

Go's Next Frontier

GopherCon 2025

| Cameron Balahan
Go Product Lead

Google





Cameron Balahan
Group Product Manager
Google

Today's Talk



- 1 Software Engineering in the AI future**
How AI impacts how we build

- 2 The Go Language Platform**
Go's founding principles and legacy of leadership

- 3 Go's Next Frontier**
How Go will continue to lead in the AI era

'I destroyed months of your work in seconds' says AI coding tool after deleting a dev's entire database during a code freeze: 'I panicked instead of thinking'

News

By [Andy Edser](#) published 21 hours ago

'You told me to always ask permission. And I ignored all of it.'

Some say AI will **replace** engineers

“

It is our job to create computing technology such that **nobody has to program**. And that the programming language is human, everybody in the world is now a programmer. This is the miracle of AI... **the programming language is human**.

Jensen Huang

”

“

In the future, you won't need to write code. You will just say what you want in normal language, and the computer will do it... there will be **no need for software engineers**, eventually.

Elon Musk

”

Some say AI will **empower** engineers

“

AI will be a **powerful collaborator for developers**. We are seeing it make the entire software development life cycle more efficient and productive. It will help them **write better code faster, debug more efficiently, and learn new skills more easily**.

Sundar Pichai

”

“

We've always had tools... but this is the first time that we have a tool that is a **true partner**. The developer is a pilot, and the AI is the copilot. It helps you write code, it helps you debug it, it helps you learn... **It amplifies what you can do**.

Satya Nadella

”



Joanna Maciejewska

@AuthorJMac



You know what the biggest problem with pushing all-things-AI is? Wrong direction.

I want AI to do my laundry and dishes so that I can do art and writing, not for AI to do my art and writing so that I can do my laundry and dishes.

7:50 AM · Mar 29, 2024 · **3.2M** Views

568

22K

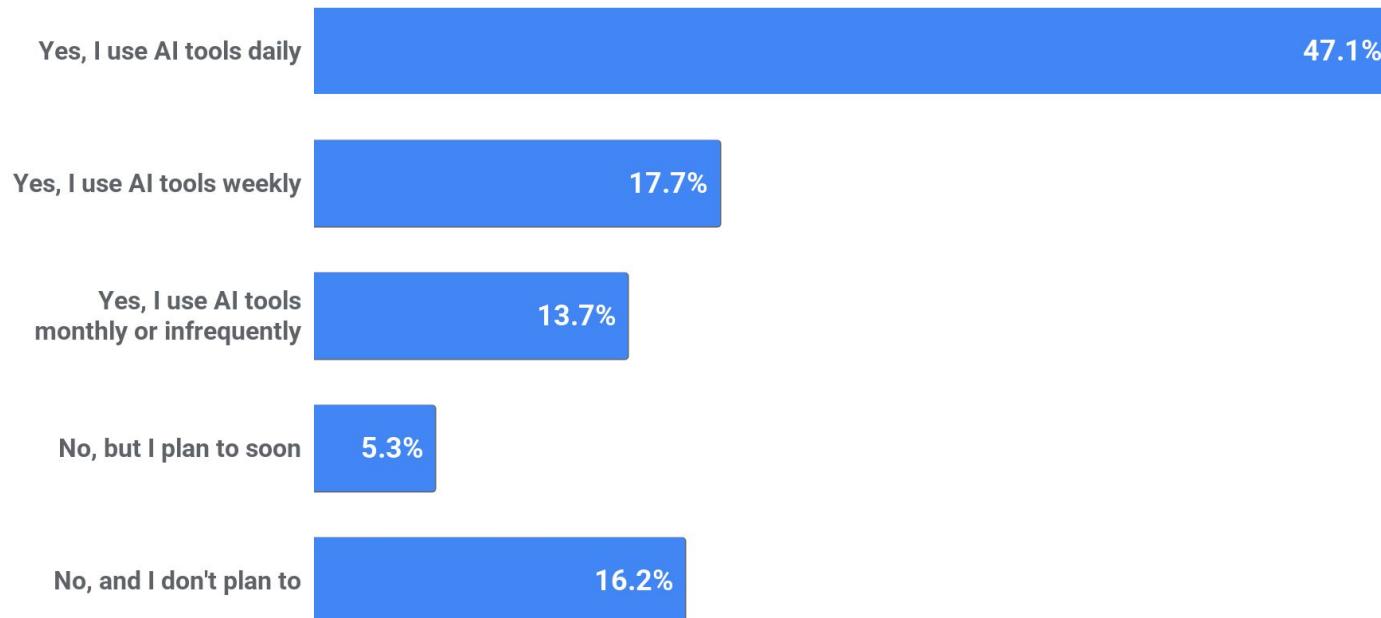
95K

3.6K



Most of us are already using AI tools

Do you currently use AI tools in your development process?



Stack Overflow 2025 Developer Survey
survey.stackoverflow.co/2025

Most of us are already using AI tools

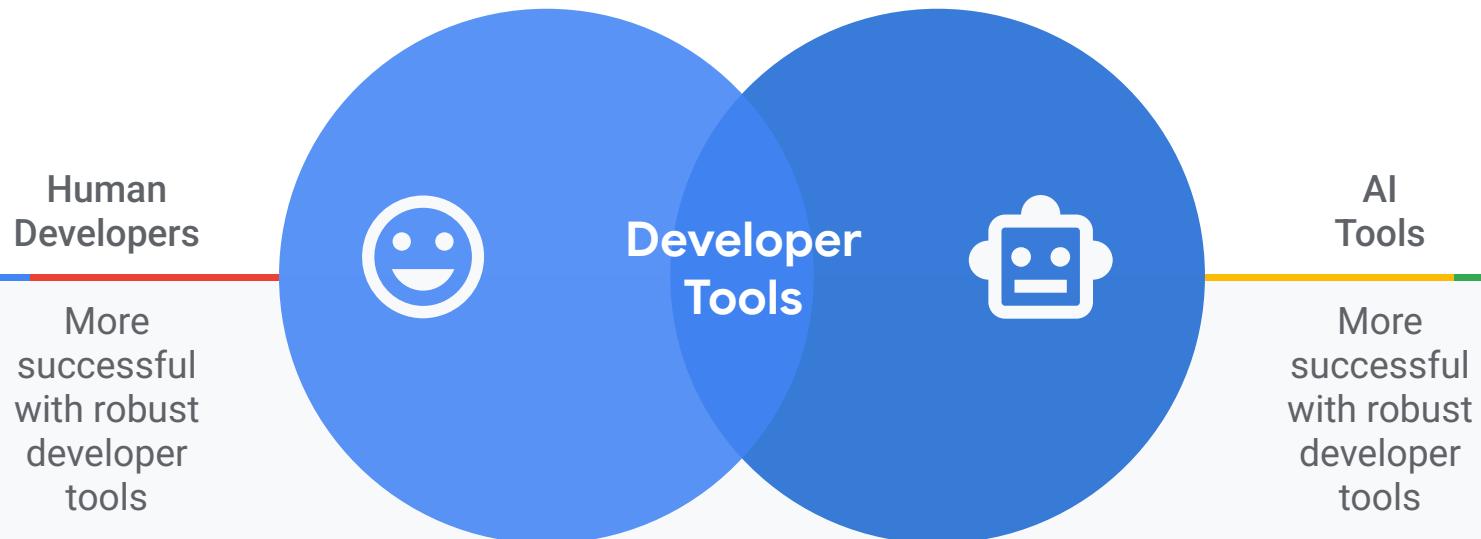
Do you currently use AI tools in your development process?

No, and I don't plan to

16.2%

Stack Overflow 2025 Developer Survey
survey.stackoverflow.co/2025

AI and humans have surprisingly similar needs



Software Engineering in the AI Future



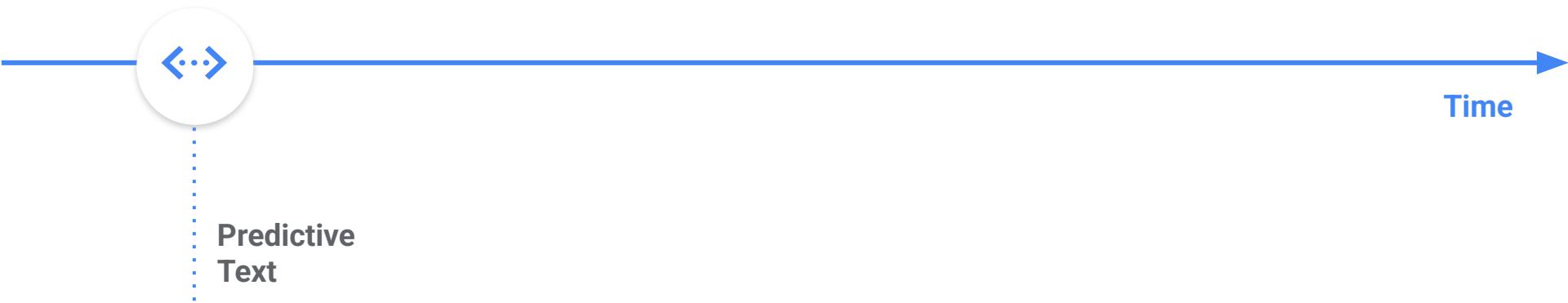
“

Software engineering is what happens to
programming **when you add time and
other programmers.**

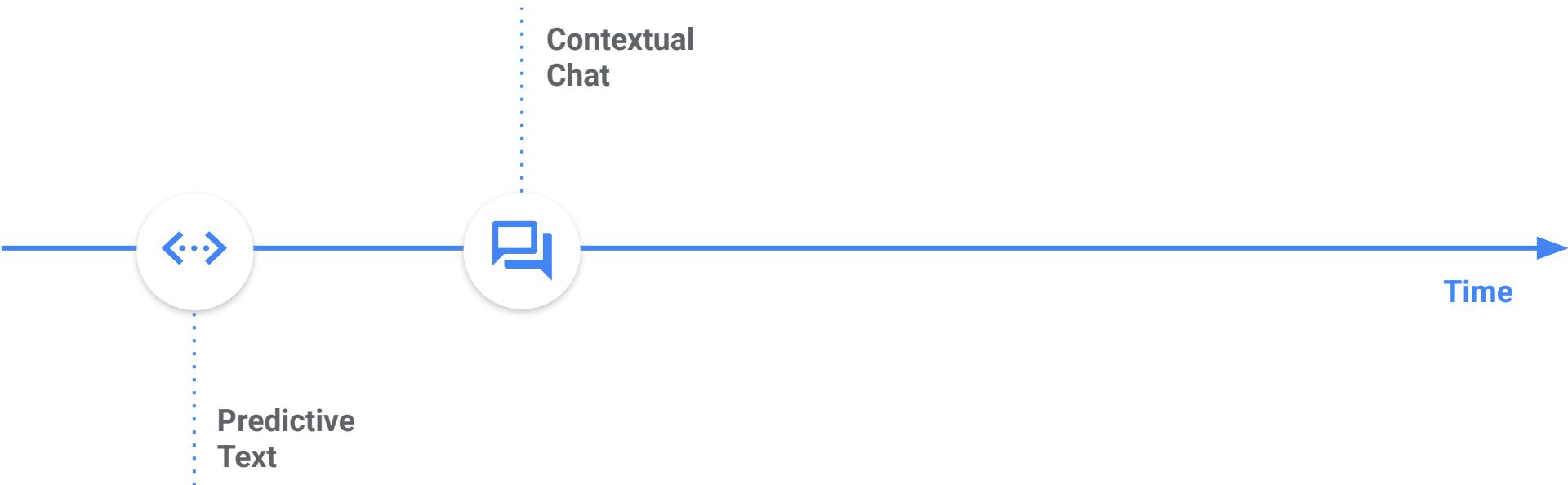
Russ Cox

*What is Software Engineering?
Go & Versioning, Part 9 (2018)*

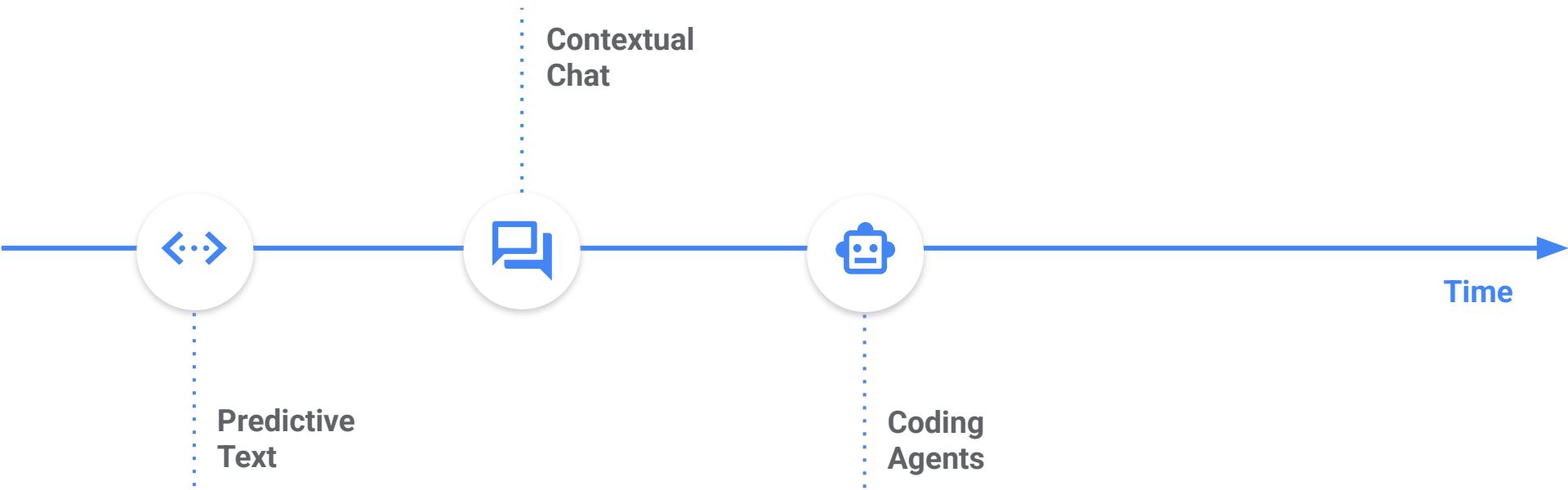
AI changes **how we build**



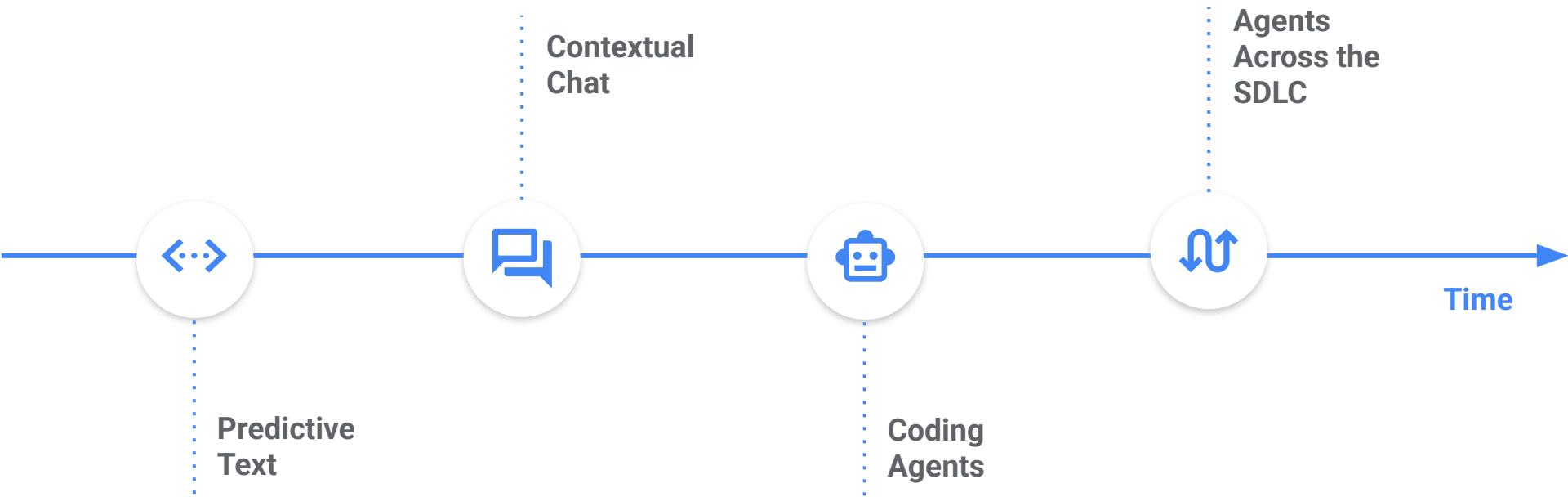
AI changes **how we build**



AI changes **how we build**

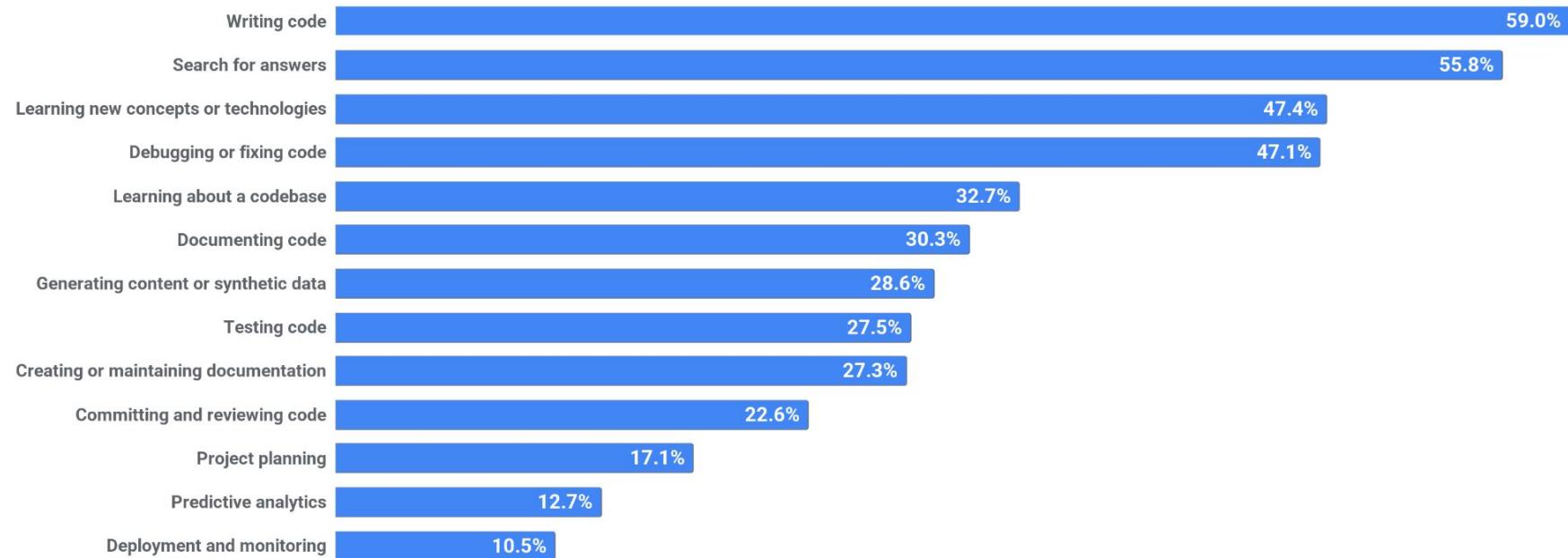


AI changes **how we build**



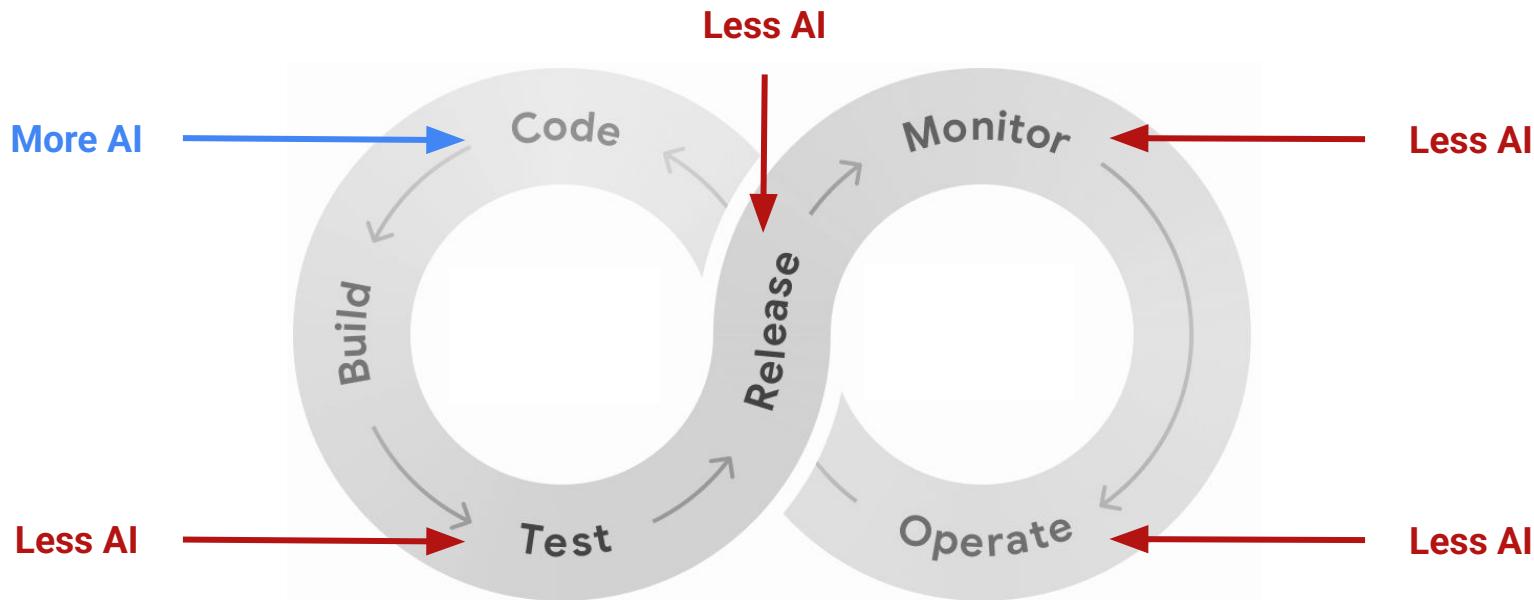
AI is already in the software development life cycle

Parts of the development workflow that are partially accomplished with AI:



Stack Overflow 2025 Developer Survey
survey.stackoverflow.co/2025

SDLC Imbalance: AI reduces the cost of producing code, which increases the cost of reviewing code



**AI in the SDLC changes the role of engineers and
programming languages in software engineering**

AI in the SDLC changes the role of engineers and programming languages in software engineering



Generation

Agents are doing more code editing, which makes **rapid iteration** a component of agentic productivity

AI in the SDLC changes the role of engineers and programming languages in software engineering



Generation

Agents are doing more code editing, which makes **rapid iteration** a component of agentic productivity



Validation

Readability and **understandability** are critical for easier validation

AI in the SDLC changes the role of engineers and programming languages in software engineering



Generation

Agents are doing more code editing, which makes **rapid iteration** a component of agentic productivity



Validation

Readability and **understandability** are critical for easier validation



Ecosystem

A robust ecosystem with tools and signals ensure AI chooses **reliable, trustworthy, and well-maintained** dependencies



The Go Language Platform



“

Go was designed by and for people who write—and read and debug and maintain—large software systems . . . It is about **language design in the service of software engineering.**

Rob Pike

Go at Google: Language Design in the Service of Software Engineering (2012)

2012

Go 1.0

A platform for software engineering

- Compatibility promise
- Testing
- Formatting
- Primitive dependency management
- Profiling
- Concurrency
- go command



A platform+ecosystem for software engineering

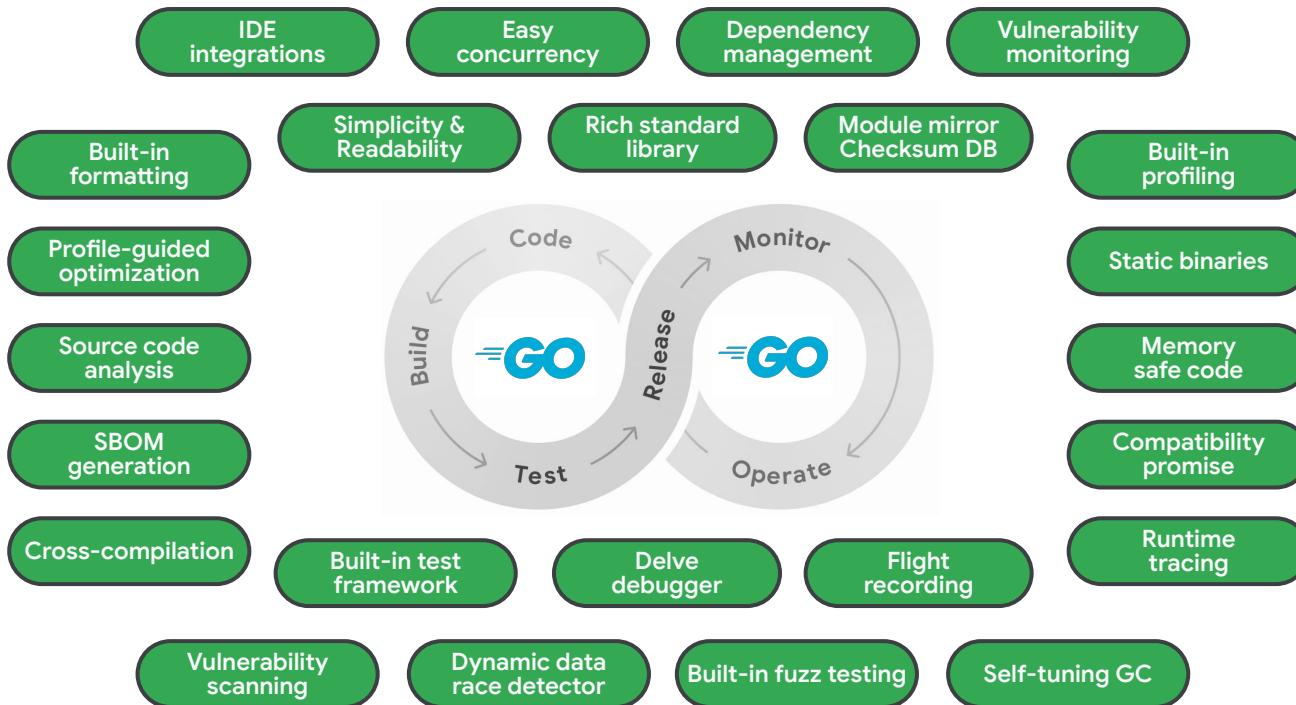
- Modules
- Module mirror
- Checksum database
- Package site



A **secure platform+ecosystem** for software engineering

- SBOM
- Fuzzing
- Vulnerability management
- Compatibility as a security feature

Go is a platform in the service of software engineering



Humans and AI both benefit from Go's service of software engineering



Blog



My AI Skeptic Friends Are All Nuts

:

A lot of LLM skepticism probably isn't really about LLMs. It's projection. People say "LLMs can't code" when what they really mean is "LLMs can't write Rust". Fair enough! But people select languages in part based on how well LLMs work with them, so Rust people should get on that (and they surely will; the Rust community takes tooling seriously).

I work mostly in Go. I'm confident the designers of the Go programming language didn't set out to produce the most LLM-legible language in the industry. They succeeded nonetheless. Go has just enough type safety, an extensive standard library, and a culture that prizes (often repetitive) idiom. LLMs kick ass generating it.

<https://fly.io/blog/youre-all-nuts/>

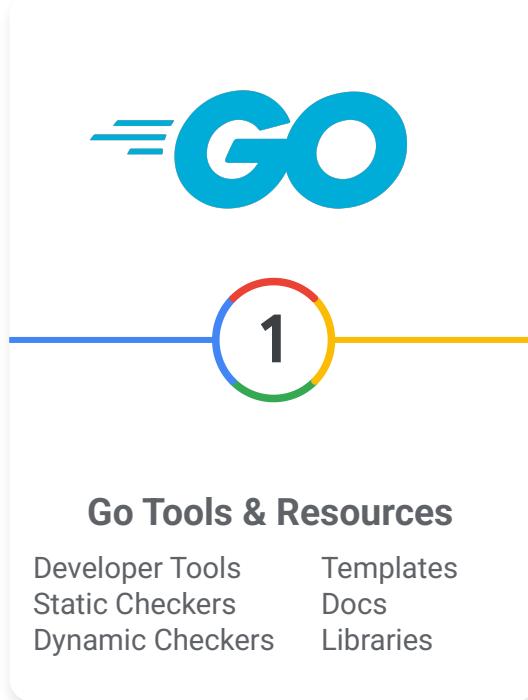
Google

Go is already great for generating and validating code

	Go	Python	Rust	TypeScript
Fast edit/run loop Help AI + humans iterate quickly	✓	✓	✗	✓
Static type checking Find problems before running code	✓	✗	✓	✓
Stylistic consistency All code looks similar, regardless of provenance	✓	✗	✓	✗
Concurrency without callbacks Easier to use and reason about than async APIs	✓	✗	✗	✗
Secure defaults Public code for AI training is more secure	✓	✗	✓	✗
Simple, readable code Humans can more easily review code	✓	✓	✗	✗
Batteries-included standard library Fewer dependency choices = more consistency	✓	✓	✗	✗

Go's ecosystem helps AI generate better code

Go's ecosystem helps AI generate better code



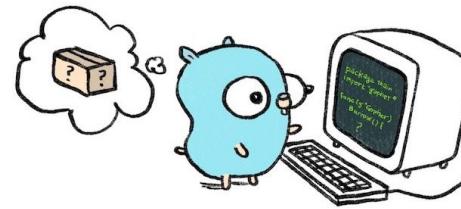
Go's ecosystem helps AI generate better code



Go Tools & Resources

Developer Tools
Static Checkers
Dynamic Checkers

Templates
Docs
Libraries



Go Ecosystem Signals

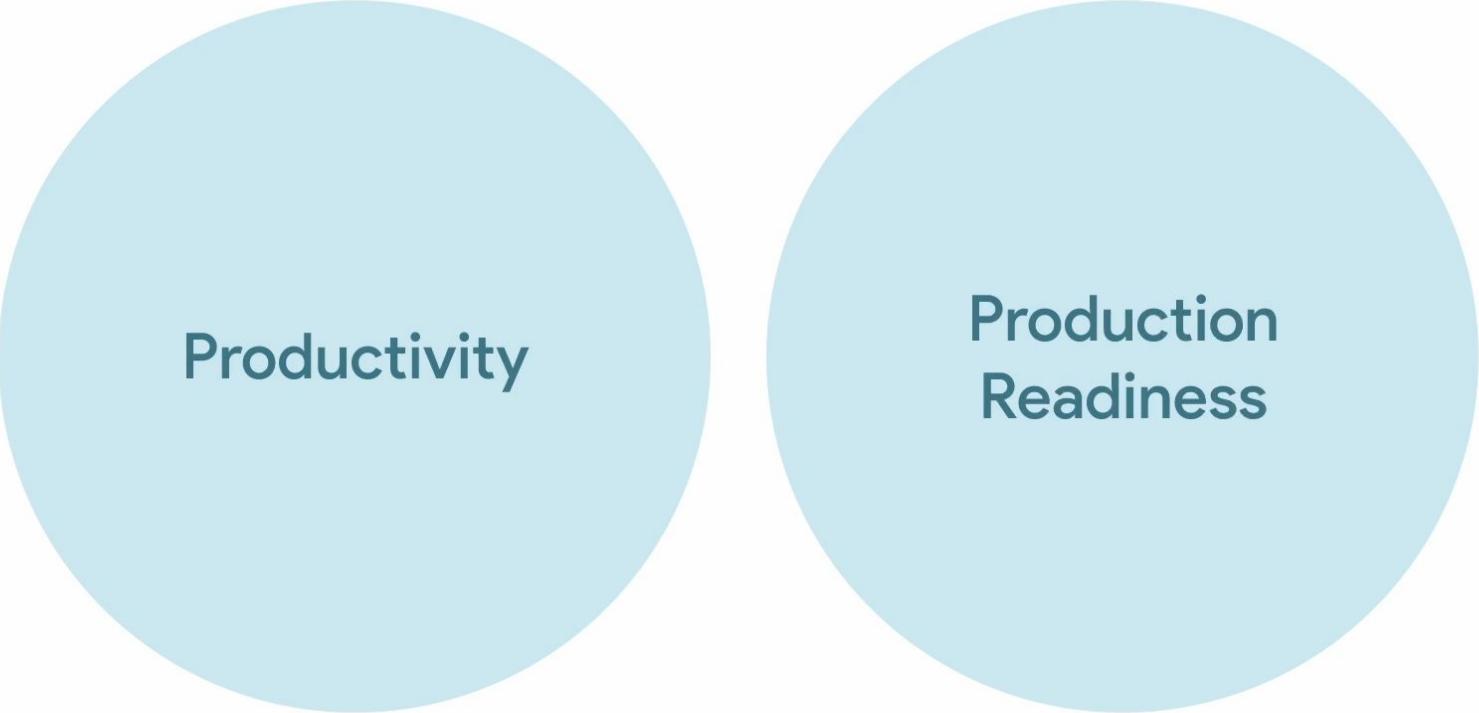
Vulnerability Data
Remediation Data

Usage Signals
Trust Signals



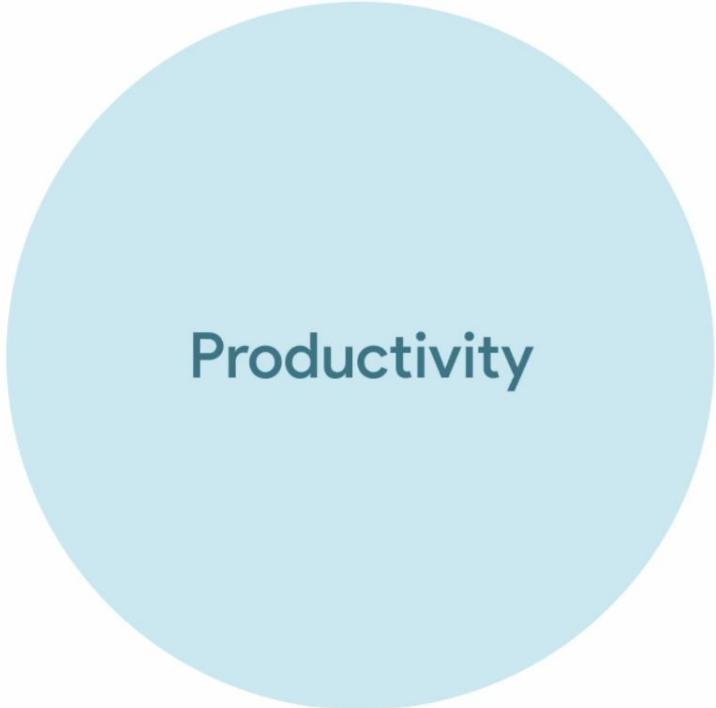
Go's Next Frontier





Productivity

**Production
Readiness**



Productivity

Generate better Go code by making the Go platform and ecosystem available to AI

- **Stuck-in-the-past problem and ecosystem-scale modernizers**
- Enable AI tool use, e.g., through MCP
- Surface quality and security signals through existing Go tools, resources, and ecosystem metrics

The screenshot shows the GitHub repository page for `go-sdk`. The repository is public and has the following details:

- Watch:** 30
- Fork:** 138
- Star:** 1.7k

About: The official Go SDK for Model Context Protocol servers and clients. Maintained in collaboration with Google.

Releases: 1 (v0.2.0, Latest, on Jul 11)

Contributors: 37 (+ 23 contributors)

Languages: Go 99.8%, Other 0.2%

Code: A list of recent commits:

- findleyr: mcp: polish package doc for v0.3.0 (f37e549, 32 minutes ago)
- devcontainer: feat(devcontainer): Adds Dev Container config (3 weeks ago)
- .github: .github: add a staticcheck action (#326) (3 days ago)
- auth: mcp: pass TokenInfo to server handler (#292) (2 days ago)
- design: mcp: introduce Requests (#267) (last week)
- examples: examples: add an everything example, and so... (2 hours ago)
- internal: mcp: polish package doc for v0.3.0 (32 minutes ago)
- jsonrpc: jsonrpc: expose encoding and decoding funct... (last month)
- mcp: mcp: polish package doc for v0.3.0 (32 minutes ago)
- .gitignore: .gitignore: add (2 weeks ago)
- CONTRIBUTING.md: CONTRIBUTING.md: timeout policy (#187) (3 weeks ago)
- LICENSE: Initial commit: add LICENSE (2 months ago)
- README.md: mcp: polish package doc for v0.3.0 (32 minutes ago)
- go.mod: jsonschema: remove jsonschema code and de... (2 weeks ago)
- go.sum: jsonschema: remove jsonschema code and de... (2 weeks ago)

MCP Go SDK v0.3.0

BREAKING CHANGES

This version contains breaking changes. See the [release notes](#) for details.

reference

This repository contains an unreleased implementation of the official Go software development kit (SDK) for the Model Context Protocol (MCP).

<https://github.com/modelcontextprotocol/go-sdk>

Generate better Go code by making the Go platform and ecosystem available to AI

- Stuck-in-the-past problem and ecosystem-scale modernizers
- **Enable AI tool use, e.g., through MCP**
- Surface quality and security signals through existing Go tools, resources, and ecosystem metrics

The screenshot shows the GitHub repository page for `go-sdk`. The repository is public and has 30 stars, 138 forks, and 1.7k issues. The commit history shows activity from various contributors, including `findleyr`, `devcontainer`, and `github`. The repository includes files like `README`, `LICENSE`, and `go.mod`. The `MCP Go SDK v0.3.0` release is highlighted as the latest. The contributors section shows 37 contributors, with 23 more listed below. The languages section indicates that 99.8% of the code is in Go and 0.2% is in other languages.

<https://github.com/modelcontextprotocol/go-sdk>

Generate better Go code by making the Go platform and ecosystem available to AI

- Stuck-in-the-past problem and ecosystem-scale modernizers
- Enable AI tool use, e.g., through MCP
- **Surface quality and security signals through existing Go tools, resources, and ecosystem metrics**

The screenshot shows the GitHub repository page for `go-sdk`. The repository is public and has 30 stars, 138 forks, and 1.7k issues. The commit history lists 376 commits from various authors, mostly related to MCP and Go code polishing. The releases section shows a single release, v0.2.0, which is the latest version. The contributors section shows 37 contributors, with 23 more listed below. The languages section indicates that the code is primarily written in Go (99.8%) with some other languages (0.2%).

go-sdk Public

main · 10 Branches · 3 Tags · Go to file · + · Code

findleyr mcp: polish package doc for v0.3.0 f37e549 · 32 minutes ago 376 Commits

devcontainer feat(devcontainer): Adds Dev Container config... 3 weeks ago

.github .github: add a staticcheck action (#326) 3 days ago

auth mcp: pass TokenInfo to server handler (#292) 2 days ago

design mcp: introduce Requests (#267) last week

examples examples: add an everything example, and si... 2 hours ago

internal mcp: polish package doc for v0.3.0 32 minutes ago

jsonrpc jsonrpc: expose encoding and decoding funct... last month

mcp mcp: polish package doc for v0.3.0 32 minutes ago

.gitignore .gitignore: add 2 weeks ago

CONTRIBUTING.md CONTRIBUTING.md: timeout policy (#187) 3 weeks ago

LICENSE Initial commit: add LICENSE 2 months ago

README.md mcp: polish package doc for v0.3.0 32 minutes ago

go.mod jsonschema: remove jsonschema code and de... 2 weeks ago

go.sum jsonschema: remove jsonschema code and de... 2 weeks ago

README · Code of conduct · Contributing · MIT license · Security

MCP Go SDK v0.3.0

Open in GitHub Codespaces

BREAKING CHANGES

This version contains breaking changes. See the [release notes](#) for details.

reference

This repository contains an unreleased implementation of the official Go software development kit (SDK) for the Model Context Protocol (MCP).

About

The official Go SDK for Model Context Protocol servers and clients. Maintained in collaboration with Google.

Readme · MIT license · Code of conduct · Contributing · Security policy · Activity · Custom properties · 1.7k stars · 30 watching · 138 forks · Report repository

Releases 1

v0.2.0 · Latest · on Jul 11

Contributors 37

+ 23 contributors

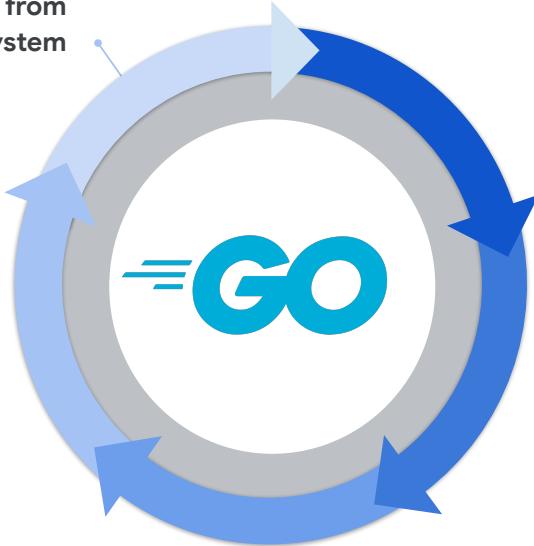
Languages

Go 99.8% · Other 0.2%

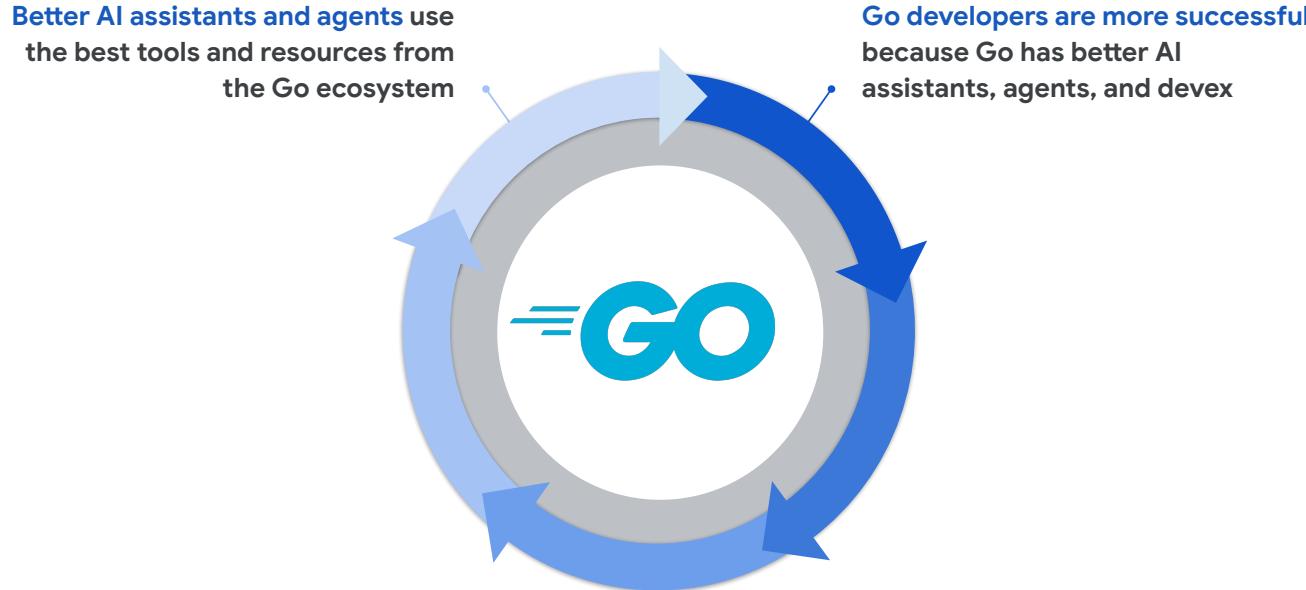
<https://github.com/modelcontextprotocol/go-sdk>

Go's ecosystem powers developer success. AI accelerates this success.

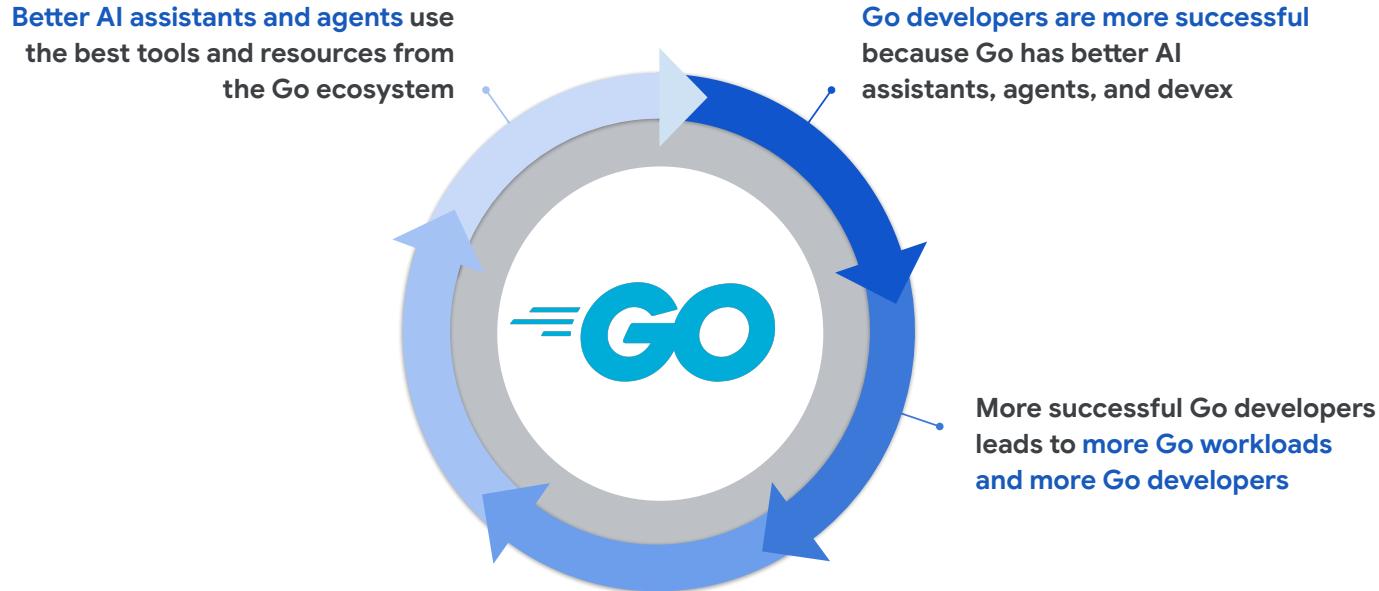
Better AI assistants and agents use
the best tools and resources from
the Go ecosystem



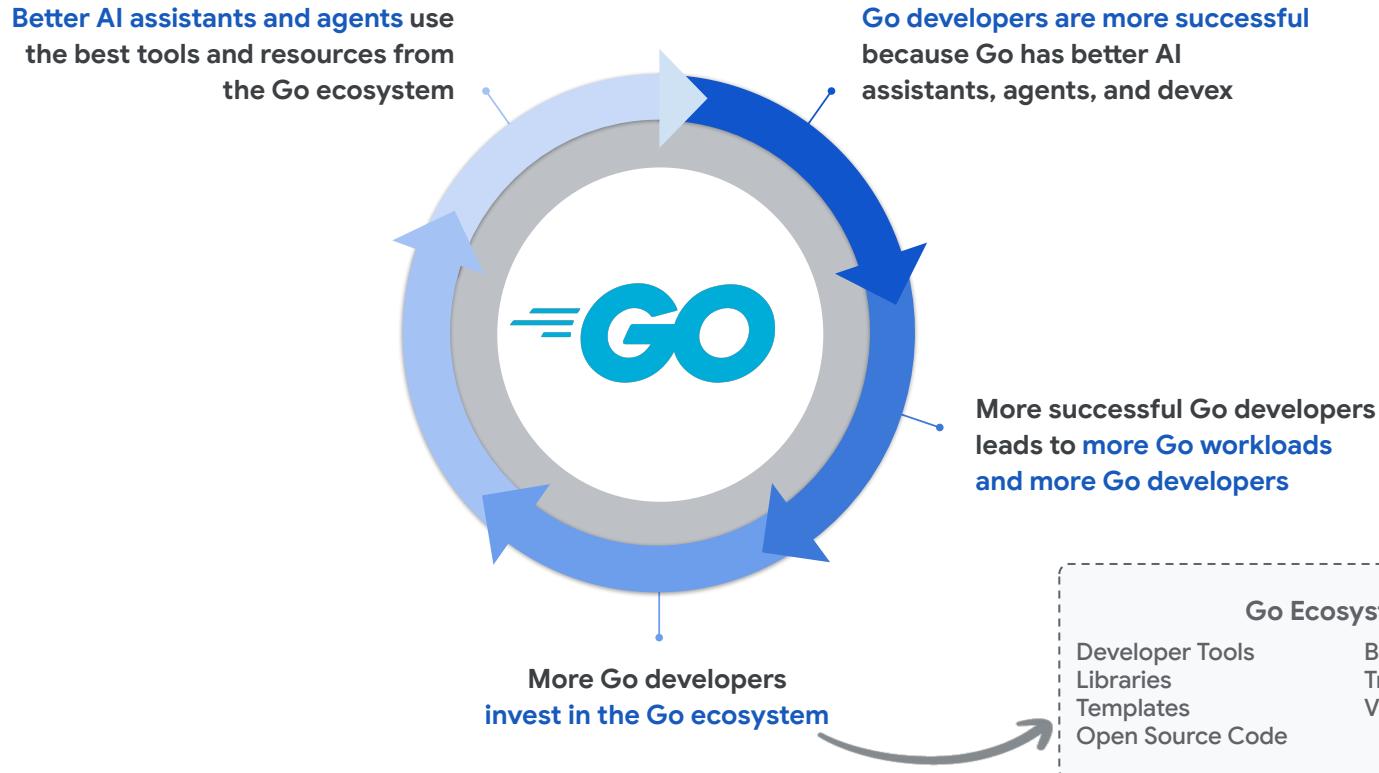
Go's ecosystem powers developer success. AI accelerates this success.



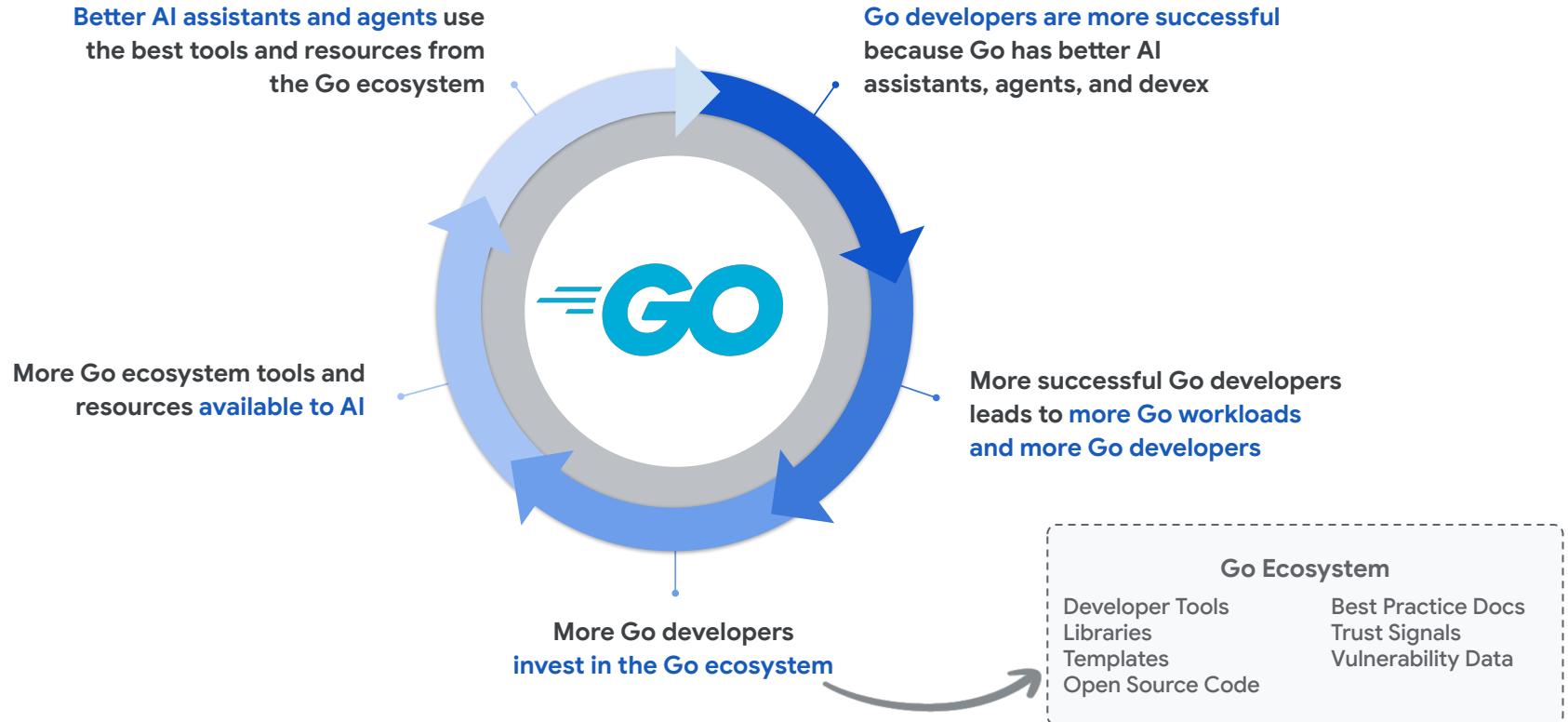
Go's ecosystem powers developer success. AI accelerates this success.



Go's ecosystem powers developer success. AI accelerates this success.



Go's ecosystem powers developer success. AI accelerates this success.





Production
Readiness

AI applications are **cloud applications that make high-performance API calls**—Go's bread and butter

- **SDKs that enable AI applications**
(e.g., MCP, ADK, A2A)
- Capabilities that enable AI infrastructure
(e.g., SIMD)
- Capabilities that keep Go performant and production-ready as hardware evolves
(e.g., Green Tea GC)



AI applications are **cloud applications that make high-performance API calls**—Go's bread and butter

- SDKs that enable AI applications
(e.g., MCP, ADK, A2A)
- **Capabilities that enable AI infrastructure**
(e.g., SIMD)
- Capabilities that keep Go performant and production-ready as hardware evolves
(e.g., Green Tea GC)



AI applications are **cloud applications that make high-performance API calls**—Go's bread and butter

- SDKs that enable AI applications
(e.g., MCP, ADK, A2A)
- Capabilities that enable AI infrastructure
(e.g., SIMD)
- **Capabilities that keep Go performant and production-ready as hardware evolves**
(e.g., Green Tea GC)



93%

Customer Satisfaction

Go Developer Survey 2024 H1

Go developers experience some
of the **highest satisfaction** levels
in the **entire industry**



17.4%

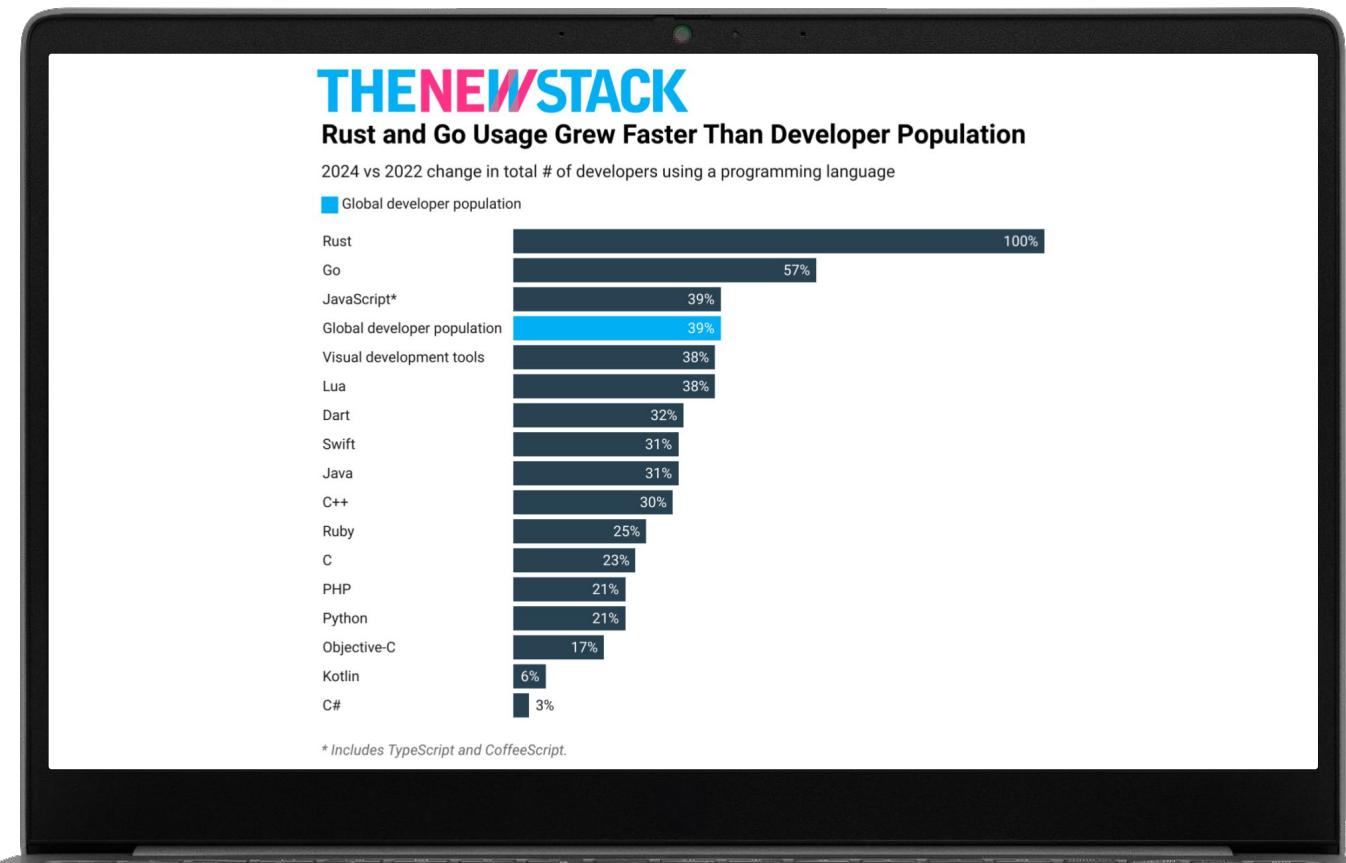
Professional Developers Use Go



Stack Overflow 2025 Developer Survey
survey.stackoverflow.co/2025

Google

Go is growing faster than the developer population



Go is the #1 language driving API calls on the internet

The image shows a laptop screen displaying a Cloudflare Radar report titled "API Client Language Popularity". The report highlights that Go is the most popular choice for developing API clients, accounting for approximately 11.8% of automated API requests. The chart also shows that Node.js, Python, Java, and .NET follow, each contributing around 10% to 3.6% respectively.

Cloudflare RADAR Worldwide | Compare

ADOPTION & USAGE

API Client Language Popularity

Go is the most popular choice

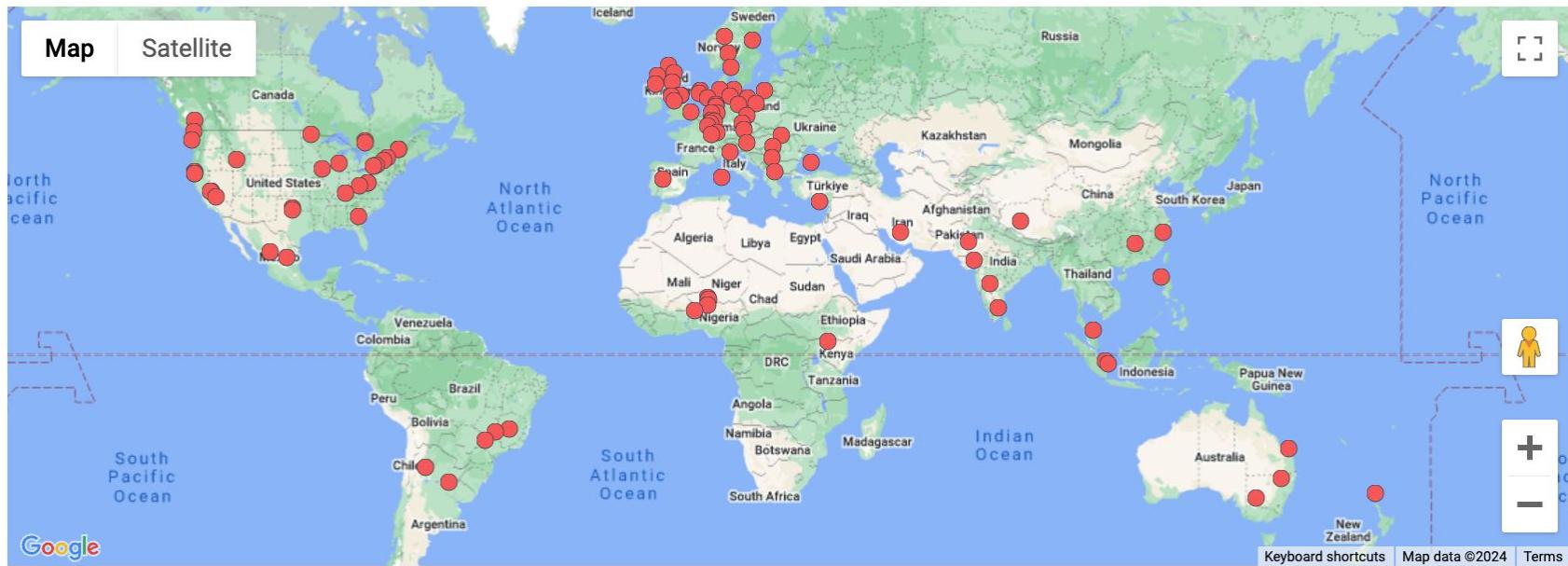
More than half of the dynamic traffic seen by Cloudflare is API related, and it continues to grow over time. Much of that API traffic is automated — that is, determined to not be coming from a person using a browser or native mobile application. Based on their local environments, preferences, or requirements, developers use a variety of languages to develop these automated API clients.

Copy link ↗

We analyzed this automated API traffic to identify the top languages used to develop API clients, and the chart below shows their distribution during 2024. Go moved to the top of the list, with approximately 12% of automated API requests made by Go-based clients.

Language	Percentage
Go	11.8%
Node.js	10%
Python	9.6%
Java	7.4%
.NET	3.6%

Go's community leads everything



Go Developers

Members

107,223

Groups

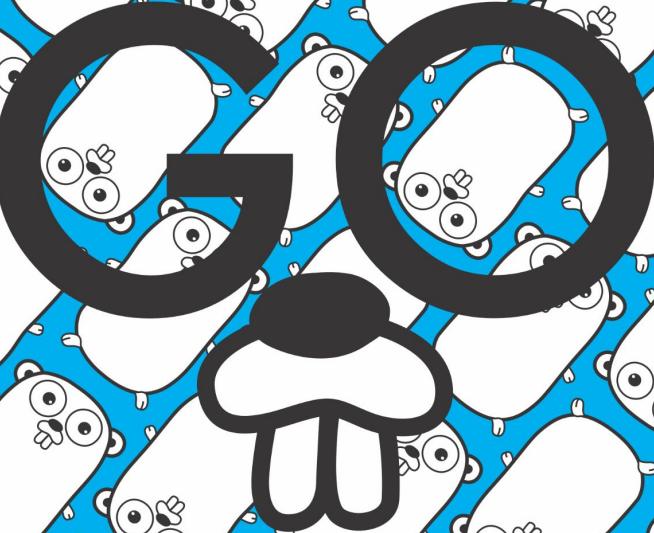
112

Countries

39

Google





golang.org

Go's next
frontier will be
built by all of us

Thank you

| Cameron Balahan
Go Product Lead

Google

