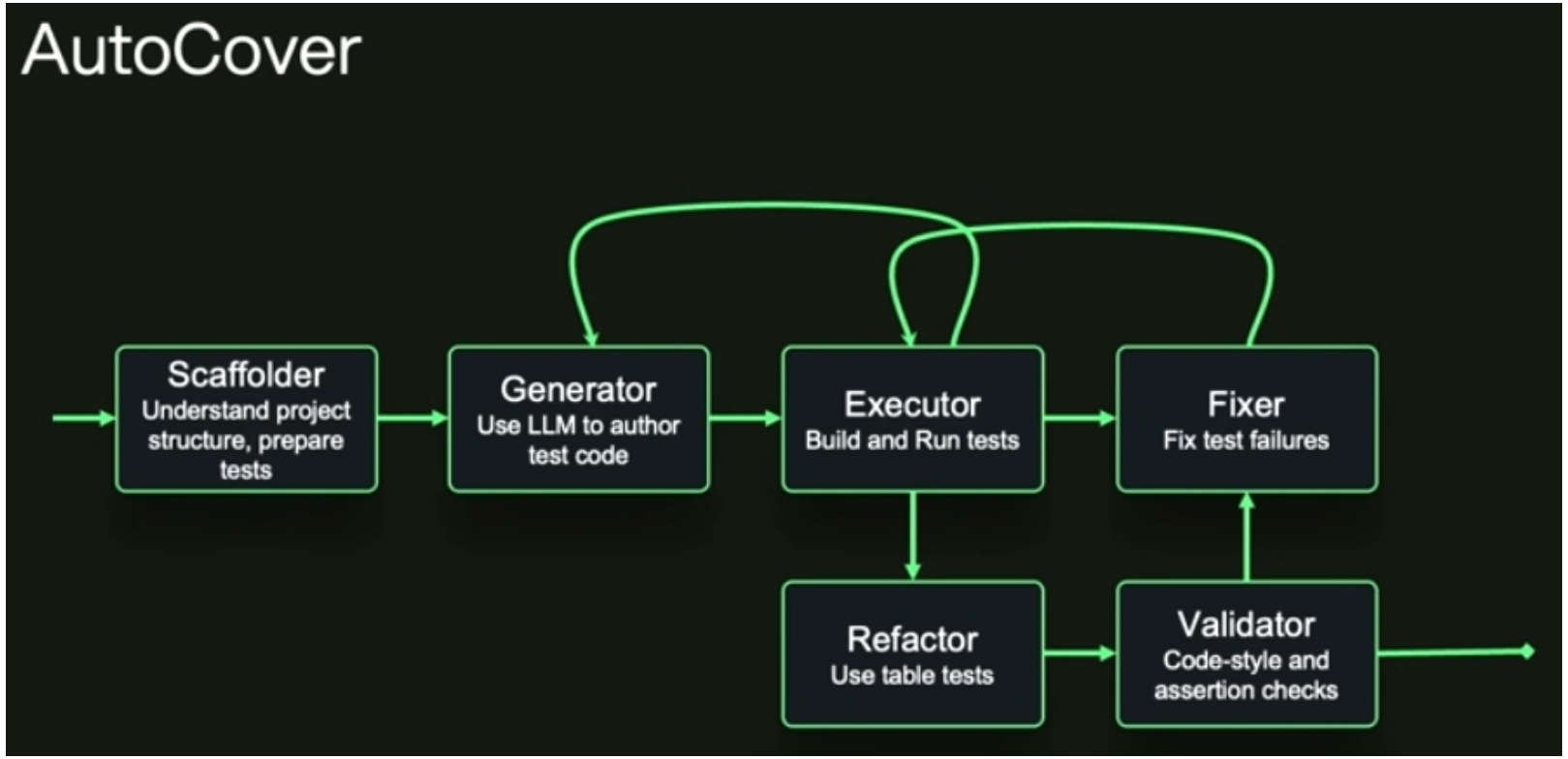
Tool	Key Features	Focus Areas
Testim.io	Machine learning for test authoring and maintenance	Automated UI tests
Applitools	Visual AI for automatic visual validation	Visual regression testing, cross-browser/device testing
Functionize	NLP for test creation, ML for maintenance	End-to-end tests, test analytics
Sauce Labs	AI and ML for visual testing and anomaly detection	Comprehensive testing suite, visual test
mabl	Machine learning for end-to-end testing automation	Visual regressions, JavaScript errors, link checks
ReTest	AI for difference testing	Regression testing without explicit assertions
Parasoft	AI and ML for noise reduction in static analysis	Static analysis, test optimization
Sealights	AI and analytics for test process insights	Test quality, risk assessment, resource optimization
TestCraft	AI for test maintenance and execution	Codeless Selenium test automation
Katalon Studio	AI for healing test objects and UI test automation enhancement	UI test automation

Uptake of AI-based methods in software testing tools – July 2025

Tool	Key Features	Focus Areas
Selenium	Equipped with AI to provide extensive flexibility for running and executing tests	Automating Web testing
Code Intelligence	Dynamic fuzz testing	AI-powered security testing for C/C++
Functionize	Machine learning-based tests use big data to understand website updates and self-heal workflows	End-to-end tests, automa
Testsigma	Automate tests in plain English	Low code test autom API
Katalon Studio	Testing at any scale	Fully fledged code, l platform
Applitools	Visual AI for automatic visual validation	Visual regression testing, cro g
Eggplant Digital Automation Intelligence	Model-based digital twin testing	Application and user interface testing
Digital.ai Continuous Testing	Continuous testing in DevOps environments	Web and mobile testing
Testcraft	Robust Selenium-based automated testing solution as a browser extension	Web testing
Testim	Machine learning for test authoring and maintenance	Web and mobile testing in agile environments

Now, also GUIs, APIs, databases and other backend services being supported

Perspective: Software testing with Agentic AI

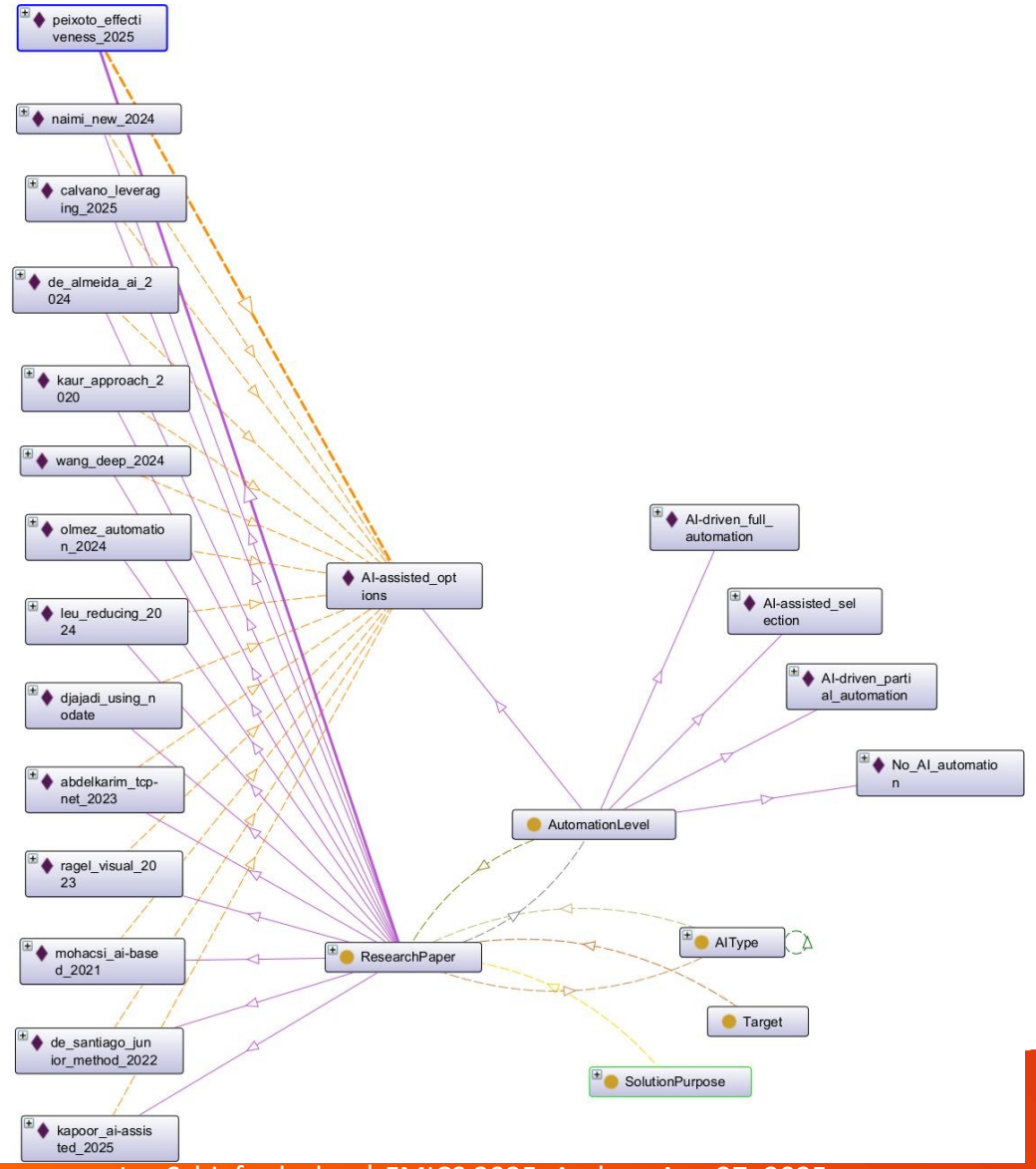


Uber solution with

- dedicated model
- internal tool
- 21.000 developer hours saved

Sourabh Shirhatti and Matas Rastenis: How Uber Built AI Agents That Save 21,000 Developer Hours, Jun. 2025
https://youtu.be/Bugs0dVcNI8?feature=shared

Perspective: Dynamic *ai4st* taxonomy



Combine taxonomy with

- agents for paper search
- agents for content analysis and classification
- agents for synchronizing with body of knowledge

Outline

① Software engineering with AI

② Testing with AI

③ The taxonomy

④ Wrapup

Summary: Software testing with AI

- Since long time
  - Mutation-based testing
  - Fuzz testing
  - Metamorphic testing
  - Evolutionary testing

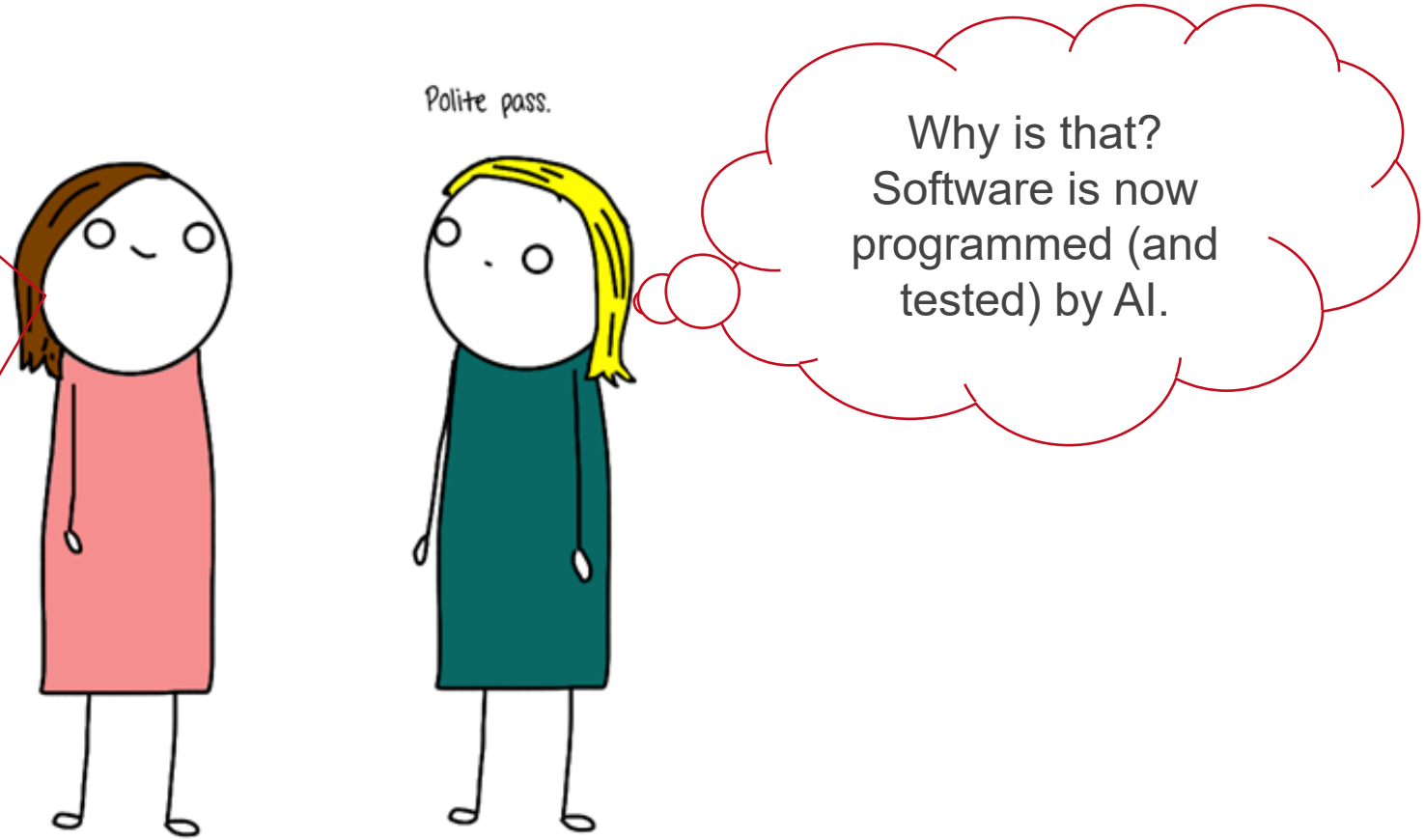
- Currently
  - NLP for test generation – „resembles“ scenario-based testing
  - Design and coding support – „resembles“ pair-programming
  - New testing methods – visual testing

- Prospectively
  - „AI“ for test prioritization, test effort estimation, test process optimization, MBT

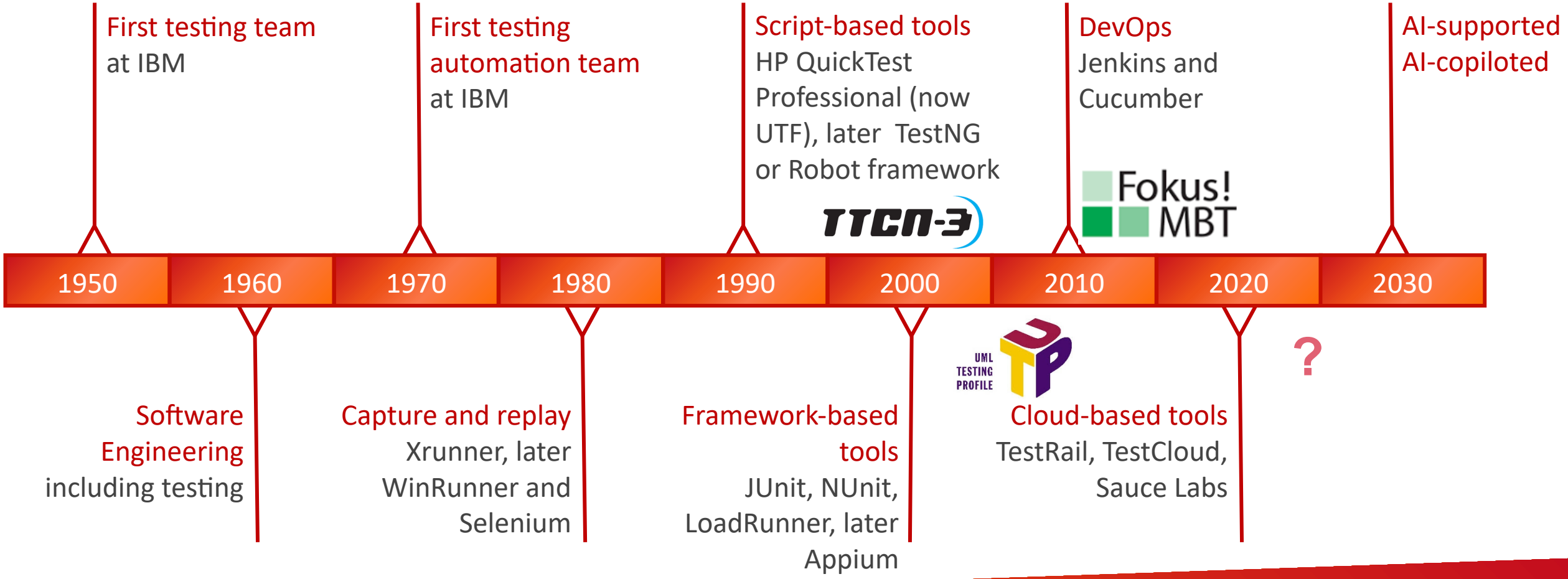
AI offers exciting new methods and tools to improve (automated) software testing and its research.

Still, AI-based methods and tools need to be improved significantly.

AI prompt engineering is a new field. AI prompts are potential test objects.



History of software test automation



Thank you for your attention.

If you want to follow up:

Navigating the growing field of research on AI for software testing
https://github.com/schieferdecker/ai4st

I used beside other the following tools :