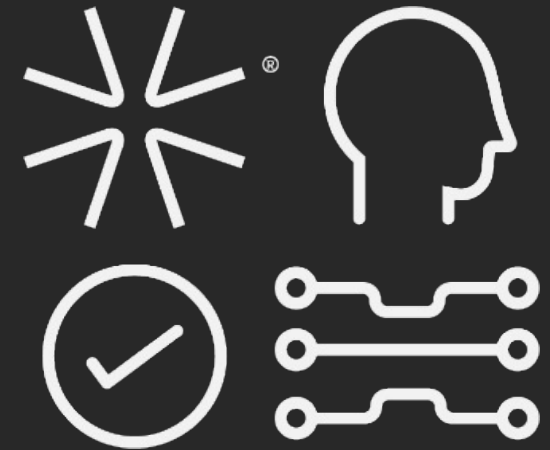


Challenges On Software Evolution

Trust as the new software currency



October 2025

Dr. Yiannis Kanellopoulos
Founder & CEO
yiannis@code4thought.eu

* code4thought ®

Overview



- 01 - About code4thought
- 02 - Our convictions about Software and AI
- 03 - On Software Evolution and The need for Testing
- 04 - On AI-Leveraged Software Engineering
- 05 - Closing Remarks

About code4thought

Origins

Founded in 2017 with a unique purpose, to render technology transparent, for large-scale software and AI-based systems.



Location

Delivery center in Athens
R&D center in Patras

Software Quality

We evaluate and monitor the **non-functional** quality of any software system at every stage of their life cycle, through the analysis of source code and architecture with the SIGRID platform of our partner SIG



Powered by

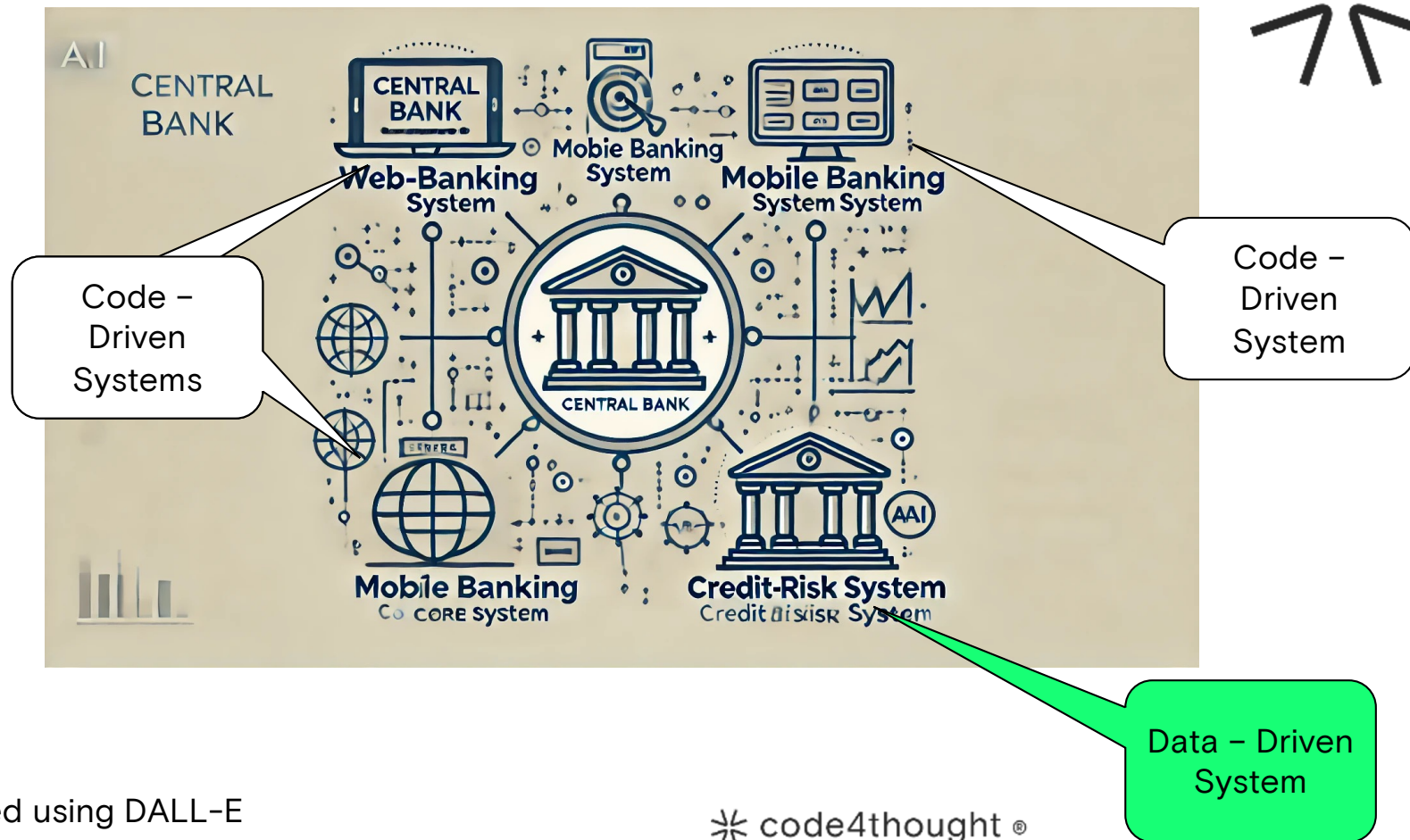
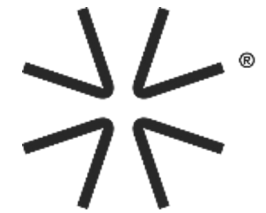
- _decisively investing in **own R&D**
- _international partnerships & project-participation
- _strong ties with Academia
- _ **TIER1 customer base**
- _team with academic & enterprise experience
- _participations and publications in international venues
- _cross domain & technology agnostic solutions
- _factual and well-researched approach

Trustworthy AI

We help organizations govern any type of AI-based system at every stage of their life cycle, by **testing** & auditing AI models & their datasets with iQ4AI, our own platform



As such: We are able to test any kind of software *



* Image generated using DALL-E

* code4thought ®

When it comes to AI: Our frame of mind

“So, I think what we should be doing, let's not even argue about what AI is, let's just talk about how do you govern the manufacturing of software.”

Joanna Bryson

“Data is socially constructed”

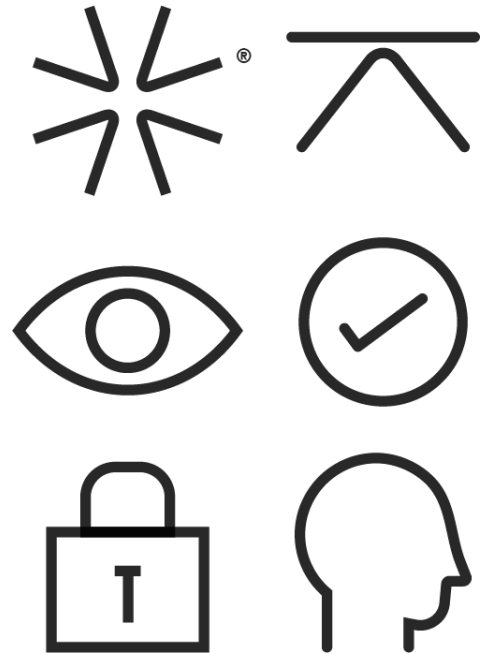
Meredith Broussard

“Organizations, who design systems, are constrained to produce designs which are copies of the communication structures of these organizations.”

Conway's Law

Our convictions about Software

- Good software performs its functions unnoticed*.
- 90% of a software engineer's time is spent in understanding the code they need to maintain/change,
- In a software system's lifecycle, 20% is spent for its initial development and 80% for its support/maintenance.

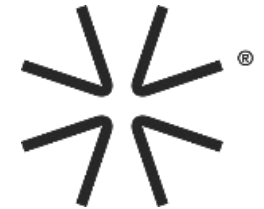


“How Does Your Software Measure Up? Mastering the DNA of our information society”

inaugural lecture by Prof. dr. ir. Joost Visser

* code4thought ®

On Software Evolution or Software as the invisible code of digital life



How it started



Andreessen Horowitz

<https://a16z.com> › why-software-is-eating-the-world ›

Why Software Is Eating the World

20 Aug 2011 — **Software is also eating much of the value chain of industries** that are widely viewed as primarily existing in the physical world. In today's ...

How it's going



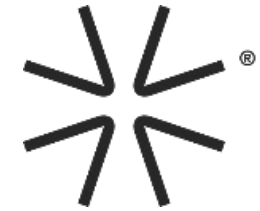
MIT Technology Review

<https://www.technologyreview.com> › 2017/05/12 › nvidi... ›

Nvidia CEO: Software Is Eating the World, but AI Is Going to ...

12 May 2017 — Nvidia CEO: Software Is Eating the World, but **AI Is Going to Eat Software**. Jensen Huang predicts that health care and autos are going to be ...

Different Types of Software

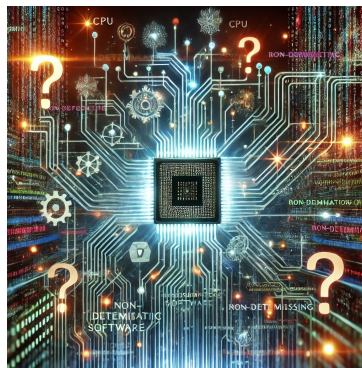


Deterministic – Code Driven

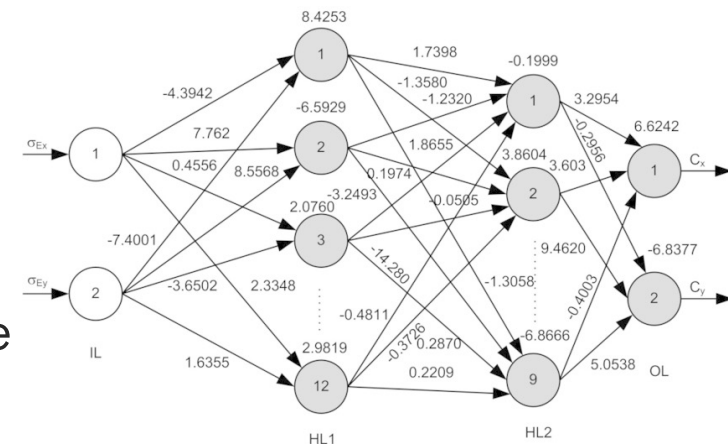
```
In [7]: 1 down = 0
        2 up = 100
        3 for i in range(1,10):
        4     guessed_age = int((up+down)/2)
        5     answer = input('Are you ' + str(guessed_age) + " years old?")
        6     if answer == 'correct':
        7         print("Nice")
        8         break
        9     elif answer == 'less':
        10        up = guessed_age
        11    elif answer == 'more':
        12        down = guessed_age
        13    else:
        14        print('wrong answer')
```

```
Are you 50 years old?less
Are you 25 years old?more
Are you 37 years old?less
Are you 31 years old?less
Are you 28 years old?more
Are you 29 years old?correct
Nice
```

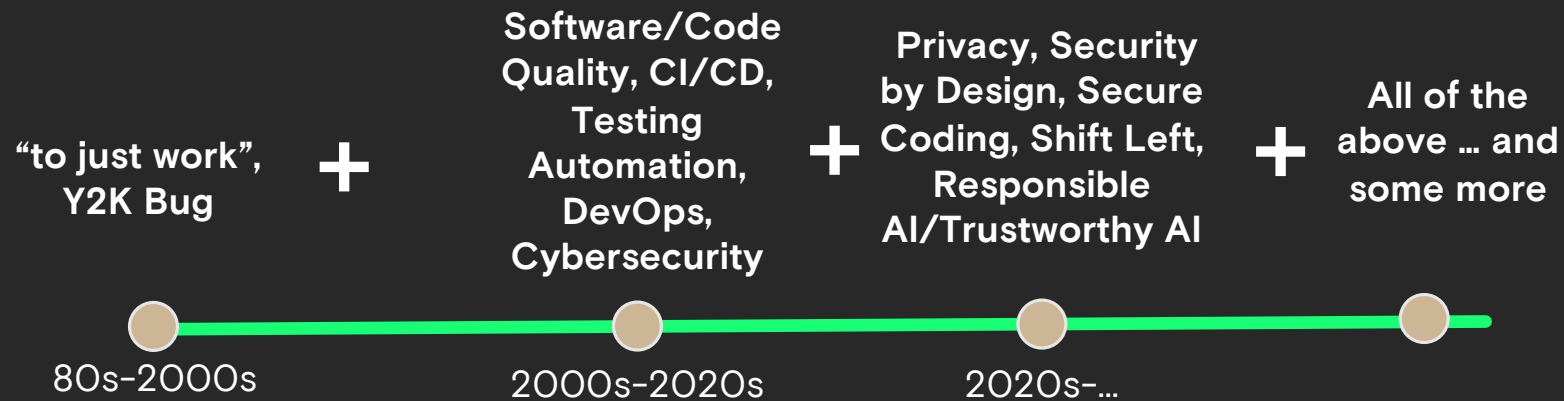
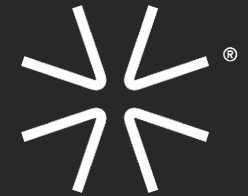
Non Deterministic – Generative



Probabilistic – Data Driven



The (Industry) evolution of Requirements



Testing an AI system: What is different* ?

**Source code
remains
important**

**(Way) More data
affecting critical
decisions**

**Non
deterministic
behavior**

* Compared to a typical software system?

✱ code4thought ®

The Toolkit

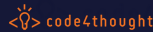
5 years ago

From Software Quality to AI Behavior		
	Code-Driven	Data-Driven
Existence of Industry Standards and Certifications	✓	✗
Formal Training and Professional Certifications	✓	-
Methodologies, Tooling, Processes	✓	-
Regulations, Legal Requirements	✓	-

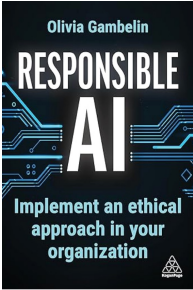
✗ Doesn't exist

✓ Fully exist

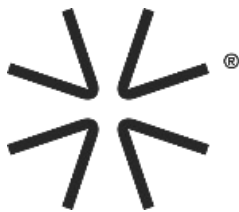
- Partially exist



From the “How F.A.T. (or F.Acc.T) is your ML Model? Quality in the era of Software 2.0” talk in 2020 Toronto Machine Learning Summit



Currently



ISO/IEC 42001:2023

Information technology — Artificial intelligence — Management system

Published (Edition 1, 2023)

ISO/IEC 5338:2023

Information technology — Artificial intelligence — AI system life cycle processes

Published (Edition 1, 2023)

ISO/IEC 25059:2023

Software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality model for AI systems

Published (Edition 1, 2023)

ISTQB® Certified Tester - AI Testing (CT-AI)



[JOIN/REACTIVATE](#)[ABOUT US](#)[CAREERS](#)[CREDENTIALING](#)[MEMBERSHIP](#)[ENTERPRISE](#)[PARTNERSHIPS](#)



Specialize and Master

Introduction to AI for Auditors

This course provides an introduction to how you can auditing practice. Learn AI auditing concepts, how processes and more.



[JOIN/REACTIVATE](#)[ABOUT US](#)[CAREERS](#)[SUPPORT](#)[STORE](#)[SIGN IN](#)[CREDENTIALING](#)[MEMBERSHIP](#)[ENTERPRISE](#)[PARTNERSHIPS](#)[TRAINING & EVENTS](#)[RESOURCE](#)



Specialize and Master

Auditing Generative AI: Methodology, Risk Mitigation, and Best Practices

This course introduces IT auditors to generative AI (GenAI), discusses the importance of AI strategy, explores use cases and addresses challenges associated with auditing GenAI.



Automated Employment Decision Tools: Frequently Asked Questions

The NYC Department of Consumer and Worker Protection (DCWP) enforces Local Law 144 of 2021 which regulates automated employment decision tools. These FAQ provide general information and guidance. Sections include:

- I. Overview of the Law
- II. General Bias Audit Requirements
- III. Data Requirements
- IV. Independent Auditors
- V. Responsibility for Bias Audits
- VI. Notice Requirements
- VII. Complaints

I. Overview of the Law

1. What is Local Law 144 of 2021?

The Law prohibits employers and employment agencies from using an automated employment decision tool (AEDT) in New York City unless they ensure a bias audit was done and provide required notices.

The Law was enacted in 2021. It took effect on January 1, 2023. Enforcement begins on July 5, 2023.

EU AI Act: first regulation on artificial intelligence

The use of artificial intelligence in the EU is regulated by the AI Act, the world's first comprehensive AI law. Find out how it protects you.



AI leveraged Software Engineering: How To

NLP

- Automate documentation
- Enhance Code Readability
- Improve Team Communication

LLM-driven Soft Programming

- Train programs with examples rather than instructions
- Learning from existing code to generate new solutions
- Speed-up the creation of unit-tests

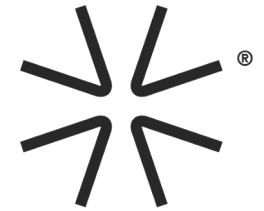
Hard Programming

- Traditional programming languages will keep developing
- Focus on code control, performance optimizations and security

What we experience

Software Developers employ Productivity Gains	Need to mitigate new types of Risks	Focus on Quality
• Increase efficiency and innovation • Facilitating complex problem solving • Dealing with tedious tasks • Creating better and up-to-date software documentation	• Antipatterns • Generate technical debt at scale • Trustworthiness/Ethical/IP & Privacy concerns	• Thoughtful design decisions • More time for reviewing, improving and testing code • Code Reviews as a means to follow the human-in-the loop principle.

Closing Remarks



The next decade will not be about choosing between traditional software and AI systems but about ensuring they evolve together responsibly. As leaders, we need to make trust, explainability, and governance non-negotiable parts of this journey.

From testing code to testing AI, progress comes when **academia's** curiosity meets **industry's** pragmatism — building systems that are not only powerful, but also responsible.



Thank you