



S T A T E O F T H E

GOPHER NATION

A U G

2 0 1 7

STATE OF THE

GOPHER NATION

AUG 2017

Golang UK
CONFERENCE

Railway Britain



Globetrotter
partner programme

WRITTEN & DELIVERED BY
STEVE FRANCIA ([spf13](#))

FOR [GOLANGUK 2017](#)



WHERE WE'VE BEEN



MAJOR GO
MILESTONES

GO'S TIMELINE

Major milestones
for the Go Project
over the last 10
years

2007

2009

2012



Birth

Started at Google as a
20% project

Open Source

2 years later, Go is
open sourced

1.0

After nearly 5 years of
development, Go 1.0 is
released and attention
shifts to using Go

2014



GopherCon

The first major Go conference held. By all accounts, it was an overwhelming success

2015



Go In Go

As of Go 1.5 the Go compiler and runtime are written in Go

2015



WWG & GOB

Women Who Go and GoBridge are formed to support inclusion and diversity

2016



SSA

The SSA backend is released in Go 1.8, bringing further speed improvements.

2017



Go 2 annc.

Russ Cox announces the intent to start working on Go 2



ACCOMPLISHMENTS

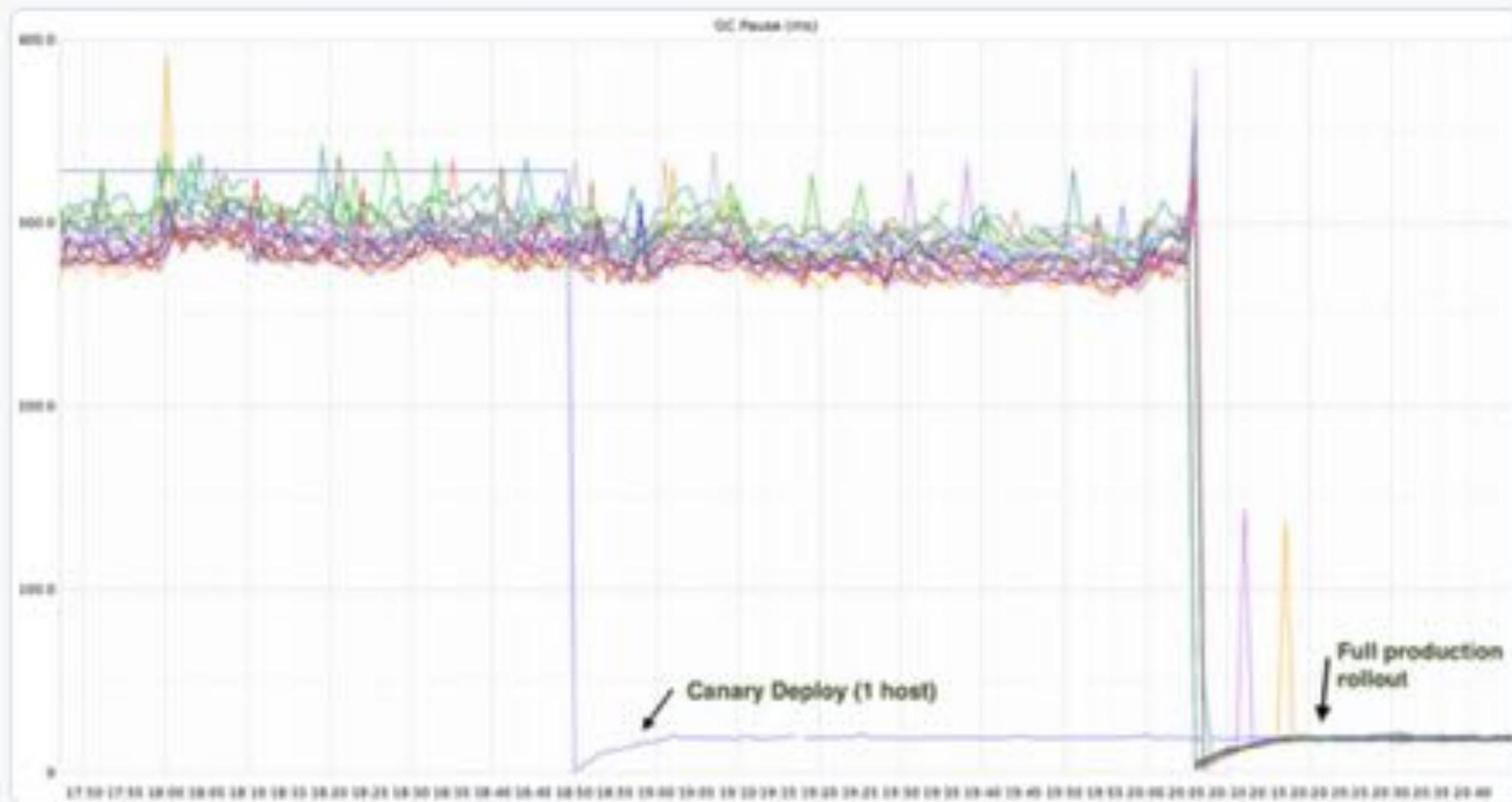


GO'S GARBAGE COLLECTION : A STORY IN TWEETS



Brian Hatfield @brianhatfield · 19 Aug 2015

Amazing GC pause time improvements in Go 1.5.



◀ 10

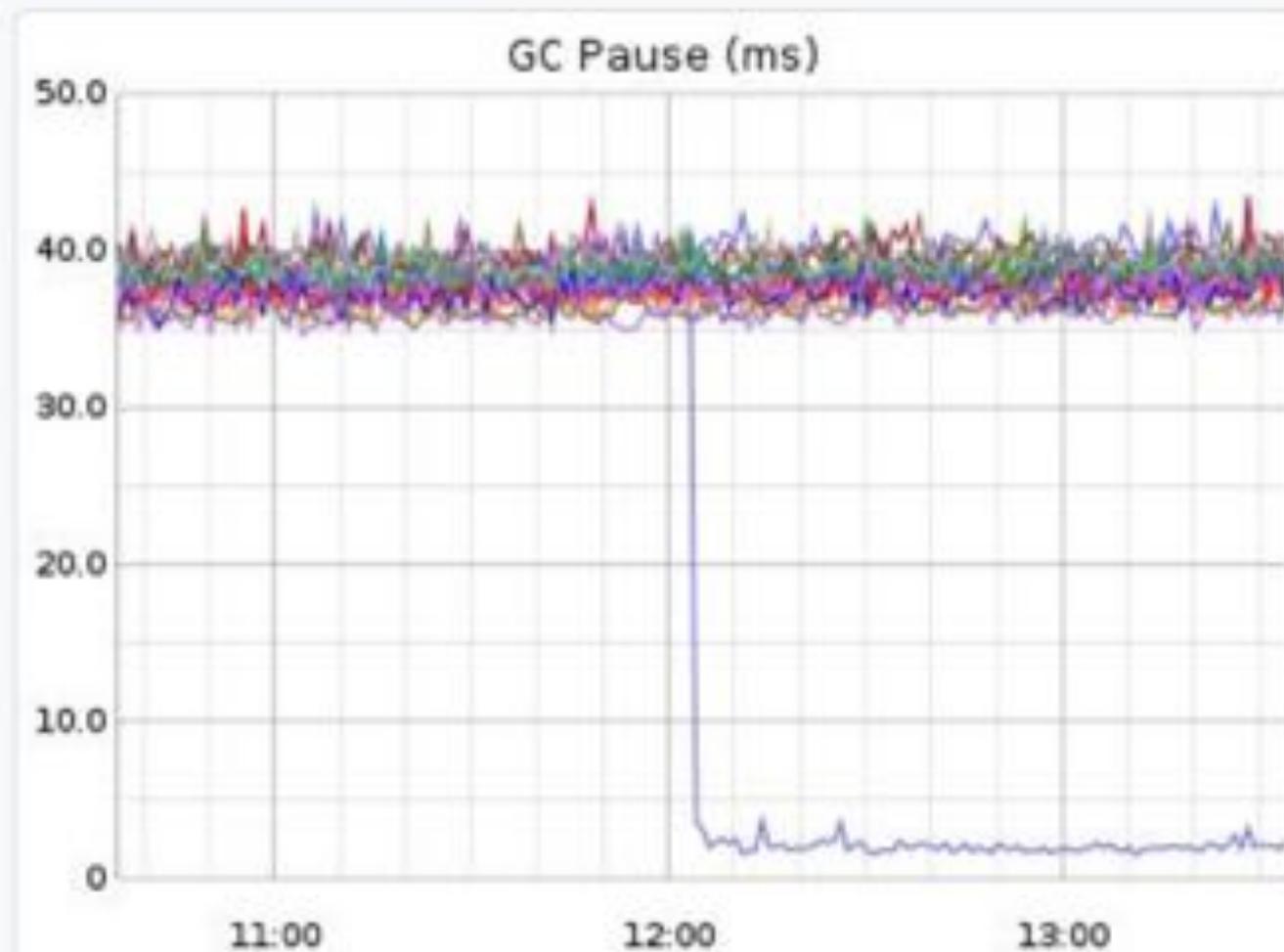
↑↓ 227

❤ 230



Brian Hatfield @brianhatfield · 28 Jan 2016

They did it again in Go 1.6 RC 1!



4

144

165



Brian Hatfield @brianhatfield · 22 Aug 2016

Excited to canary Go 1.7! Continued improvement in GC pause, and improvements in various request latencies/perf!

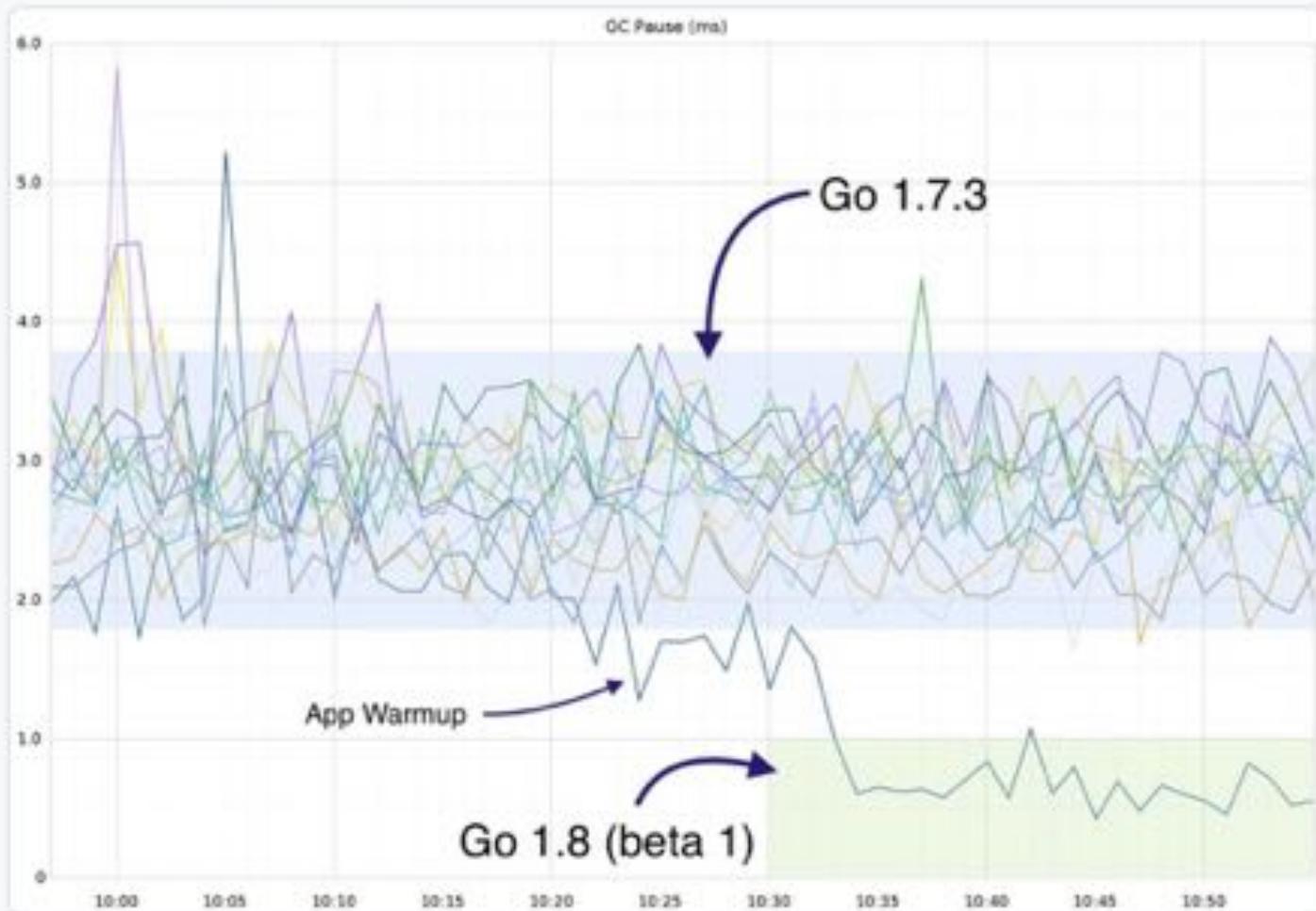




Brian Hatfield @brianhatfield · 1 Dec 2016

SUB. MILLISECOND. PAUSE. TIME. ON. AN. 18. GIG. HEAP.

(Trying out Go 1.8 beta 1!)



Most Loved, Dreaded, and Wanted Languages

Loved

Dreaded

Wanted

Rust

Smalltalk

TypeScript

Swift

Go

Loved

Dreaded

Wanted

73.1%
64.1%
63.9%

Go #5 most loved, #3 most wanted on Stack Overflow

Python

20.6%

JavaScript

18.6%

Go

13.5%



HashiCorp

Terraform



HashiCorp

Packer



HashiCorp

Consul



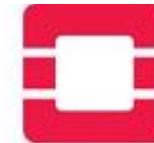
LANTERN



Grafana



etcd



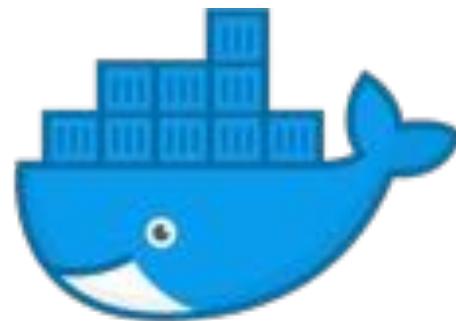
openstack.



NATS



kubernetes



docker



Caddy



Syncthing

Go is go to
language for:

- Reliability
- Scalability
- Durability
- Simplicity
- Performance



CHALLENGES

What changes would improve Go most?

Dependency Management

Go's Dependency Management story

isn't good (yet). We realized our deficiencies in this space later than we should have.



New User Experience

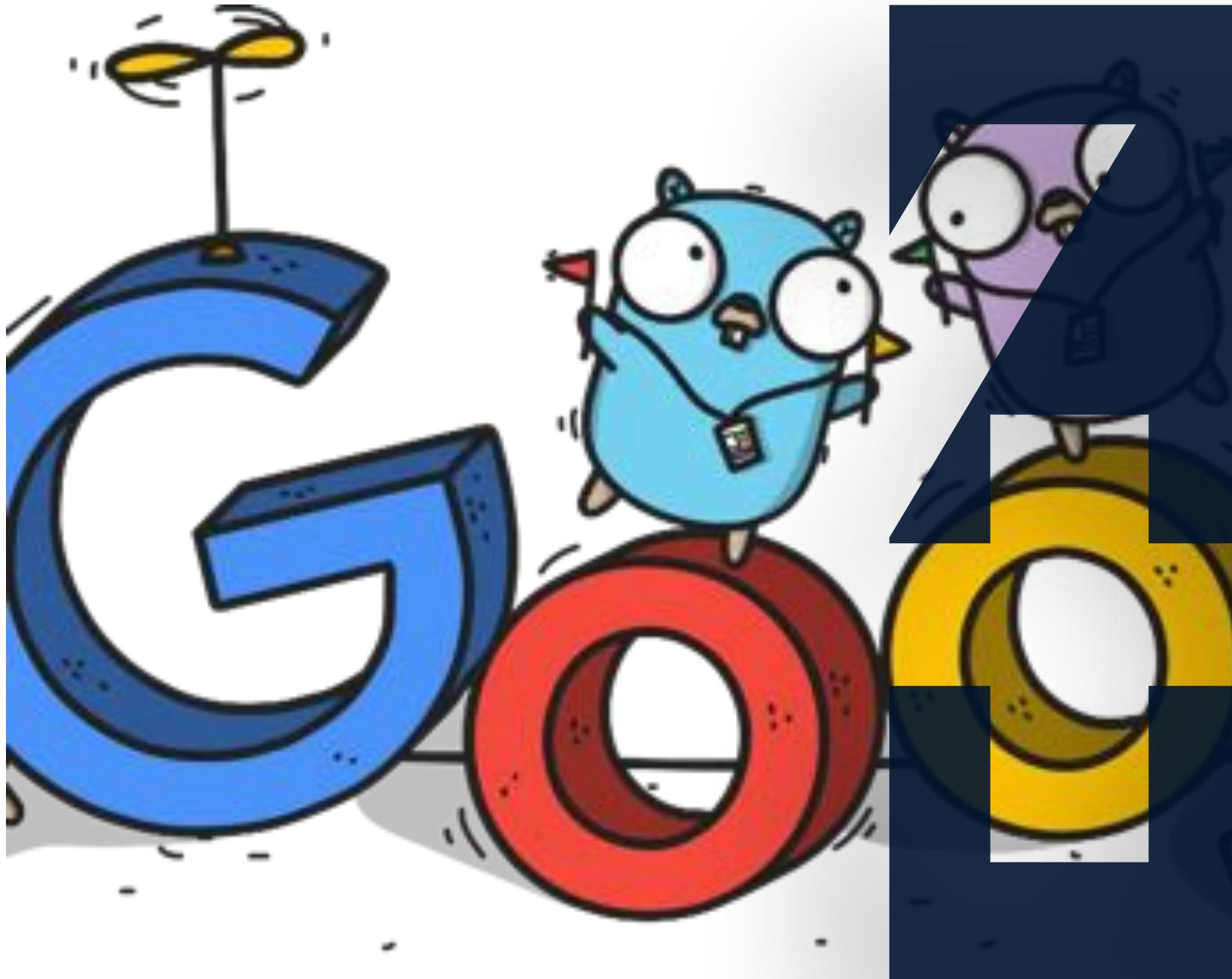
The Go new user experience is intimidating and needlessly complex. It often takes users 30 - 90 minutes before they are able to use Go.





Community Culture

Like many developer communities Go has had its struggles with diversity, inclusiveness and elitism.

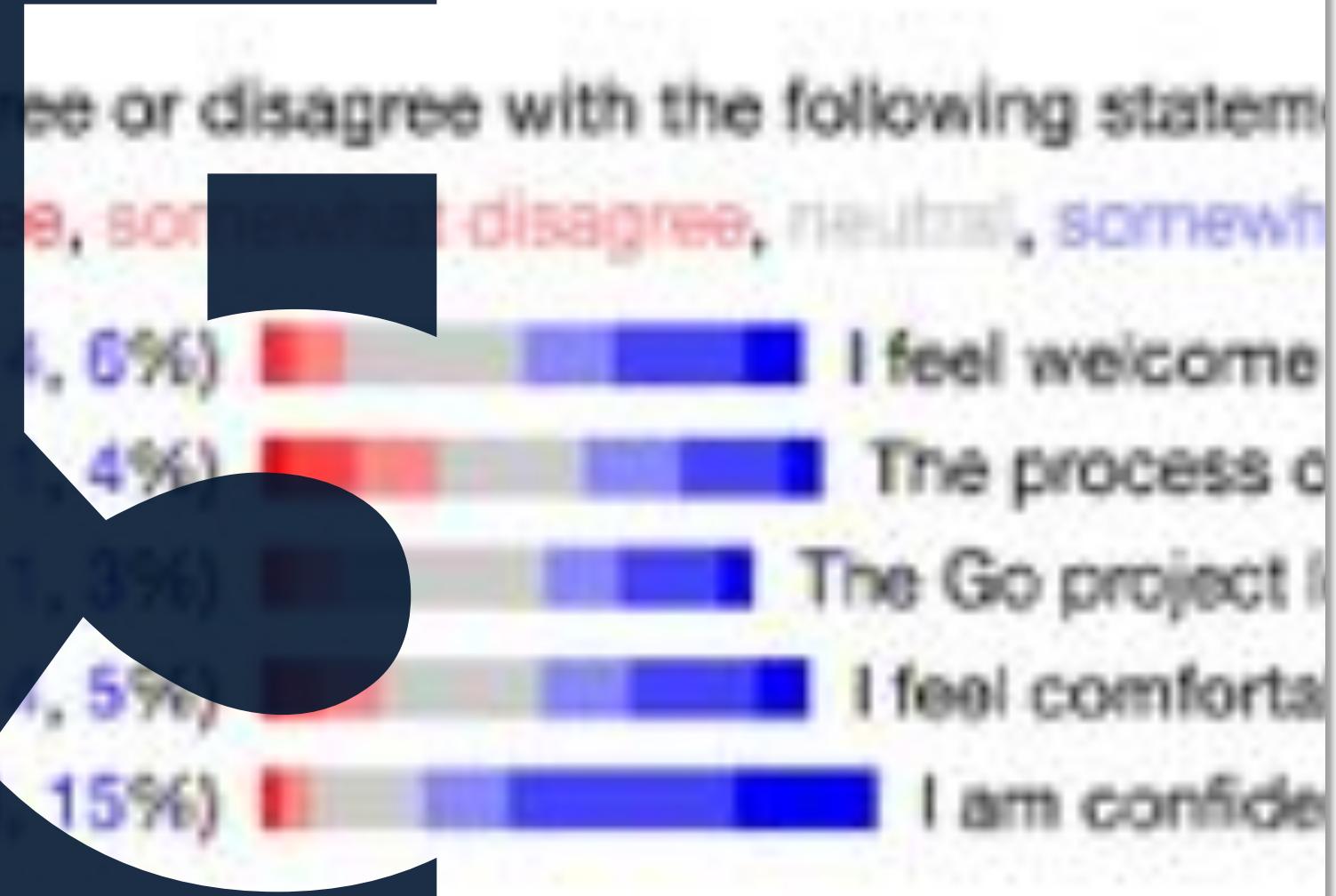


Single Perspective

Go's development has largely been led by Google. Google's engineers' experiences have helped shape Go into the language we love, but has also led to a limited perspective where experiences by others aren't well understood.

Project Participation

It's too hard and too intimidating for many in the community to contribute to the Go project.



TODAY

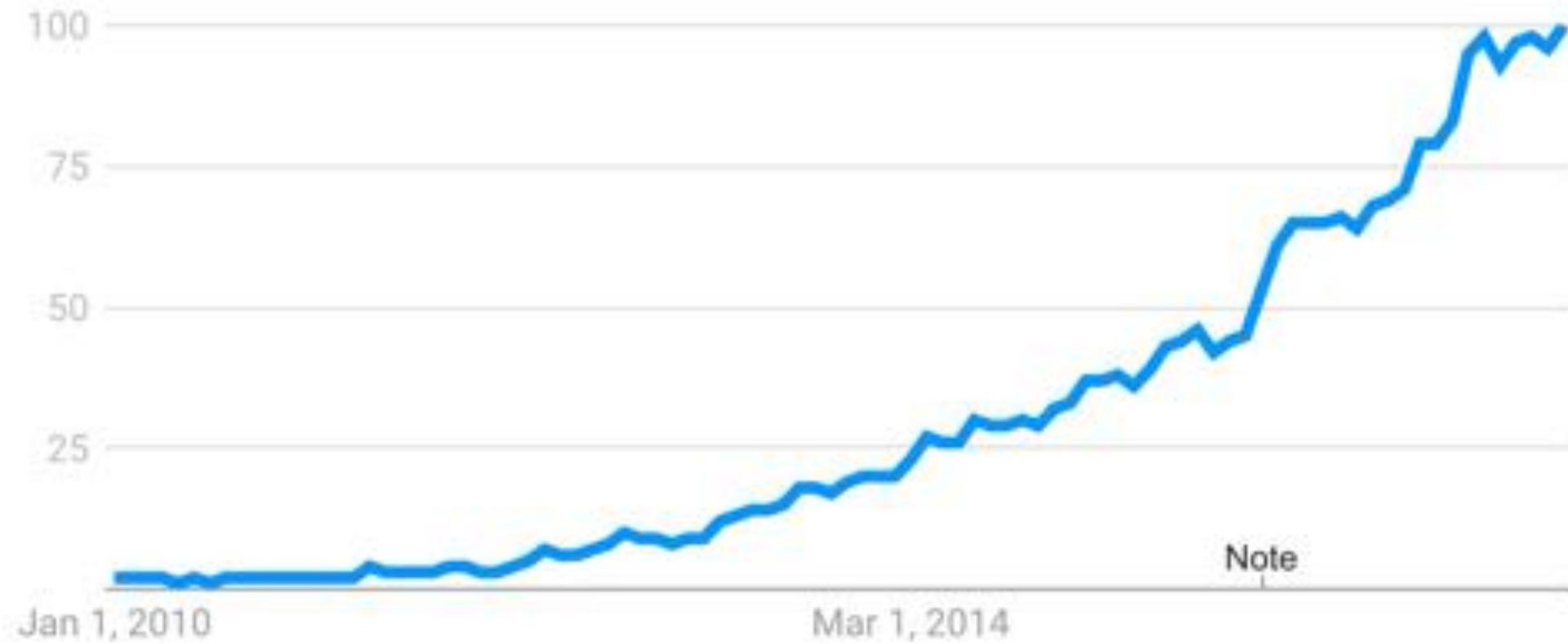


WORLDWIDE
ADOPTION

Interest over time

Google Trends

● golang



Go language soars to new heights in popularity

For the first time, Go makes the top 10 in the Tiobe index of programming language popularity

Go breaks into the Tiobe Top 10

The Go language has broken into the top 10 in the Tiobe index of programming language popularity for the first time.

With an all-time high rating of 2.363 percent, Go ranks as the 10th most popular programming language in this month's index, ahead of languages such as Perl, Swift, Ruby, and Visual Basic. The Tiobe Programming Community index assesses language popularity using a formula based on

MORE LIKE THIS



Kotlin's a rising star in language popularity index



Facebook's PHP dialect makes inroads among popular programming languages



Java and C continue to decline in popularity



VIDEO



15 most popular languages used on GitHub by opened Pull Request and percentage change from previous period



Go #9 on GitHub
usage & 2nd fastest
growing

Language Rank**Types****Spectrum Ranking**

1. Python



100.0

2. C



99.7

3. Java



99.4

4. C++



97.2

5. C#



88.6

6. R



88.1

7. JavaScript



85.5

8. PHP



81.4

9. Go



76.1

10. Swift



75.3

Go #9 on IEEE
rankings

Programming Languages



Go #14 on Stack
Overflow Survey
(usage)

How Many Go Developers Are There?

Posted on Thursday, July 13, 2017.

How many Go developers are there in the world? My best estimate is between half a million and a million.

My approach is to combine

500k - 1M Go Users

of Software Developers × Fraction using Go

the world and what percentage of them are using Go.

(I’m not sure if this is the right number available online, it would seem) estimating that there were 18,539,500 “strict software developers,” giving a total estimate of 18,539,500.

Evans himself has been advertising their “[Global Developer Population and Demographic Study](#)” which estimates the world’s software developers to be 21 million.

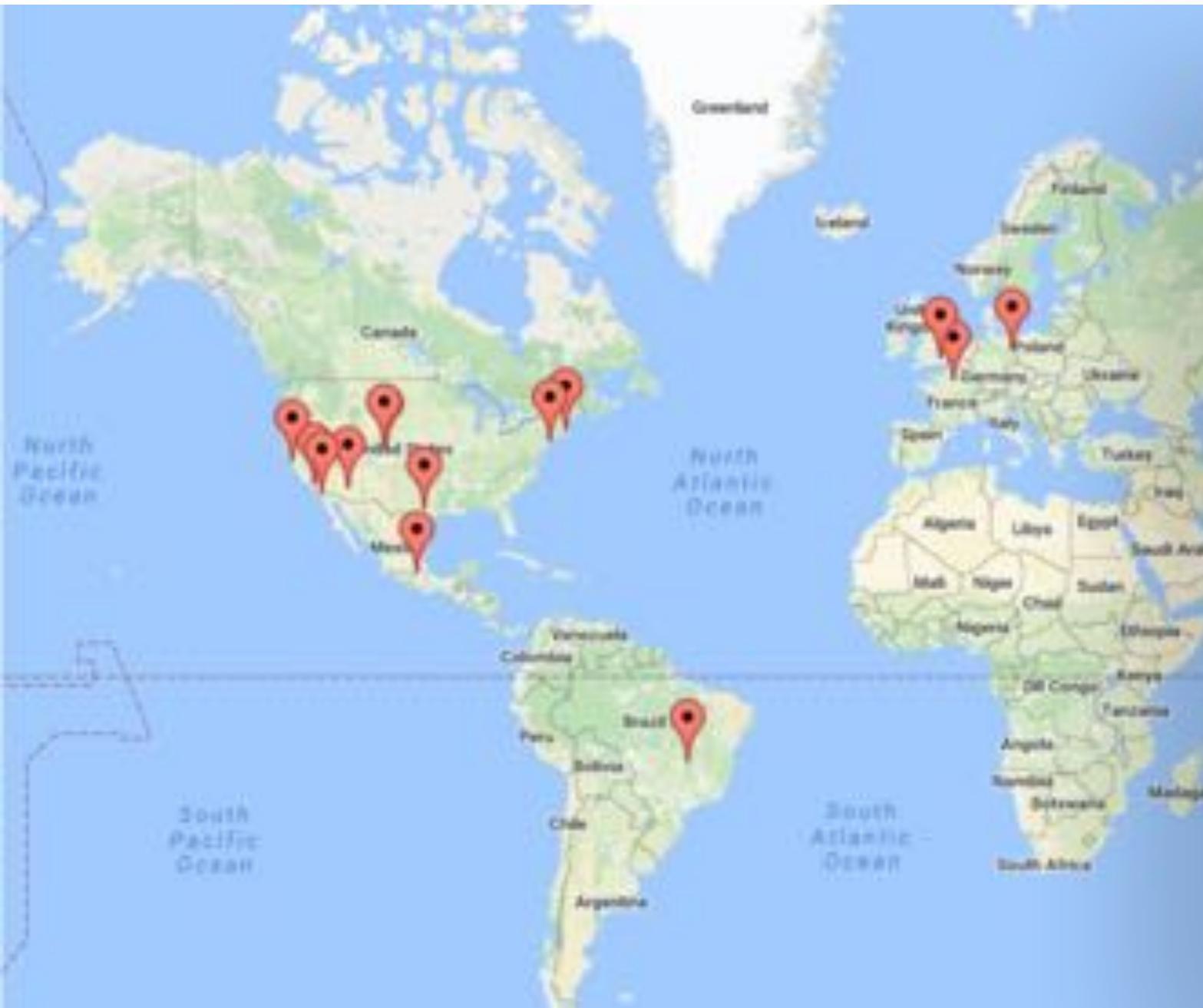
Maybe the Evans estimate is too high. The details of their methodology are key to their business and therefore not revealed publicly, so we can’t easily tell how strict or loose their definition of developer is. In January 2017, PK of the DQYDJ blog posted an analysis titled “[How Many Developers Are There in America, and Where Do They Live?](#).” That post, which includes an admirably detailed methodology section, used data from the 2016 American Census Survey (ACS) and included these employment categories as “strict” software developers:



**WORLDWIDE
COMMUNITY**



100's of Go
meetups all
around the
world



20+
Women Who
Go
chapters
around the
world



Conferences Across the World



GO ADOPTION - BANKING / COMMERCE



Etsy

ticketmaster®



stripe



ebay



NORDSTROM



Booking.com



LUSH
DIGITAL



zalando



Walmart *

skroutz

CRYPTOPAY
PAYMENT GATEWAY



ezbuy®

卡联
CardInfoLink

GO ADOPTION - GAMING / MEDIA



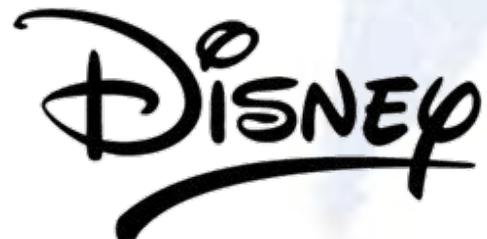
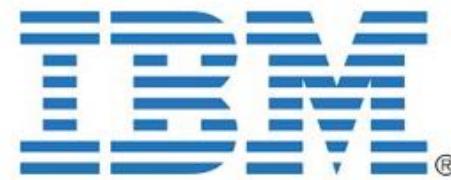
GO ADOPTION - TECH



GO ADOPTION - GENERAL



GO ADOPTION - GLOBAL COMPANIES





**ADDRESSING
OUR
CHALLENGES**

What changes would improve Go most?

Dependency Management

Go's Dependency Management story

isn't good (yet). We realized our deficiencies in this space later than we should have.



Managing Dependencies

- 1 Dependency Management Working Group
- 2 Dep tool as "official experiment"
- 3 Prototype for dependency management integrated with go tool



New User Experience

The Go new user experience is intimidating and needlessly complex. It often takes users 30 - 90 minutes before they are able to use Go.



Improving New User Experience

- 1 Developer Experience Working Group
- 2 User Personas
- 3 One line Installer

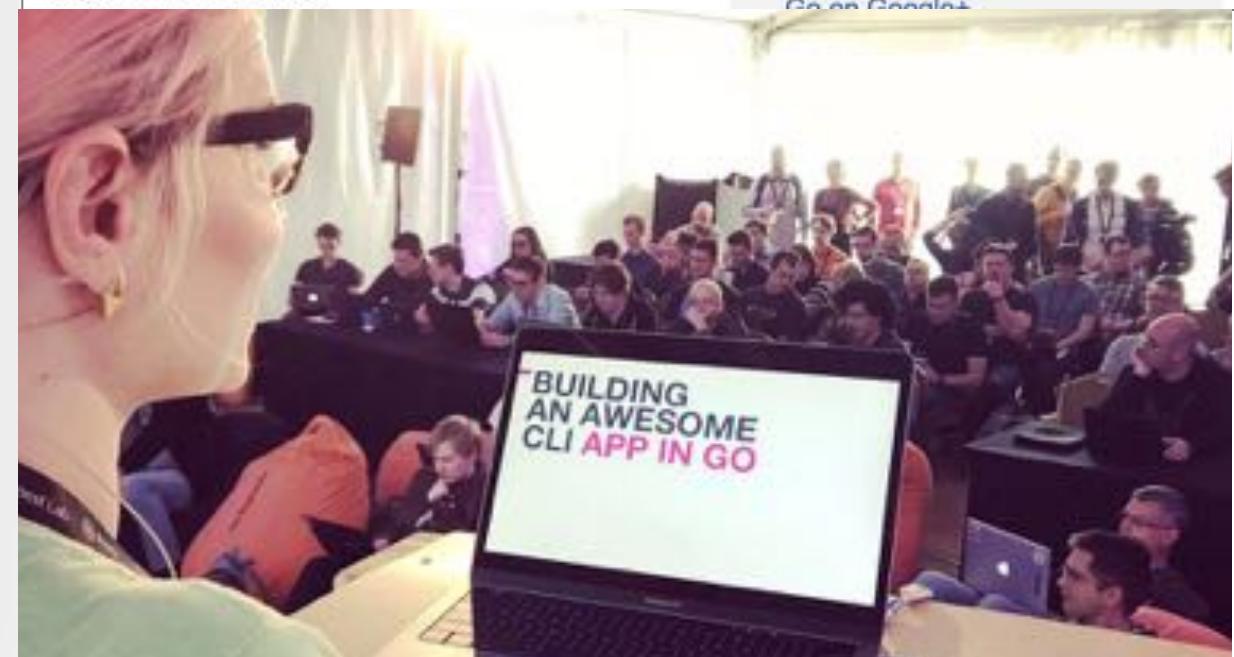
The Go Blog

Introducing the Developer Experience Working Group

10 April 2017

Over the last several years, Go's audience has shifted from early adopters to mainstream users. Today, our users come from a wide variety of backgrounds, experiences, and expectations. The needs of users are growing faster than the Go project can currently address them. To streamline the experience for first-time Go users, we've created the Developer eXperience Working Group (DXWG).

For the next three months, this group will work together on delivering:



[Next article](#)

[Toward Go 2](#)

[Previous article](#)

[HTTP/2 Server Push](#)

[Links](#)

[golang.org](#)

[Install Go](#)

[A Tour of Go](#)

[Go Documentation](#)

[Go Mailing List](#)

[Go on Google+](#)



Community Culture

Like many developer communities Go has had its struggles with diversity, inclusiveness and elitism.

Improving Culture

- 1 Establish, live by, and support the Go Code of Conduct
- 2 Support efforts like Women Who Go and Go Bridge
- 3 Talking about our mistakes





Single Perspective

Go's development has largely been led by Google. Google's engineers' experiences have helped shape Go into the language we love, but has also led to a limited perspective where experiences by others aren't well understood.

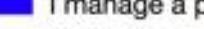
Increasing User Feedback

1 Go User Survey with 3500+ responses

2 Experience Reports

3 Expanding the Go team to include more diverse viewpoints

The following apply to me: (multiple choice)

2,386 (66%)		I program in Go outside of work
2,235 (62%)		I program at work in Go
2,004 (56%)		I program at work in another language
618 (17%)		I manage a programming team
337 (9%)		I am a student
78 (2%)		Other
10 (0%)		I No response

Reading the data: This question was "multiple choice," so the percentages add up to well over 100%. All graphs in this post show both the total count and the corresponding percentage of the 3,595 surveys completed.

I work in the following areas: (multiple choice)

2,272 (63%)		Web development
-------------	---	-----------------

Experience Reports

Justin Clift edited this page 11 hours ago · 64 revisions

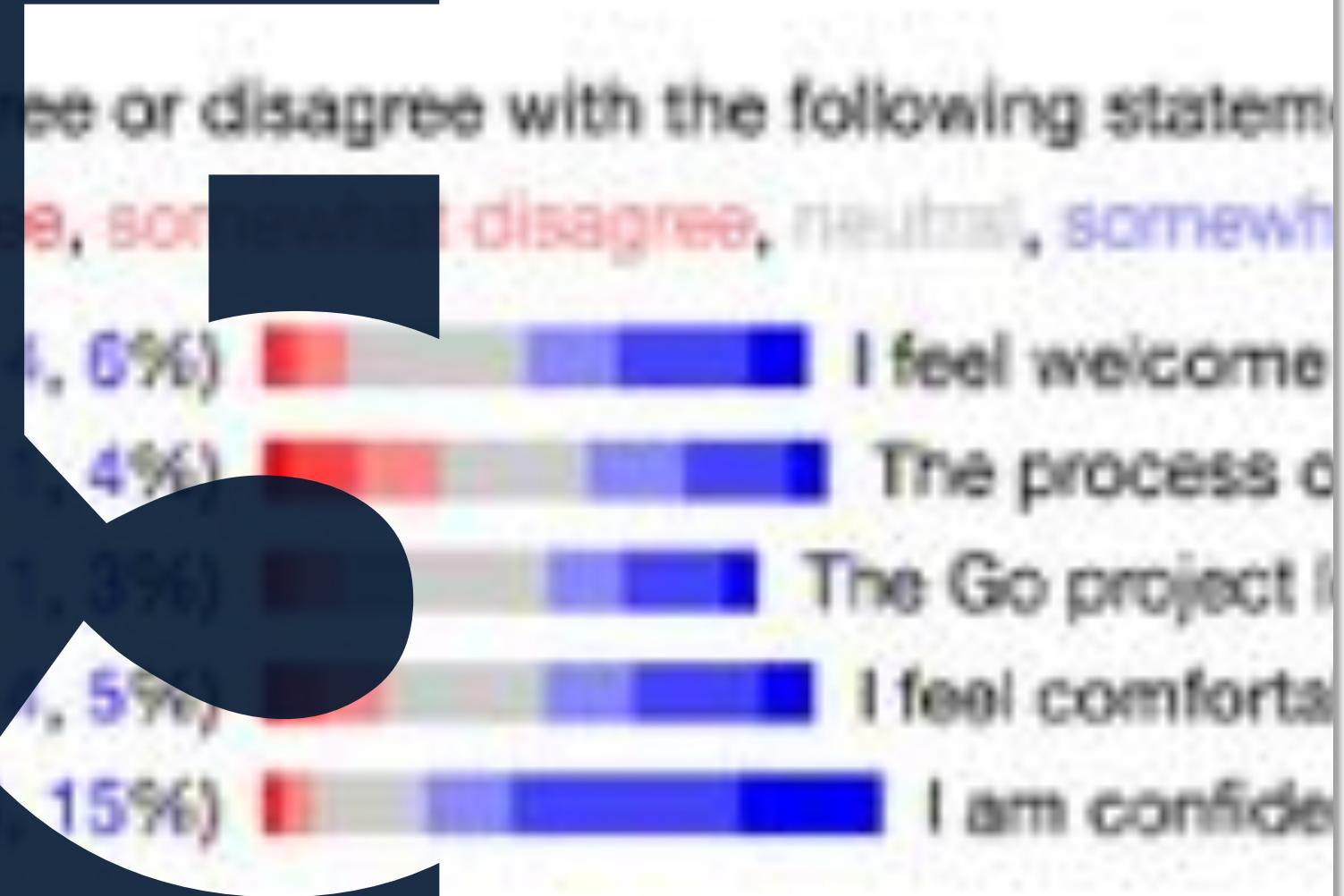
This page collects experience reports about problems with Go that might inform our design of solutions to those problems. These reports should focus on the *problems*: they should not focus on and need not propose solutions. To propose solutions, see the [proposal process](#).

We hope to use these experience reports to understand where people are having trouble writing Go, to help us prioritize future changes to the Go ecosystem. (We do not promise to reply to these. If you need immediate help answering questions about Go, see <https://golang.org/help/> for resources.)

The best experience reports tell: (1) what you wanted to do, (2) what you actually did, and (3) why that wasn't great, illustrating those by real concrete examples, ideally from production use. Please write these reports about the problems most significant to you, post them on your own blog, or on Medium, or as a [Github Gist](#) (use a `.md` extension for

Project Participation

It's too hard and too intimidating for many in the community to contribute to the Go project.



Expanding Go Team & Contributors

- 1 Formation of Working Groups to extend leadership
- 2 Contribution Workshop
- 3 Contributor Summit





**MAKING
AN
IMPACT**

“

In roughly a week's time,
I went from initial
commit to shipping
replacement backends.

- Matt Robenolt @

DISQUS

“

You can write in Go as
easily as Python, but it
can save you a lot of
machine resources.

- Alexander Ponomarev @



(translated from Russian)

GO @ DROPBOX

- MOST OF DROPBOX INFRASTRUCTURE IS WRITTEN IN GO
- WE HAVE OVER 150 GO CONTRIBUTORS @ DROPBOX TO OUR SERVER REPO
- OVER 1.3 MILLION LINES OF GO @ DROPBOX



A wide-angle photograph of a tropical sunset. The sky is filled with soft, pastel-colored clouds transitioning from pink to orange and yellow. In the foreground, dark silhouettes of palm trees frame the scene. The ocean waves are visible, reflecting the warm colors of the sunset. A white rectangular box with a thin black border is centered in the middle of the image, containing the text.

WHERE WE'RE GOING



GO
- THE
LANGUAGE
OF THE CLOUD

THE CLOUD IS BUILT IN GO



"We've had great experiences with tools written in Go and now we are adopting it for our next project

- Cloud Users in 2018



GO 2

“

First and foremost, Go 2's main goal will be to fix the most significant ways Go 1.x fails at scale.

- Russ Cox

" Maybe we can do
two or three
[changes], certainly
not more than five.

- Russ Cox

Experience Reports

We need your help!

We need people to write experience reports to clearly define and communicate real issues they are experiencing using Go.

ExperienceReports

Justin Clift edited this page 11 hours ago · 64 revisions

This page collects experience reports about problems with Go that might inform our design of solutions to those problems. These reports should focus on the problems: they should not focus on and need not propose solutions. To propose solutions, see the [proposal process](#).

We hope to use these experience reports to understand where people are having trouble writing Go, to help us prioritize future changes to the Go ecosystem. (We do not promise to reply to these. If you need immediate help answering questions about Go, see <https://golang.org/help/> for resources.)

The best experience reports tell: (1) what you wanted to do, (2) what you actually did, and (3) why that wasn't great, illustrating those by real concrete examples, ideally from production use. Please write these reports about the problems most significant to you, post them on your own blog, or on Medium, or as a [Github Gist](#) (use a `.md` extension for

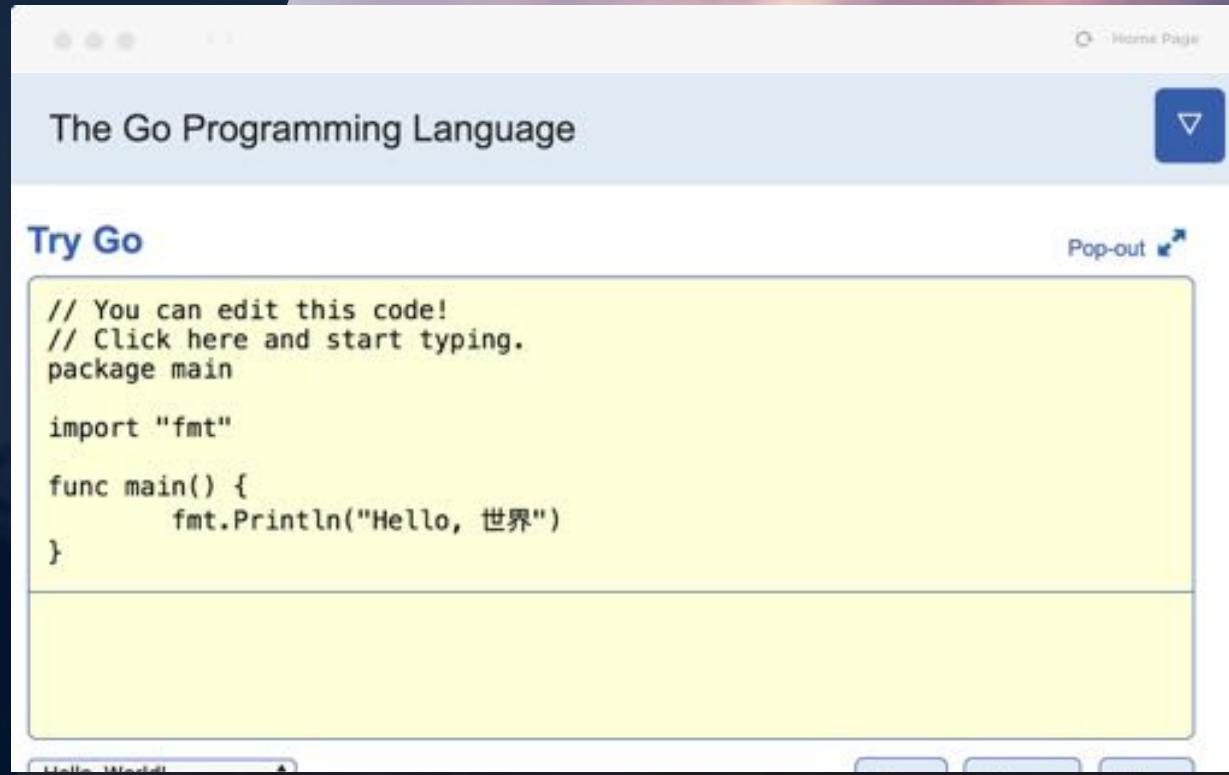


NEW
CHALLENGES
OPPORTUNITIES

Go's Messaging Problem

Go's value isn't in one key feature. People that use Go love it but struggle to communicate why to others

"Try it and you'll love it" has limited success in communicating the value Go brings to organizations



The screenshot shows a web browser window displaying "The Go Programming Language". A modal dialog box titled "Try Go" is open, containing a code editor with the following Go code:

```
// You can edit this code!
// Click here and start typing.
package main

import "fmt"

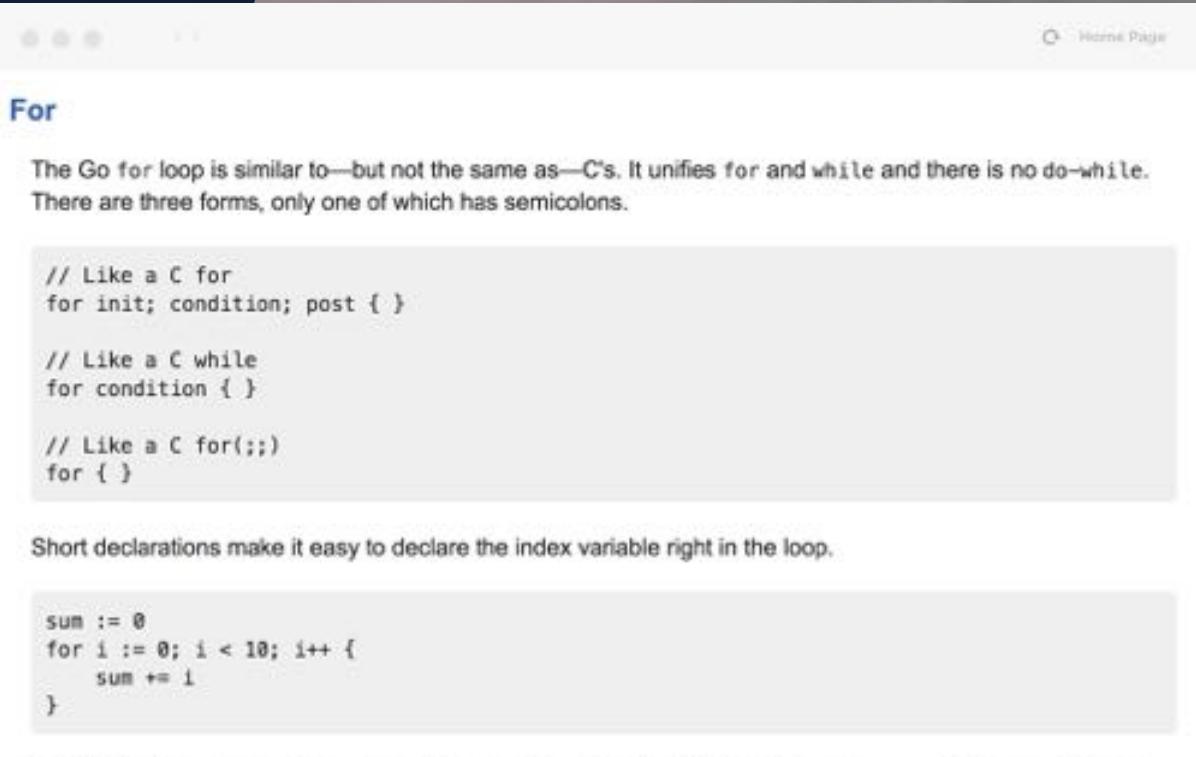
func main() {
    fmt.Println("Hello, 世界")
}
```

Below the code editor, a small preview window shows the output "Hello, 世界". The browser interface includes a "Home Page" button, a dropdown menu, and a "Pop-out" button.

Go's Accessibility Issue

Go's docs have an implicit audience of systems programmers having a formal CS education.

The work to define personas is the first of many steps to meeting Go's audience where they are.



The screenshot shows a section of the Go documentation titled "For". It explains that the Go for loop is similar to—but not the same as—C's. It unifies for and while and there is no do-while. There are three forms, only one of which has semicolons.

```
// Like a C for
for init; condition; post {}

// Like a C while
for condition {}

// Like a C for(;;)
for {}  
Short declarations make it easy to declare the index variable right in the loop.
```

```
sum := 0
for i := 0; i < 10; i++ {
    sum += i
}
```

WE NEED YOUR HELP



The Go project will never reach its potential
without contributions from people like you

