



Next-Generation Software Development

Jeff Barr

(he/him)

jbarr@amazon.com

VP & Chief Evangelist, Amazon Web Services

📍 北京

QCon

全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

📍 上海

QCon

全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月

5月

6月

8月

10月

12月

📍 上海

AiCon

全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

📍 深圳

AiCon

全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOPs
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播



参会咨询



查看会议

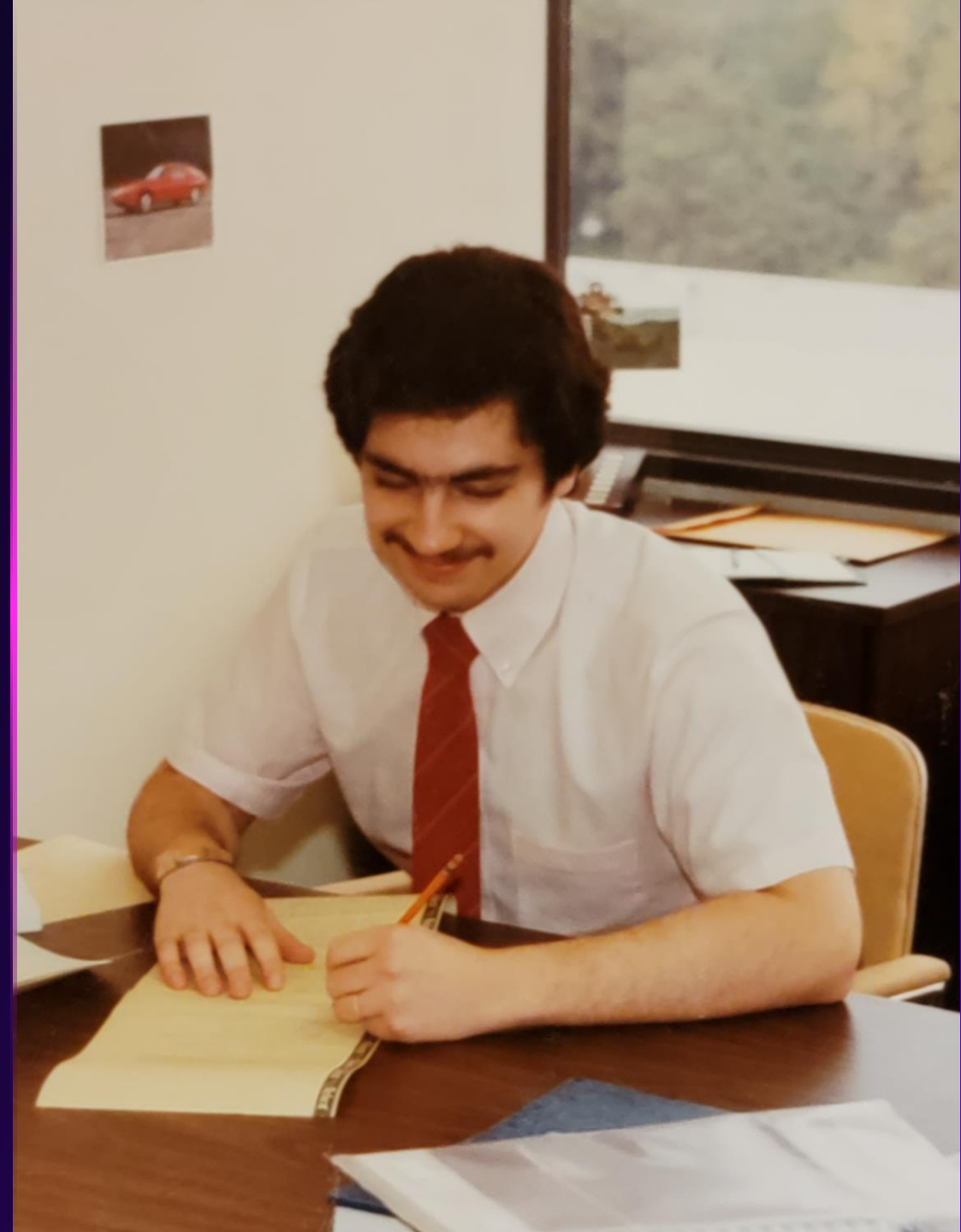
Hello, I am Jeff

Who I Am

- VP and Chief Evangelist, Amazon
- Husband, parent, grandparent
- Technologist
- Maker (3D printing, LEGO, electronics)
- BS - Computer Science (1985)
- Masters of Communication in Digital Media (2013)
- Professional developer since 1979

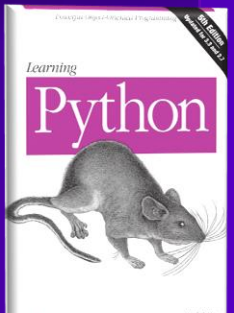
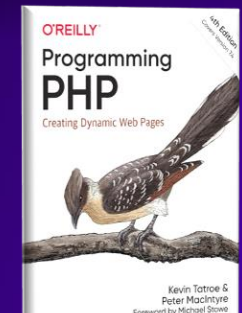
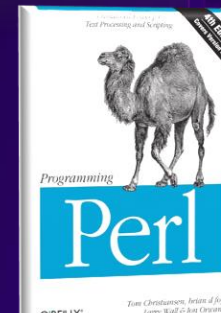
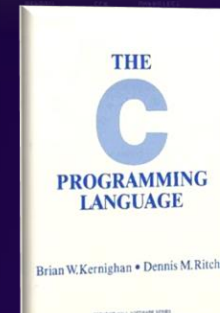
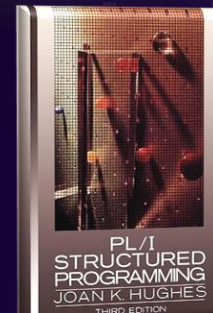
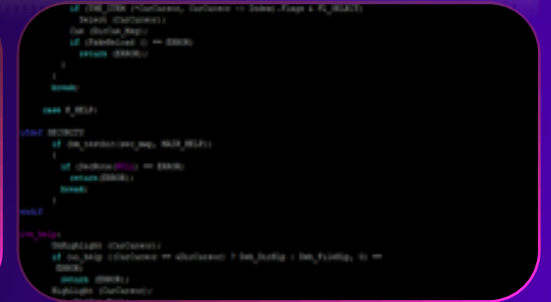
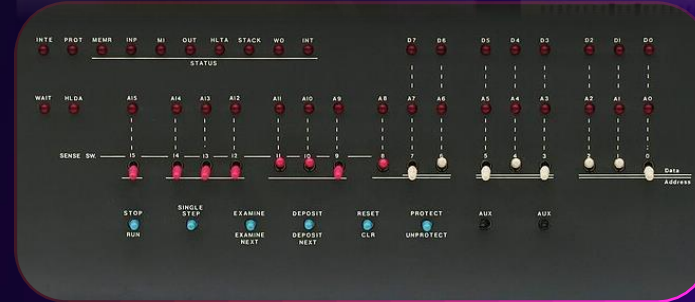
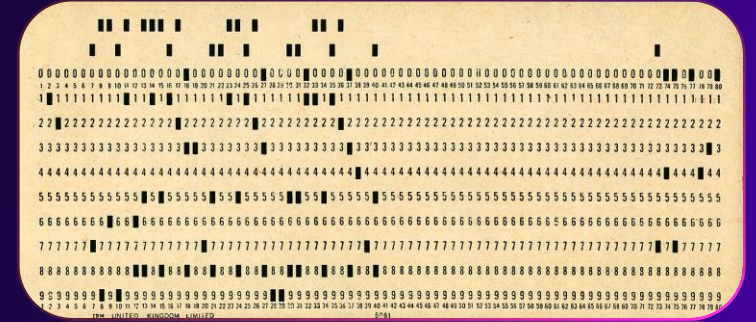
What I Do

- Amazon community events
- Amazon News Blog (2004-2024)
- Social media
- Amazon OnAir live stream
- Customer meetings



My development background

- Mainframe - IBM
- Minicomputer – PDP-8, PDP-11
- Microcomputer – 8080, 6502, 68000
- PL/I, BASIC, C, PHP, Perl, Python



Images produced by Amazon Bedrock & Amazon Nova Canvas



Japanese



Mexican



Peruvian



Armenian



Prague



Swiss



Adriatic



Romanian



“A detailed illustration, in a traditional **Chinese** style, of giants and regular sized humans using a variety of hand tools such as hammers and wrenches.”

Let's Dive In!



An exceptionally interesting point in time

- The practice of software development continues to evolve
- Many developers express a feeling of uncertainty
- How did we get here?
- Where are we going?
- Where do I fit in?
- How do we prepare?



A software developer surrounded by question marks in a modern Chinese cartoon style

Where are we today?

- Wide range of development tools, tactics, and approaches
- All are valid
- There's room for everyone:
- Informal / self-taught
- CS-educated / professional developer
- Math-educated / formal approach



a female software developer writing code using a hammer and a chisel to carve the code in stone. The scene looks like China in the 1600s. All in cartoon style

The future is here but not yet evenly distributed

Where are you today?

Hand-writing
instructions using
0's and 1's

Coding in
assembly
language

Coding in a
higher-level
language

Using an
AI-powered
coding assistant

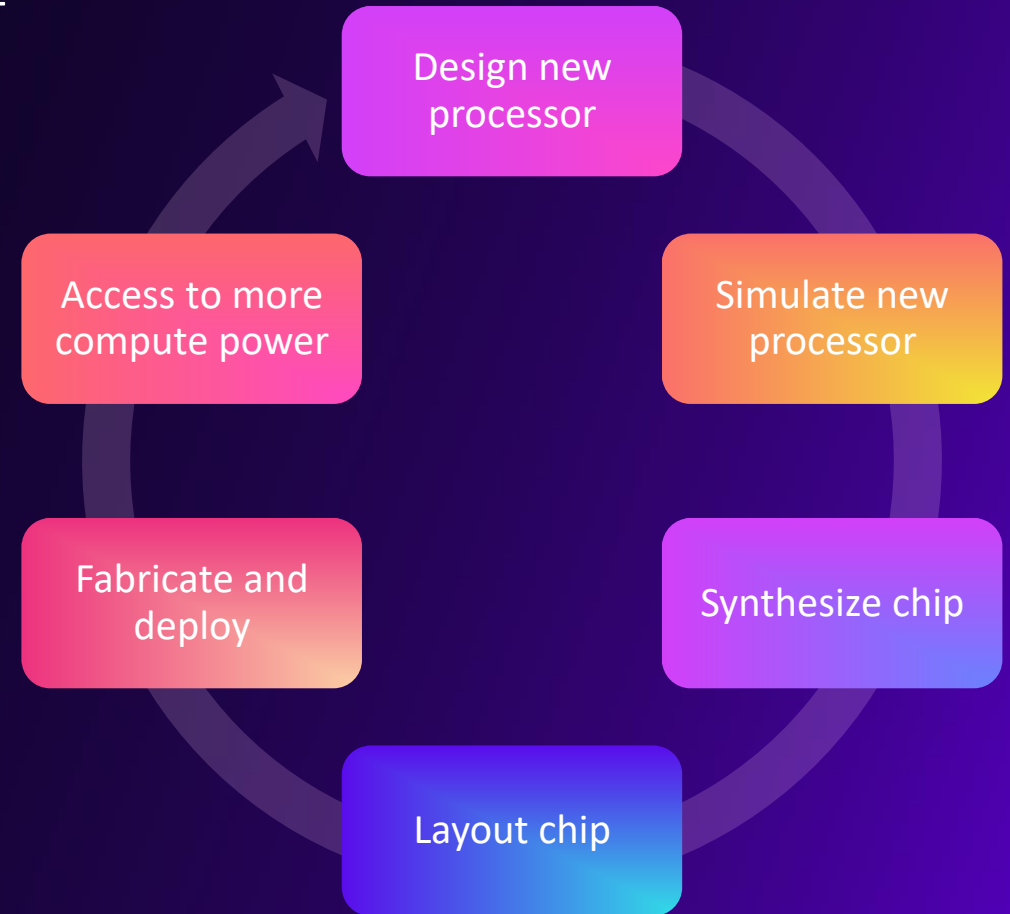
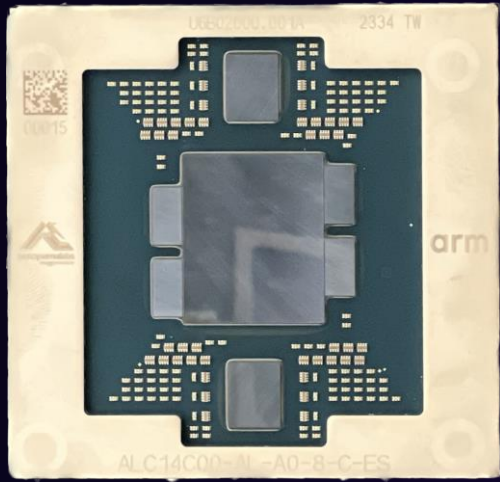
Using formal
methods to
validate your
code

Using a Quantum
Computer

happy software developers on the beach, looking out at a glorious rainbow made up
of source code fragments

The Amazon virtuous cycle in action for hardware

- Each generation of hardware accelerates production of the next one
- Time between generations shrinks
- Each generation more powerful & capable



Four generations of Amazon Graviton chips

Graviton

2018



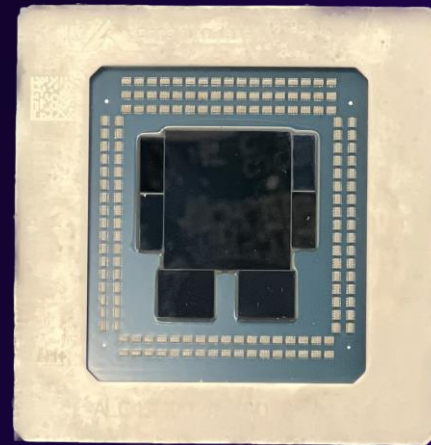
Graviton2

2019



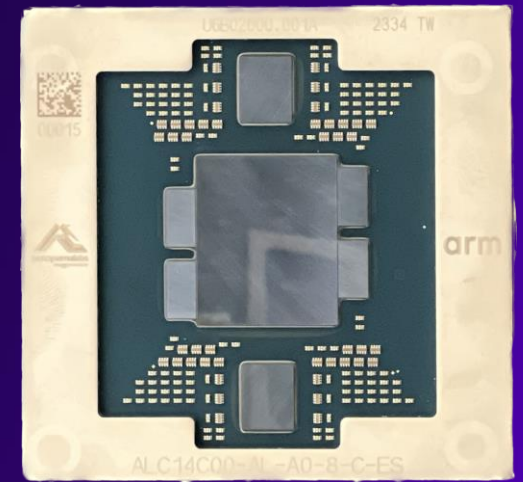
Graviton3

2021



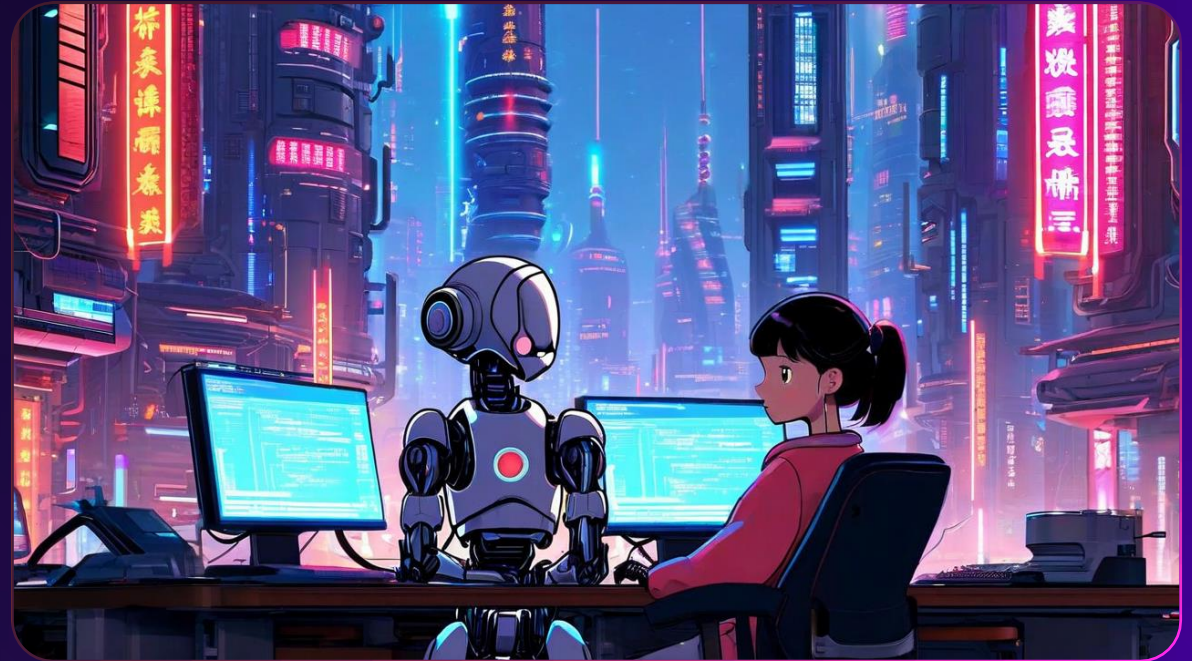
Graviton4

2023



A similar virtuous cycle for software

- Better computers support better and richer software
- We have been using software to build software since the beginning
- AI-powered coding assistants are just the next step
- Make you more powerful
- An assistant, not a replacement



A female Chinese software developer sitting next to a robot, with both of them at their own computers, all in an Chinese sci-fi landscape

Four I's of application development (I I I I)

Idea

Discover a problem that you want to solve



Intent

The desire to solve the problem in a particular way



Implementation

Solving the problem, going from idea to code

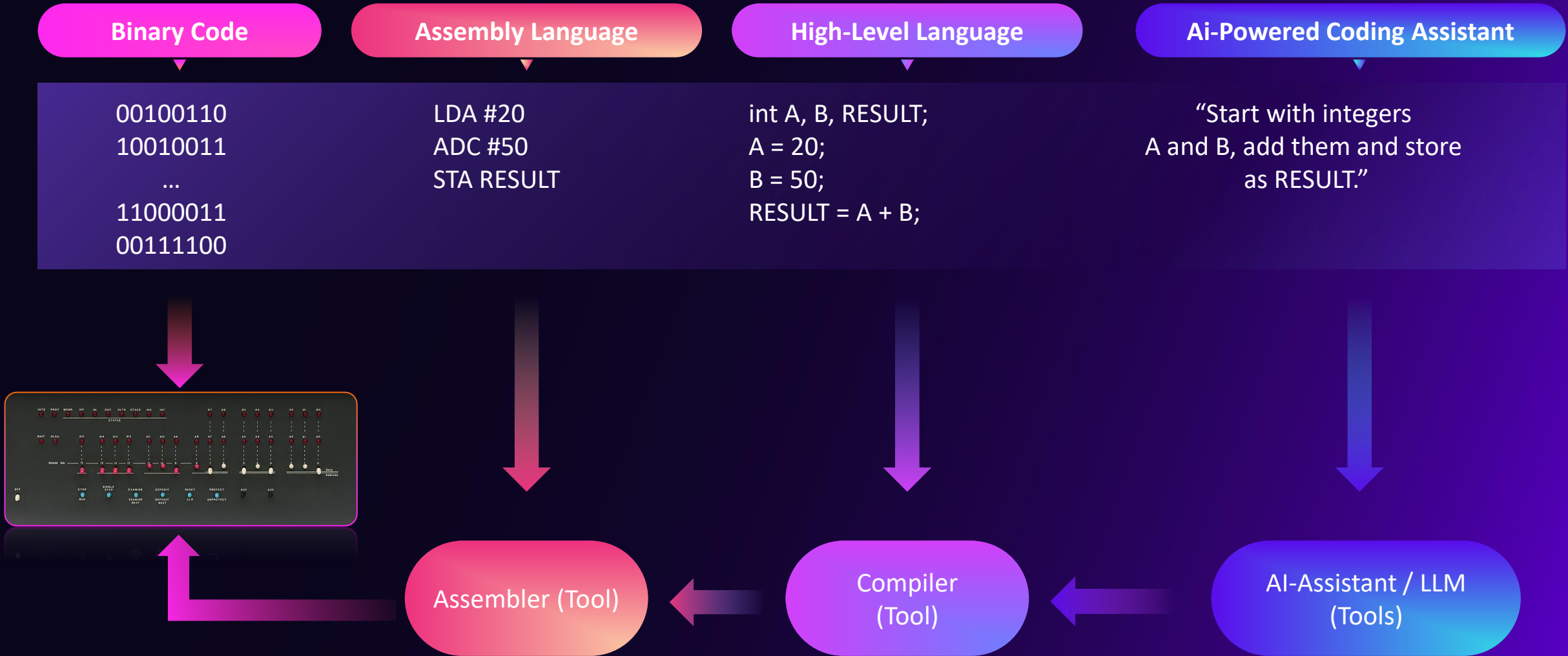


Iteration

Learning from experience, adding features



From intent to implementation



The Road to AI-Assisted Development

#define DeveloperTools "..."

- A developer tool is anything made from bits that helps you to do your job:
- Language feature
- Executable (editor, compiler)
- Function or class library
- Documentation
- Training
- ...



A futuristic Chinese-style library with books, racks of computers, keyboards, some hand tools, and more. There are three people sitting at a table in the middle

Why developer tools matter

- Tools let us “stand on the shoulder of giants” (Isaac Newton)
- Tools build on and compose with other tools
- Tools improve with (and take advantage of) available compute power, and grow more powerful themselves as a result



A detailed illustration, in a traditional Chinese style, of giants and regular sized humans using a variety of hand tools such as hammers and wrenches

Developer tools over time

- Assembler
- Editor
- Compiler
- Linter
- Debugger
- CASE tools
- Integrated Development Environment
- Source code control
- Profiler
- AI-Powered coding assistant



Language features



Compile time

- Structured programming
- Data types
- Objects
- Exceptions
- Memory safety



Run time

- Array bounds checking
- Exception handling
- Memory management / GC
- Overflow detection
- Assertion

Each one helps us to do a better job and to write better code

Packaging and reusable code

- Macro libraries
- Standard libraries
- Class libraries / JARS
- Open source



happy software developers on the beach, looking out at a glorious rainbow made up of source code fragments

And my point is....

- As humans, we imagine, build, and use tools
- Then we use them to build even more tools
- As developers we do the same
- Tools are good
- Respect (don't fear) progress



Mix it all together

#define DeveloperTools "..."

- A developer tool is anything made from bits that helps you to do your job:
- Language feature
- Executable (editor, compiler)
- Function or class library
- Documentation
- Training
- ...



A futuristic Chinese-style library with books, racks of computers, keyboards, some hand tools, and more. There are three people sitting at a table in the middle

Developer tools over time

- Assembler
- Editor
- Compiler
- Lint
- Debugger
- CASE tools
- Integrated Development Environment
- Source code control
- Profiler
- AI-Powered coding assistant



Language features

011001

Compile time

- Structured programming
- Data types
- Objects
- Exceptions
- Memory safety

Run time

- Array bounds checking
- Exception handling
- Memory management / GC
- Overflow detection
- Assertion

Each one helps us to do a better job and to write better code

Packaging and reusable code

- Macro libraries
- Standard libraries
- Class libraries / JARS
- Open source



happy software developers on the beach, looking out at a glorious rainbow made up of source code fragments

Model Training and Fine Tuning

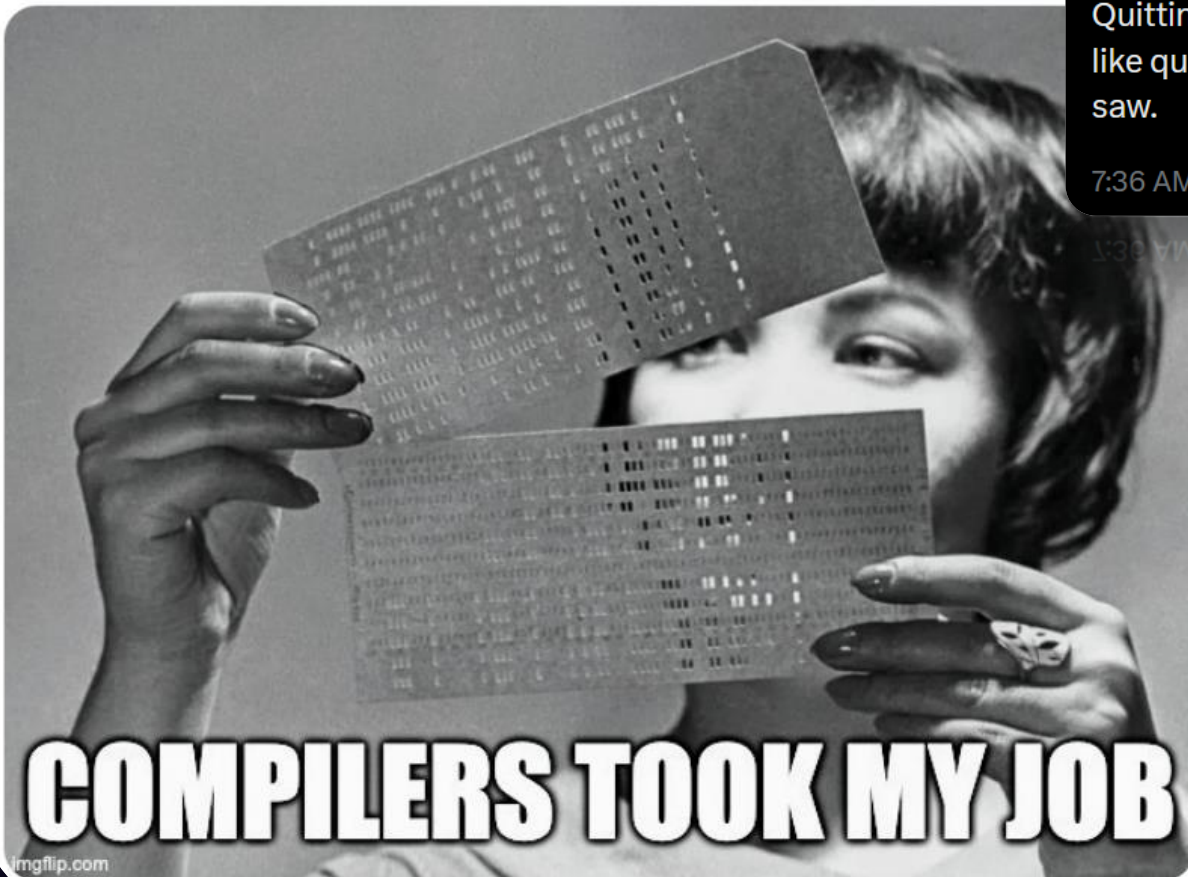
LLM with deep understanding of development processes, tools, documentation, APIs, coding styles, and code

AI-Powered Coding Assistant



Ryan Els 
@RyanEls4

Software developers in the 60s 🤔



Simon Willison 
@simonw

Quitting programming as a career right now because of LLMs would be like quitting carpentry as a career thanks to the invention of the table saw.

7:36 AM · Jul 3, 2025 · **707.4K** Views



What's Next / What's New / What's Now?



a highway that leads to a space-age city. There are several styles of signs that say "New" along the highway, and some floating in the air. Make it look really, really amazing!

Two emerging technologies

AI-Powered Coding Assistant

- Logical next step
- Make you a better developer
- From intent to code
- Increase efficiency
- Another tool for your toolbox
- Builds on what you know

Formal Verification

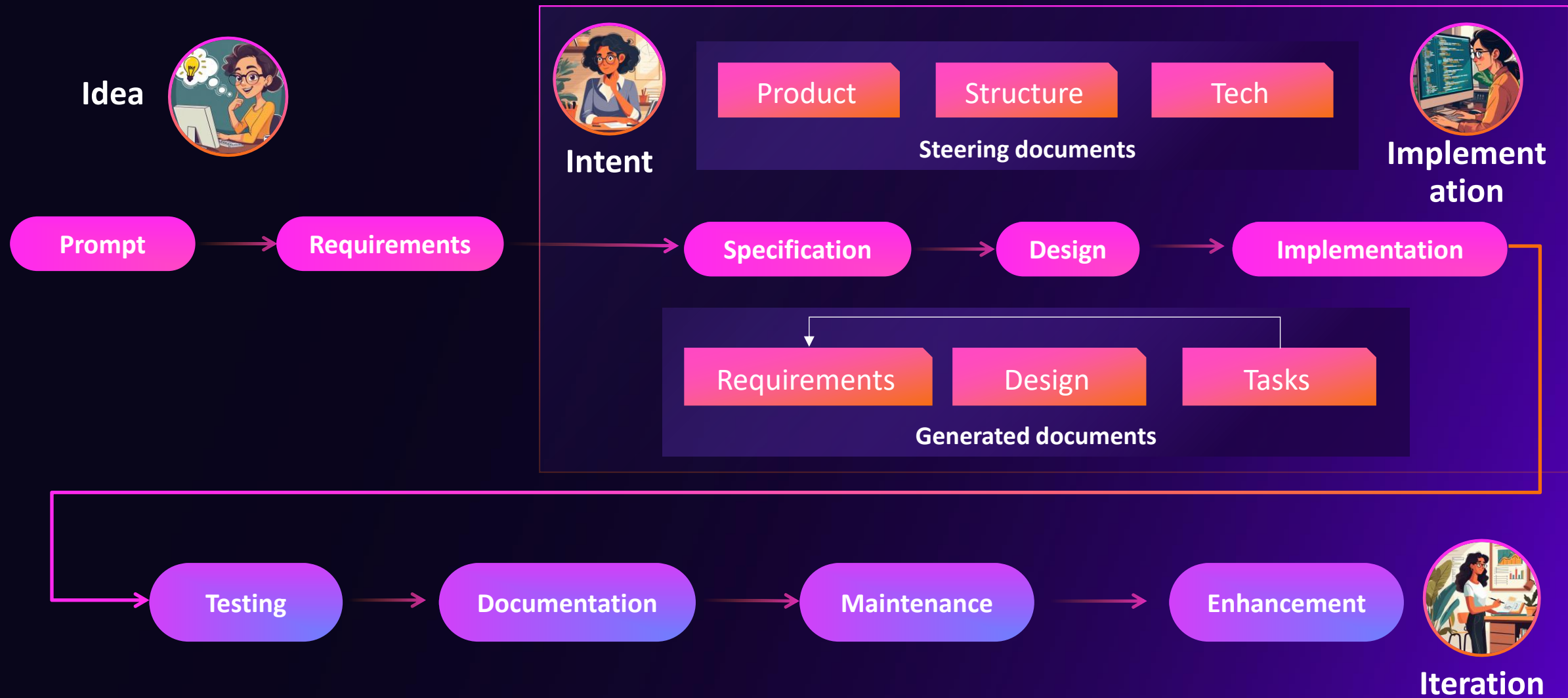
- Mathematical model
- Spec vs implementation
- Prove programs correct

Kiro – AI IDE from Amazon

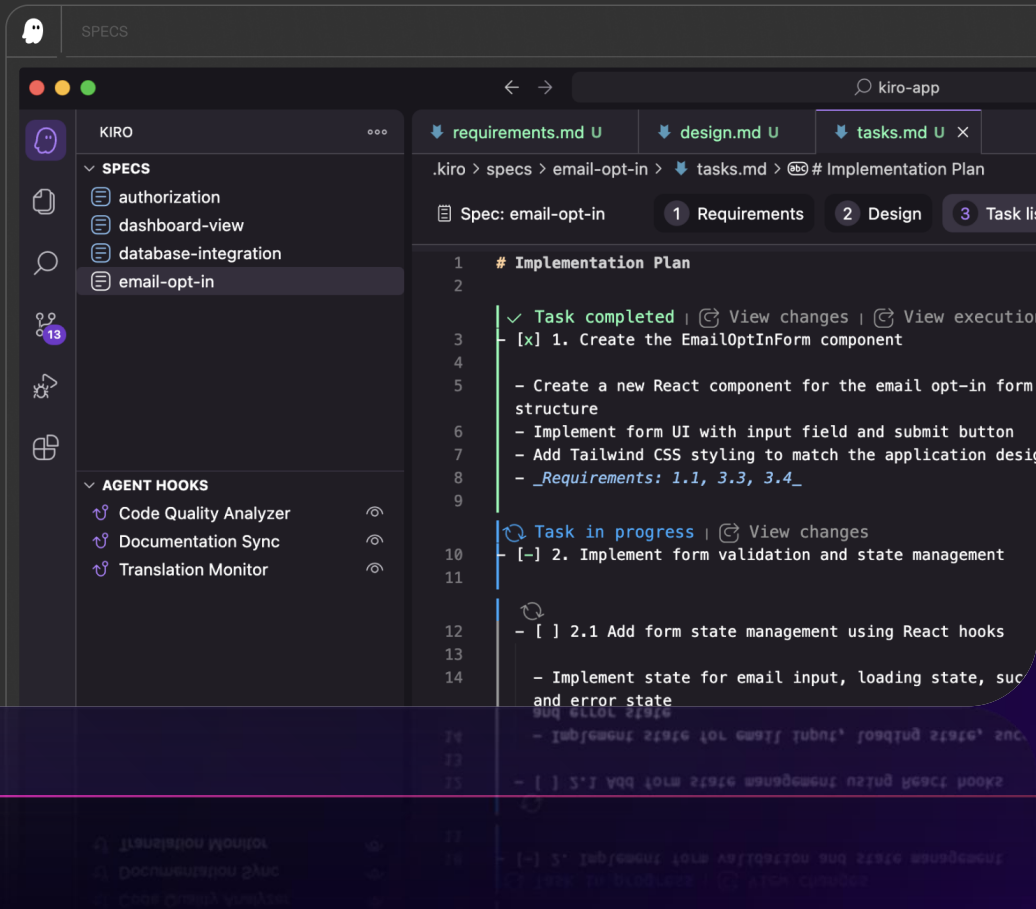
- From prototype to production
- Beyond vibe coding to spec-driven development
- Agents implement spec
- You remain in control
- Agent hooks delegate tasks to background AI agents
generate docs, unit tests, optimize code
- Get started for free



Kiro – Spec-driven development



Kiro in action



- Enter prompt via chat
- Co-create requirements
- Co-create spec
- Generate and review design
- Implement spec
- Test
- Refine
- Deploy
- Celebrate!

Kiro – requirements.md

specs > linkedin-posts-stats > requirements.md > # Requirements Document

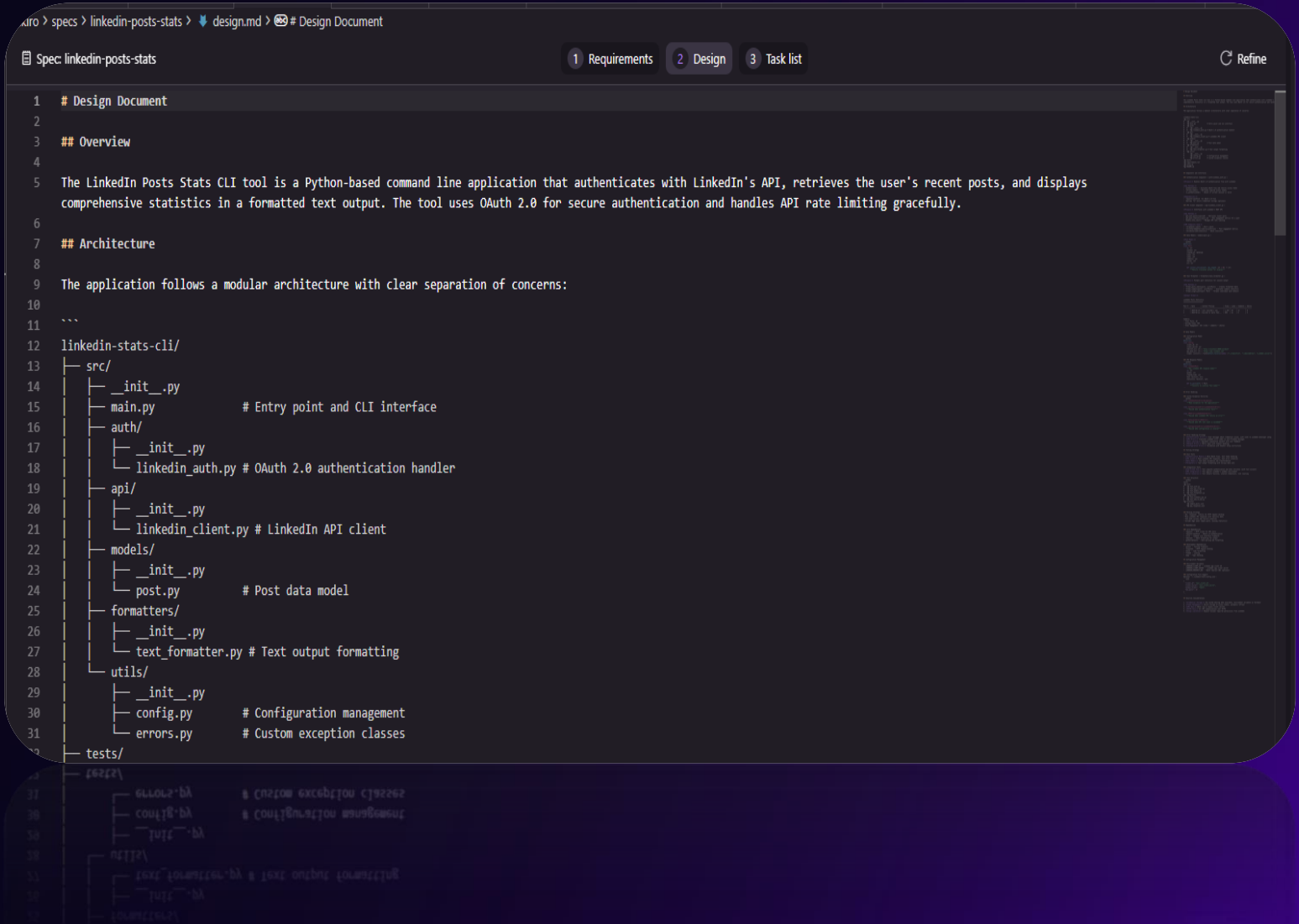
Spec linkedin-posts-stats

1 Requirements2 Design3 Task list

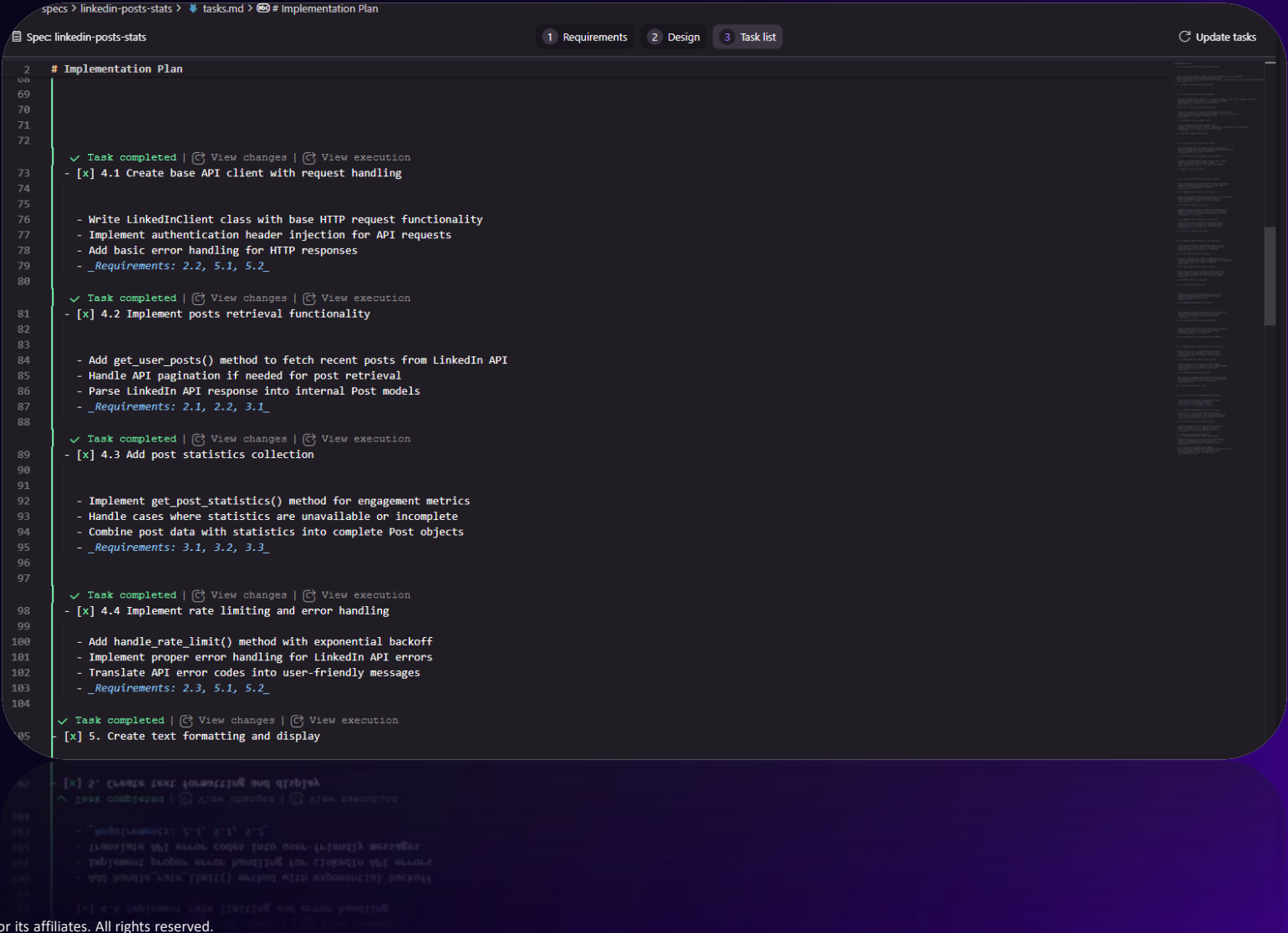
Refine

```
1  # Requirements Document
2
3  ## Introduction
4
5  This feature involves creating a command line tool that authenticates with LinkedIn, retrieves statistics for the user's 10 most recent posts, and displays them in a
  well-formatted text output. The tool will provide insights into post performance including engagement metrics like views, likes, comments, and shares.
6
7  ## Requirements
8
9  ### Requirement 1
10
11  **User Story:** As a LinkedIn user, I want to authenticate with my LinkedIn credentials, so that I can access my post statistics securely.
12
13  #### Acceptance Criteria
14
15  1. WHEN the user runs the tool THEN the system SHALL prompt for LinkedIn credentials
16  2. WHEN valid credentials are provided THEN the system SHALL authenticate successfully with LinkedIn API
17  3. IF authentication fails THEN the system SHALL display a clear error message and exit gracefully
18  4. WHEN authentication succeeds THEN the system SHALL store the session for the current execution
19
20  ### Requirement 2
21
22  **User Story:** As a LinkedIn user, I want to retrieve my 10 most recent posts, so that I can analyze my recent content performance.
23
24  #### Acceptance Criteria
25
26  1. WHEN authentication is successful THEN the system SHALL fetch the 10 most recent posts from the user's LinkedIn profile
27  2. IF the user has fewer than 10 posts THEN the system SHALL retrieve all available posts
28  3. WHEN API rate limits are encountered THEN the system SHALL handle them gracefully with appropriate delays
29  4. IF the API request fails THEN the system SHALL display an informative error message
30
31  ### Requirement 3
32
33  **User Story:** As a LinkedIn user, I want to see comprehensive statistics for each post, so that I can understand my content performance.
34
35  #### Acceptance Criteria
36
37  1. WHEN posts are retrieved THEN the system SHALL collect statistics including views, likes, comments, and shares for each post
38  2. WHEN statistics are available THEN the system SHALL include post creation date and content preview
39  3. IF certain statistics are unavailable THEN the system SHALL indicate this clearly in the output
40  4. WHEN collecting statistics THEN the system SHALL handle API pagination if required
41
42  ### Requirement 4
43
44  **User Story:** As a LinkedIn user, I want to see the statistics displayed in a clear, readable format, so that I can quickly understand my post performance.
```

Kiro – design.md



Kiro – tasks.md



Kiro – Executing a Task

Execute Task: 8. Create comprehensive test suite 8. Create comprehensive test suite

Kiro

Including Steering Documents product.md structure.md tech.md

I'll implement task 8 "Create comprehensive test suite" by starting with the first subtask. Let me first read the requirements, design, and current implementation to understand what needs to be tested.

Task: 8. Create comprehensive test suite from tasks.md

Status: In Progress

Task: 8.1 Write unit tests for data models and utilities from tasks.md

Status: In Progress

Let me examine the current implementation to understand what needs to be tested:

Read file(s) post.py config.py errors.py

Now let me check if there are any existing test files and the project structure:

Searched workspace

Checking existing test structure

Read file(s) requirements.txt

Now I'll create comprehensive unit tests for the data models and utilities. First, I'll add the necessary test dependencies and then create the test files.

Accepted edits to requirements.txt

Now I'll create the unit tests for the Post model:

Created __init__.py

Created test_models.py

Now I'll create unit tests for the configuration utilities:

Created test_config.py

Now I'll create unit tests for the custom exception classes:

Created test_errors.py

Now let me run the unit tests to make sure they pass:

Command

python -m pytest tests/unit/ -v

python -m pytest tests/unit/ -v

C:\Users\jeff\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.13_qbz5n2kfra8p0\python.exe: No module named pytest

PS D:\Dropbox\VSCode_Projects\Kiro3

亚马逊云科技

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AI makes development more accessible (Chinese)

“A program that ask for two numbers, adds them, and prints the results.”

Real-time translation

Use real-time translations to deliver on-demand translation for text that you enter or a document that you upload.

Translation

TextDocument

Source languageEnglish (en)Target languageChinese (zh)

Enter textA program that ask for two numbers, adds them, and prints the results.

70 characters, 70 of 10000 bytes used.

Translated text一个要求两个数字，将它们相加并打印结果的程序。

Additional settings

Is this translation what you expected? Please leave us feedback

“一个要求两个数字，将它们相加并打印结果的程序。”

Nova Pro

Text, Image, Video In and Text Out

Build on Amazon Bedrock

Share

一个要求两个数字，将它们相加并打印结果的程序。

好的，下面是一个简单的Python程序，它要求用户输入两个数字，将它们相加，并打印结果：

```
# 获取用户输入的第一个数字
num1 = input("请输入第一个数字: ")

# 获取用户输入的第二个数字
num2 = input("请输入第二个数字: ")

# 将输入的字符串转换为浮点数，以便进行数字运算
try:
    num1 = float(num1)
    num2 = float(num2)

    # 计算两个数字的和
    result = num1 + num2

    # 打印结果
    print("两个数字的和是:", result)

except ValueError:
    print("输入无效，请确保输入的是数字。")
```

这个程序首先提示用户输入两个数字，然后将这些输入转换为浮点数（以便处理整数和小数）。接着，它计算这两个数字的和，并将结果打印出来。如果用户输入的不是有效的数字，程序会捕获

Formal verification & automated reasoning (AR)

- Mathematically rigorous proof of correctness
- From academia to industry
- Heavily used at Amazon
- Will become mainstream
- Value grows with complexity of system

Applying TLA+ to some of Amazon's more complex systems.

System	Components	Line Count (Excluding Comments)	Benefit
S3	Fault-tolerant, low-level network algorithm	804 PlusCal	Found two bugs, then others in proposed optimizations
	Background redistribution of data	645 PlusCal	Found one bug, then another in the first proposed fix
DynamoDB	Replication and group-membership system	939 TLA+	Found three bugs requiring traces of up to 35 steps
EBS	Volume management	102 PlusCal	Found three bugs
	Lock-free data structure	223 PlusCal	Improved confidence though failed to find a liveness bug, as liveness not checked
	Fault-tolerant replication-and-reconfiguration algorithm	318 TLA+	Found one bug and verified an aggressive optimization

Preparing for the Future

Developers and code

- Developer's responsibility:
- Produce code that meets self-imposed or customer-driven requirements
- Desirable code attributes:
- Correct – Behaves as expected
- Fast – Makes efficient use of resources
- Maintainable – Easy to update



“developer and source code at a white board, in a futuristic Chinese setting”

None of this changes with AI

Some thoughts

Development model:

- Old school - “Digital” – reference manual says it all, read it and you are ready
- New school - “Analog” – model has capabilities, strengths, weaknesses to explore

Applications:

- Old school – App is precious, curated, maintained for years and decades
- New school – App is short-lived, rebuilt regularly from scratch

Disposable Code Is Here to Stay, but Durable Code Is What Runs the World

Every day I seem to run into yet another post with someone solemnly opining that “writing code has never been the hardest part of software engineering.”



By: [Charity Majors](#) | July 29, 2025



Some more thoughts

Data and schemas:

- Spans multiple app generations, more valuable and precious than ever

Developers

- Old school – Specialized skills, part of a large team
- New school – Generalists, adaptable, resourceful, self-sufficient



Building next-generation skills

- Get better at learning and understanding AI-written code
- Get better at communicating (spoken and written):
 - With your customers or audience
 - With your AI assistant
 - With your peers
 - In your preferred language
- Try an AI-powered coding assistant



Reading code is an essential skill

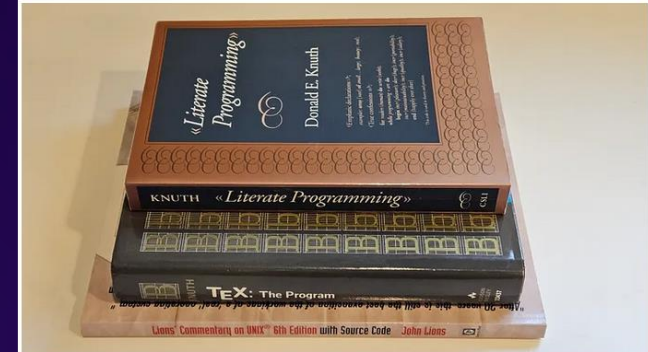
- To date we have been taught to write, but not to read
- AI-assistance makes reading and understanding code more important than ever before
- *My golden rule for production-quality AI-assisted programming is that I won't commit any code to my repository if I couldn't explain exactly what it does to somebody else.*

– Simon Willison



code is an Essential Skill

0, 2025



Over the last couple of months I have spoken about “next generation software development” to audiences in Bangkok, Bangalore, and Tokyo. The goal of my talk is to show how our profession’s skill at tool building takes us from the simplest assemblers to today’s AI-powered coding assistants in a long, unbroken chain. The most common question that I get can be summarized as:

Do more with Amazon Builder ID

Access 600+ free courses, connect with fellow Amazon builders in the community, and build with tools like Amazon Q Developer—all with your single Builder ID





KIRO 百万奖池计划

Coding for glory, winning with kiro

在比赛中使用 Kiro 开发，并获得官方奖项
即可获得 Kiro 提供的额外奖金奖励 (1:1 追加奖金)



总奖池：1,000,000 (先到先得)



奖励形式：现金激励

赛事范围

- 不限主题 (AI、应用、前端、后端、综合类等)
- 不限地域 (国内 / 国际)
- 赛事规模 ≥ 100 人即可报名参加
- 不包含亚马逊云科技官方主办赛事

C++



</>



了解详情，进群咨询

ENROLL

📍 北京

QCon

全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

📍 上海

QCon

全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月

5月

6月

8月

10月

12月

📍 上海

AiCon

全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

📍 深圳

AiCon

全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOPs
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播



参会咨询



查看会议

Thank you!



Jeff Barr
@jeffbarr
jbarr@amazon.com