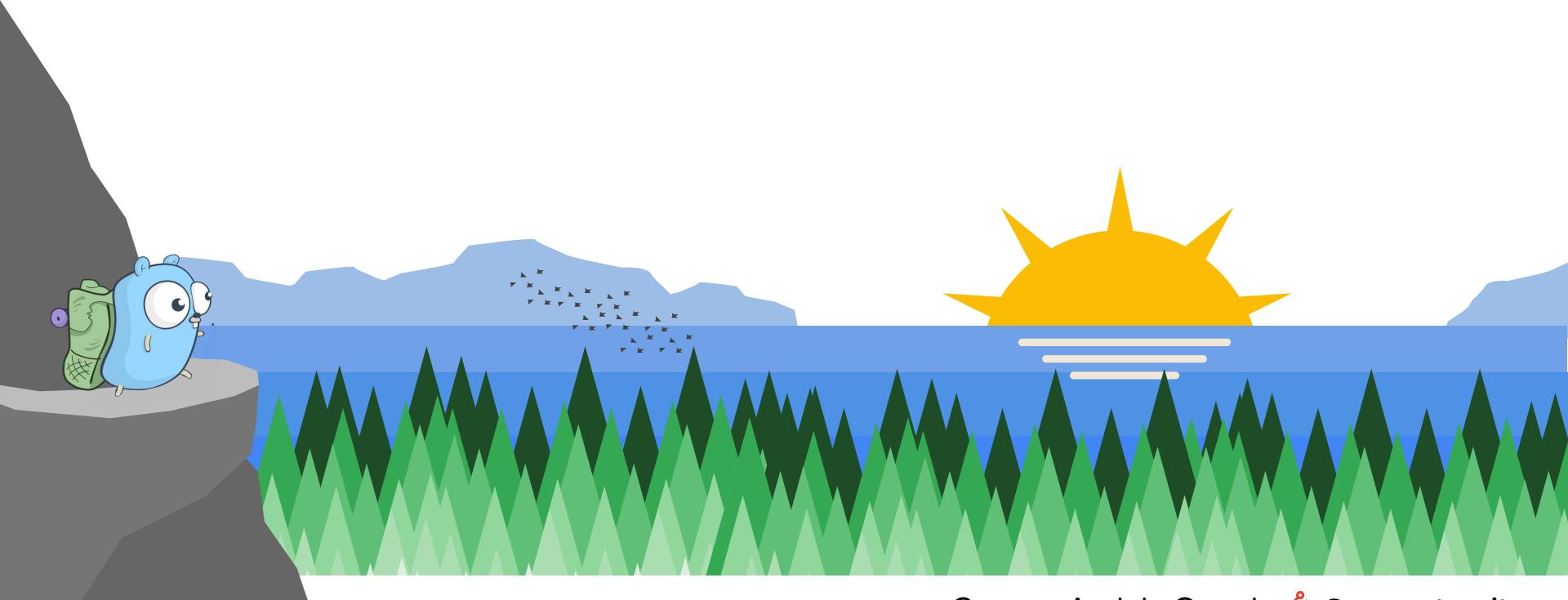


# Crossing the Chasm: Go for the Next Million Users



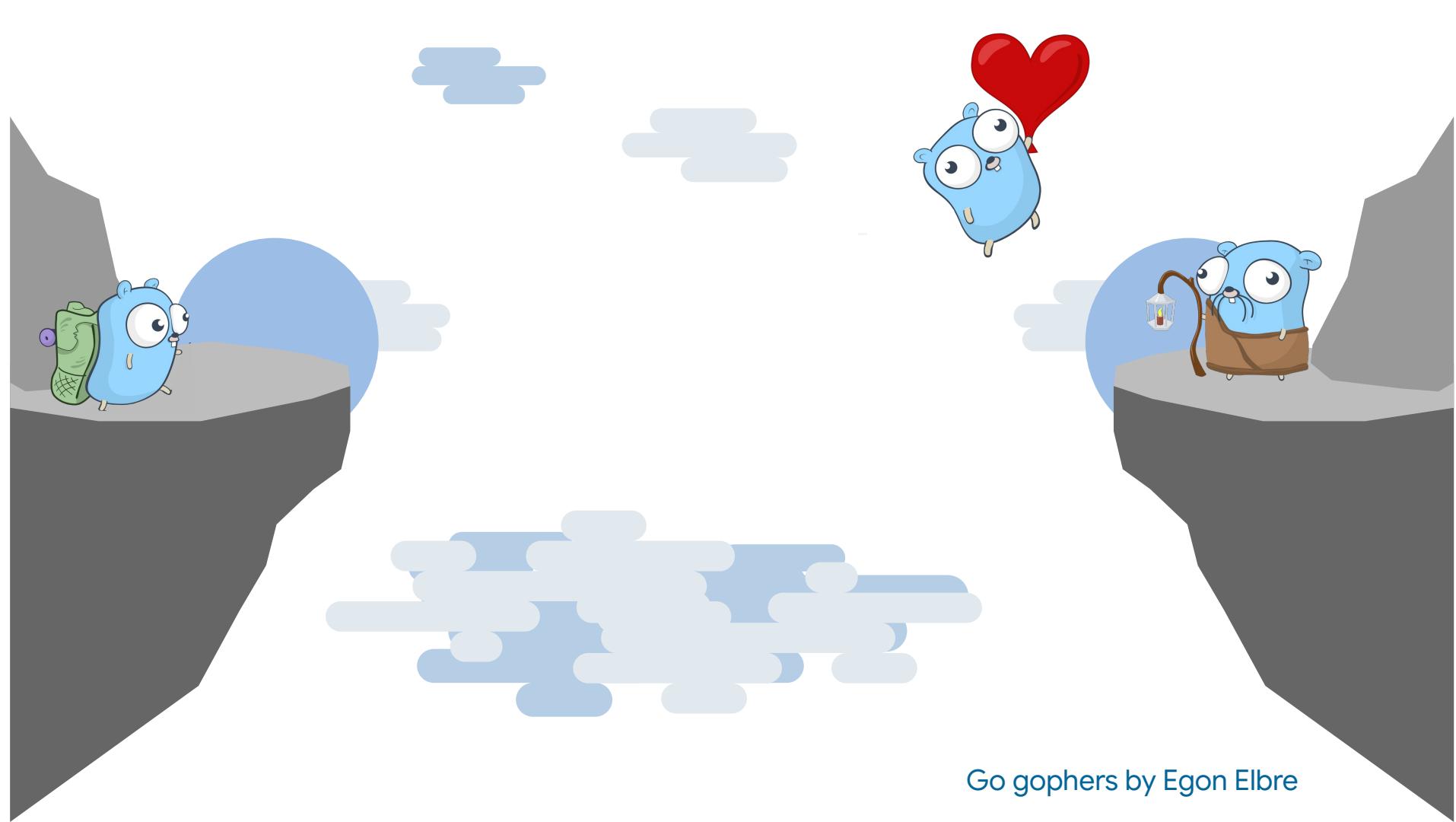
Carmen Andoh, Google  @carmatrocit

# Go's 2020 Global Developer Population ~ 2 Million



# Future of Go's Ecosystem?



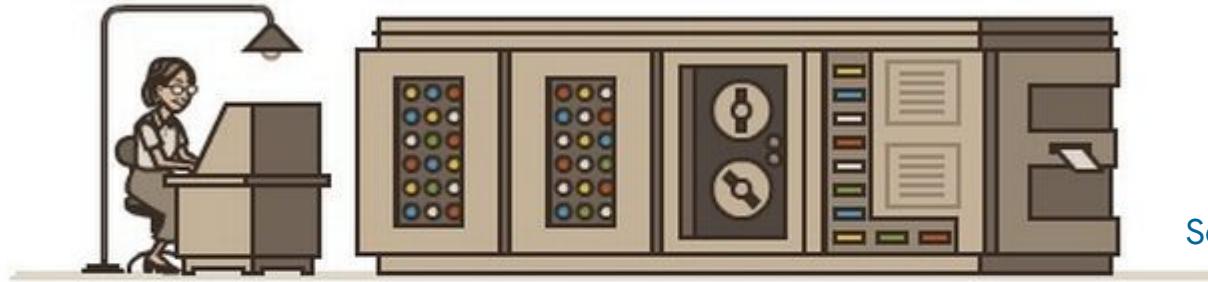


Go gophers by Egon Elbre

A programmer is asked which language they think people will be using in 30 years.

A programmer is asked which language they think people will be using in 30 years.

**“I don’t know, but it’ll be called COBOL”**



Source: Google Doodle, Grace Hopper

# Factors in Choosing A Programming Language: Ideals v Reality

"All the News  
That's Fit to Print"

# The New York Times

VOL. CLXIX .... No. 58,688      SATURDAY, MAY 9, 2020

## U.S. UNEMPLOYMENT IS WORST SINCE DEPRESSION

MONTHLY CHANGE IN JOBS SINCE THE END OF WORLD WAR II

1946 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 +1 million

-25.7

**The New York Times**  
VOL. CLXIX ... No. 58,688      SATURDAY, MAY 9, 2020

### U.S. UNEMPLOYMENT IS WORST SINCE DEPRESSION

"All the News  
That's Fit to Print"

U.S. UNEMPLOYMENT IS WORST SINCE DEPRESSION

NATIONAL UNEMPLOYMENT RATE

APRIL UNEMPLOYMENT BY STATE

Georgia Killing Puts Spotlight on a Police Force's Troubled History

If Fyffe Case, Russia Inquiry Still Isn't Safe, Is Barr's Office?

As Official Toll Ignores Reality, Mexico's Hospitals Are Overrun

New York Returns to the Armory

INTERNATIONAL

OPPORTUNITY ROAD (1)

A Good Walk, Unspelled

Opposite: A woman walks along a path in a park in New York City. Above left, a man walks his dog in a park in New York City. Above right, a woman walks her dog in a park in New York City.

INTERVIEW WITH

Barber Shakes Up Suriname

Opposite: A woman walks along a path in a park in New York City. Above left, a man walks his dog in a park in New York City. Above right, a woman walks her dog in a park in New York City.

Moors and Pussies

Opposite: A woman walks along a path in a park in New York City. Above left, a man walks his dog in a park in New York City. Above right, a woman walks her dog in a park in New York City.

THE TIMES

20,000,000

# Factors in Choosing A Programming Language: Ideals v Reality



## Wanted urgently: People who know a half century-old computer language so states can process unemployment claims

By Alicia Lee, CNN

Updated 4:00 PM ET, Wed April 8, 2020



BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

COMMON BUSINESS-ORIENTED LANGUAGE —

## IBM scrambles to find or train more COBOL programmers to help states

Some states rely on COBOL software to manage unemployment systems.

TIMOTHY B. LEE - 4/13/2020, 1:05 PM

# Factors in Choosing A Programming Language: Ideals v Reality

“The use of COBOL  
cripples the mind; its  
teaching should,  
therefore, be regarded  
as a criminal offense.”

— Edsger W. Dijkstra



## **Wanted urgently: People who know a half century-old computer language so states can process unemployment claims**

By Alicia Lee, CNN

Updated 4:00 PM ET, Wed April 8, 2020



BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

COMMON BUSINESS-ORIENTED LANGUAGE —

## **IBM scrambles to find or train more COBOL programmers to help states**

Some states rely on COBOL software to manage unemployment systems.

TIMOTHY B. LEE - 4/13/2020, 1:05 PM

# Factors in Choosing a Programming Language - **Reality**

“Most languages don’t die—or at least once they get to a certain level of acceptance they don’t die.”

- Brian Kernighan

## Factors in Choosing a Programming Language - **Reality**

“Most languages don’t die—or at least once they get to a certain level of acceptance they don’t die.”

- Brian Kernighan

“Any language with legacy code will last forever. Rewrites are expensive. If it’s not broke don’t fix it.”

- Ari Rabkin

## Factors in Choosing a Programming Language - **Reality**

“Most languages don’t die—or at least once they get to a certain level of acceptance they don’t die.”

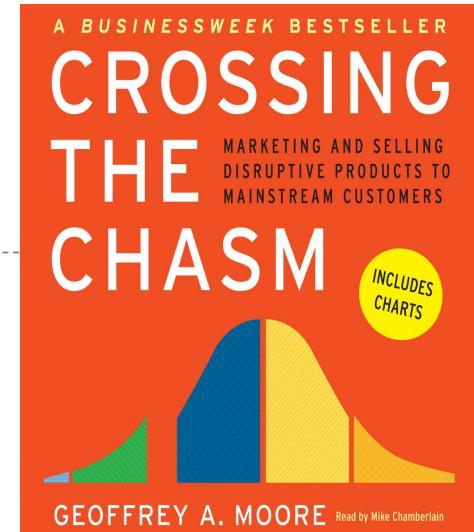
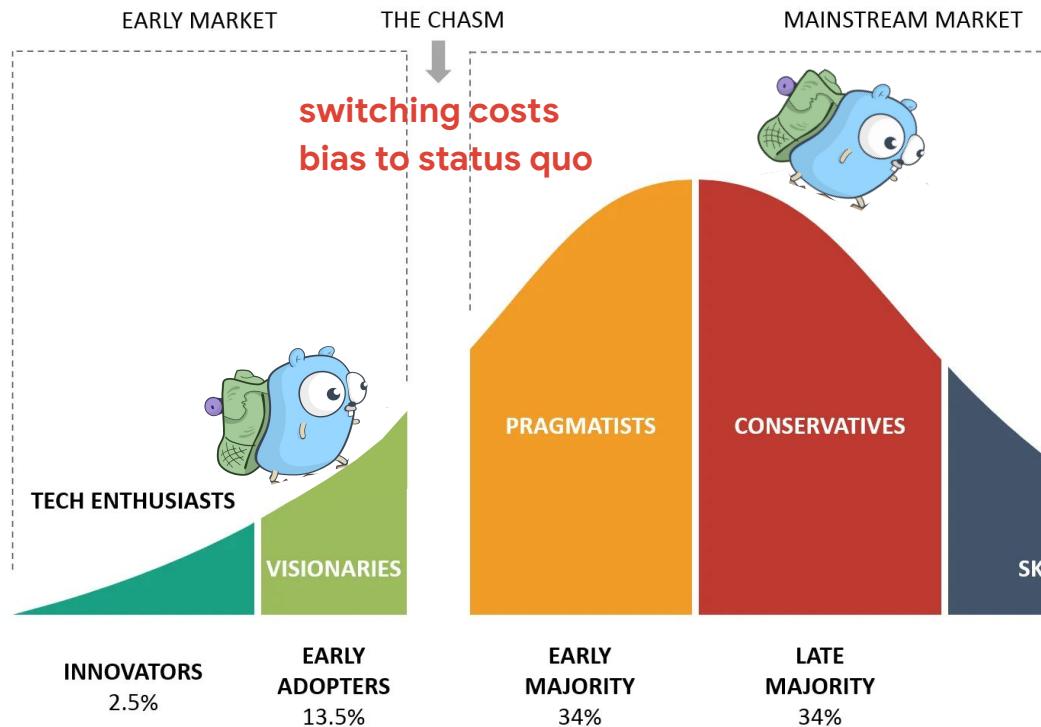
- Brian Kernighan

“Any language with legacy code will last forever. Rewrites are expensive. If it’s not broke don’t fix it.”

- Ari Rabkin

This is what is called the “**switching cost**”

# Crossing the Chasm - Geoffrey Moore 1991



# DIFFUSION OF INNOVATIONS

FIFTH EDITION



EVERETT M. ROGERS

# DIFFUSION OF INNOVATIONS

FIFTH EDITION



EVERETT M. ROGERS

1

**How, why, and at what rate new ideas and technology spread.**

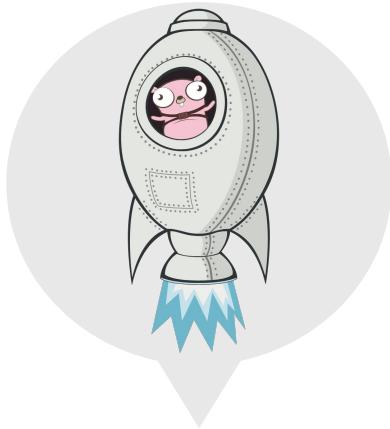
2

**Diffusion is the process by which an innovation is communicated over time among the participants of a social system.**

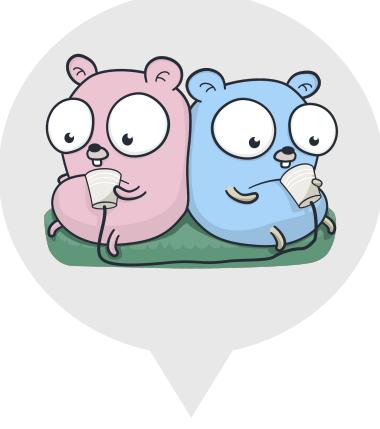
3

**Origins of the Diffusions of Innovations theory are varied and span multiple disciplines**

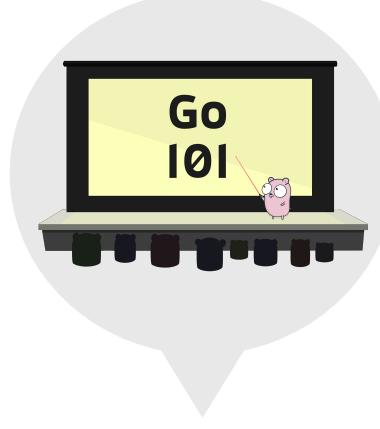
# Four Main Elements That Influence Spread of a New Idea



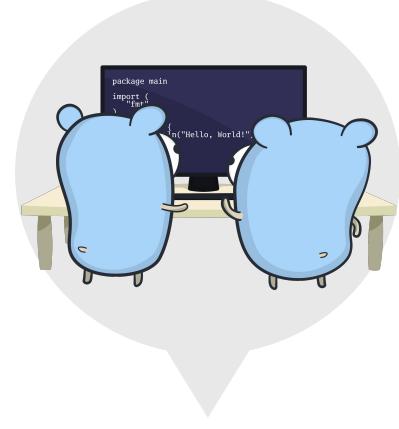
INNOVATION



COMMUNICATION CHANNELS



SOCIAL SYSTEM



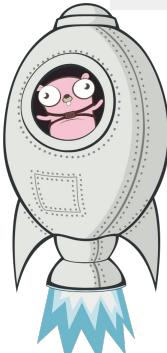
TIME

# Four Main Elements That Influence Spread of a New Idea

INNOVATION



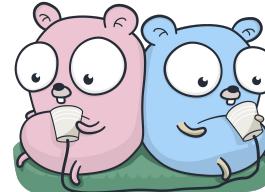
Programming Language



COMMUNICATION CHANNELS



User manuals, tutorials (training, videos, articles, etc), delivery mode (web, print, in person, social)

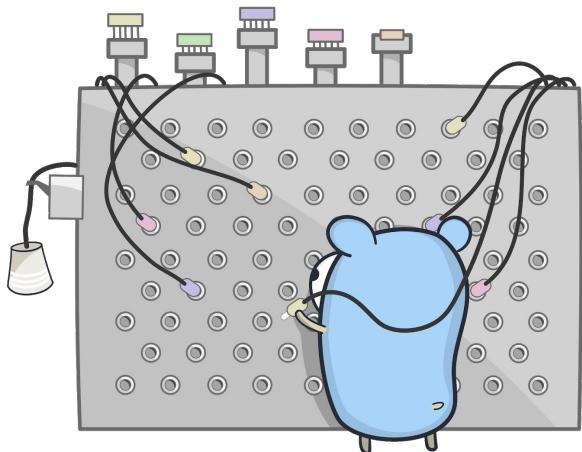


SOCIAL SYSTEM



Social and Operating Contexts of Varied Audiences

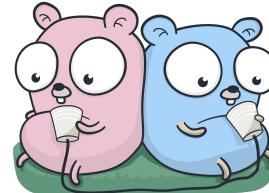
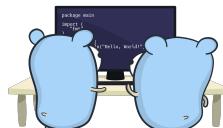
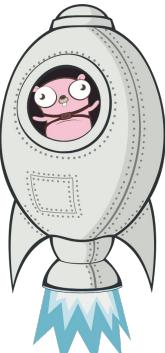
TIME



# Four Main Elements That Influence Spread of a New Idea

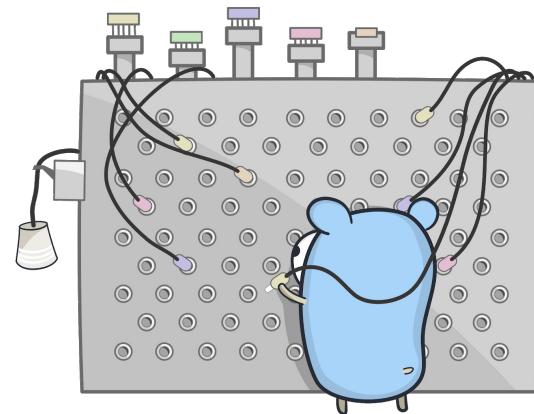
## COMMUNICATION CHANNELS

The **communication structure** determines how messages may flow through the social system, e.g. by providing communication links between individuals.

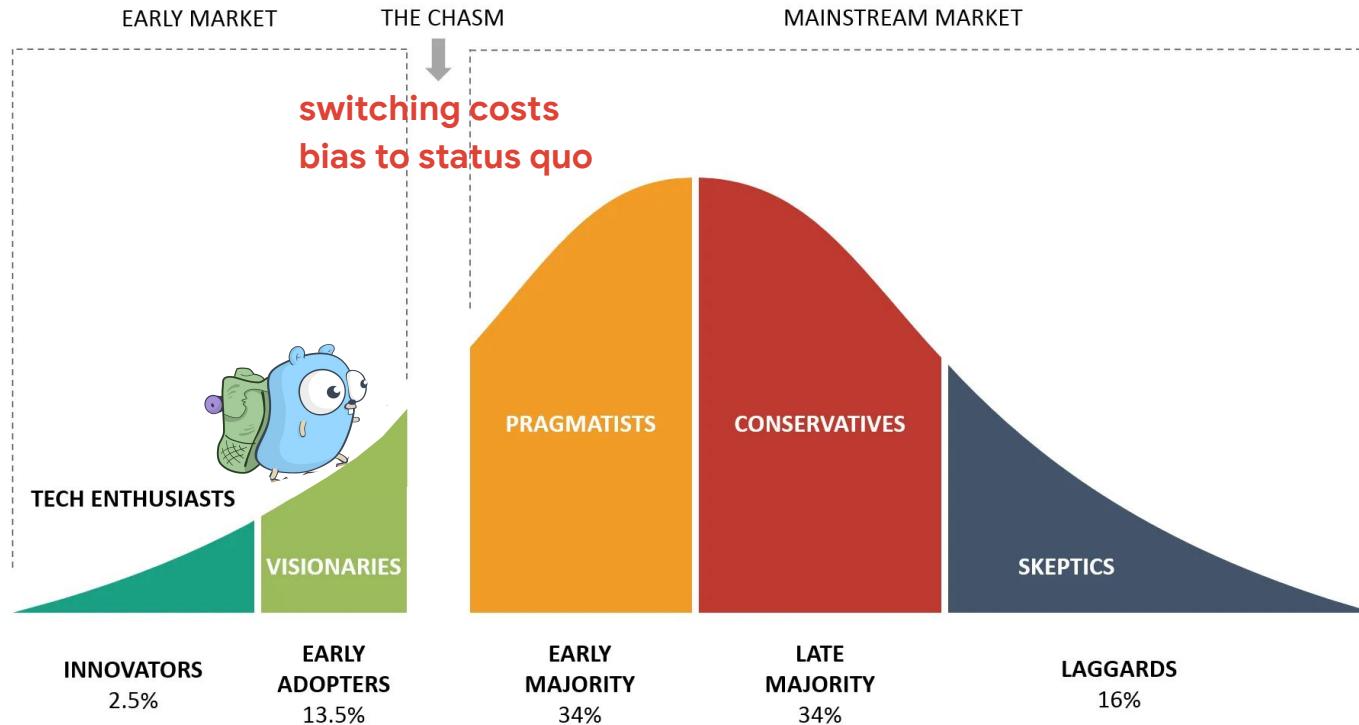


## SOCIAL SYSTEM

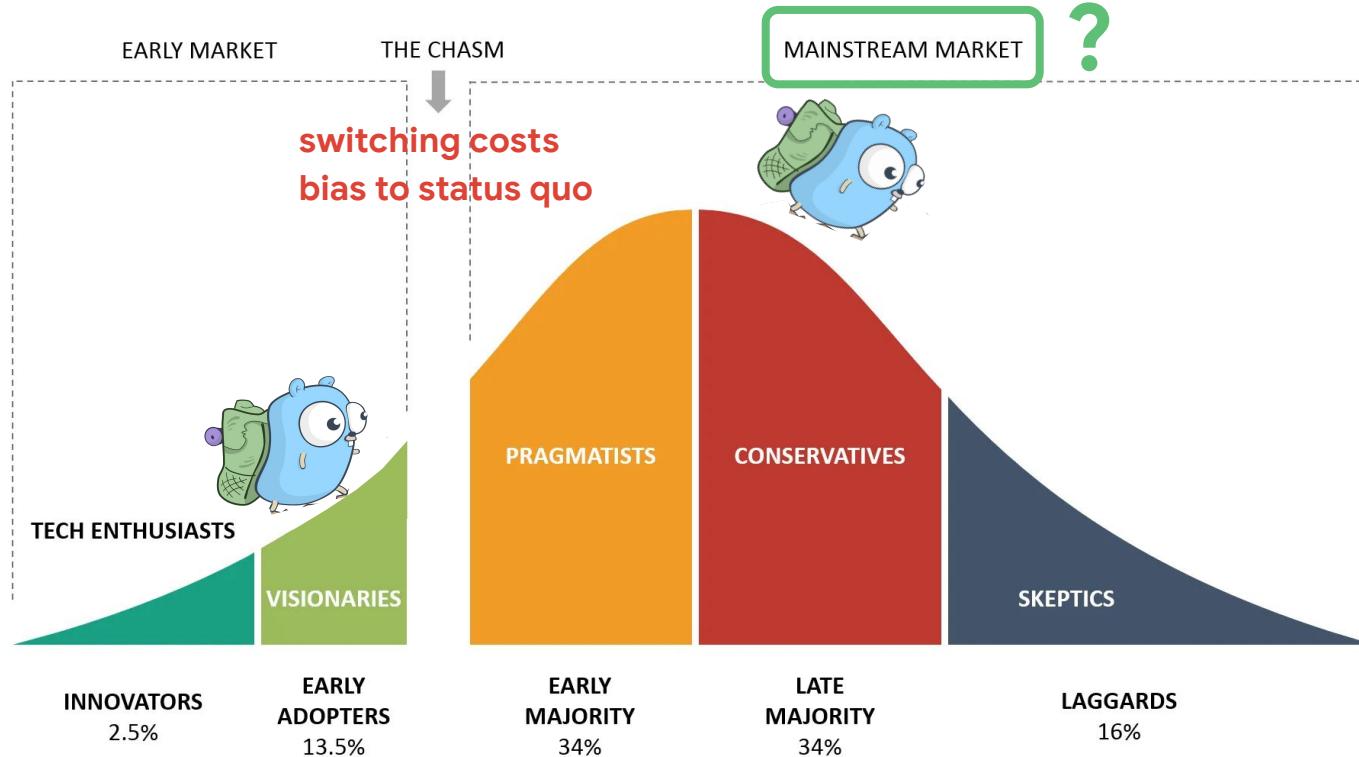
The **social structure** influences diffusion through values, norms, roles, and hierarchies.



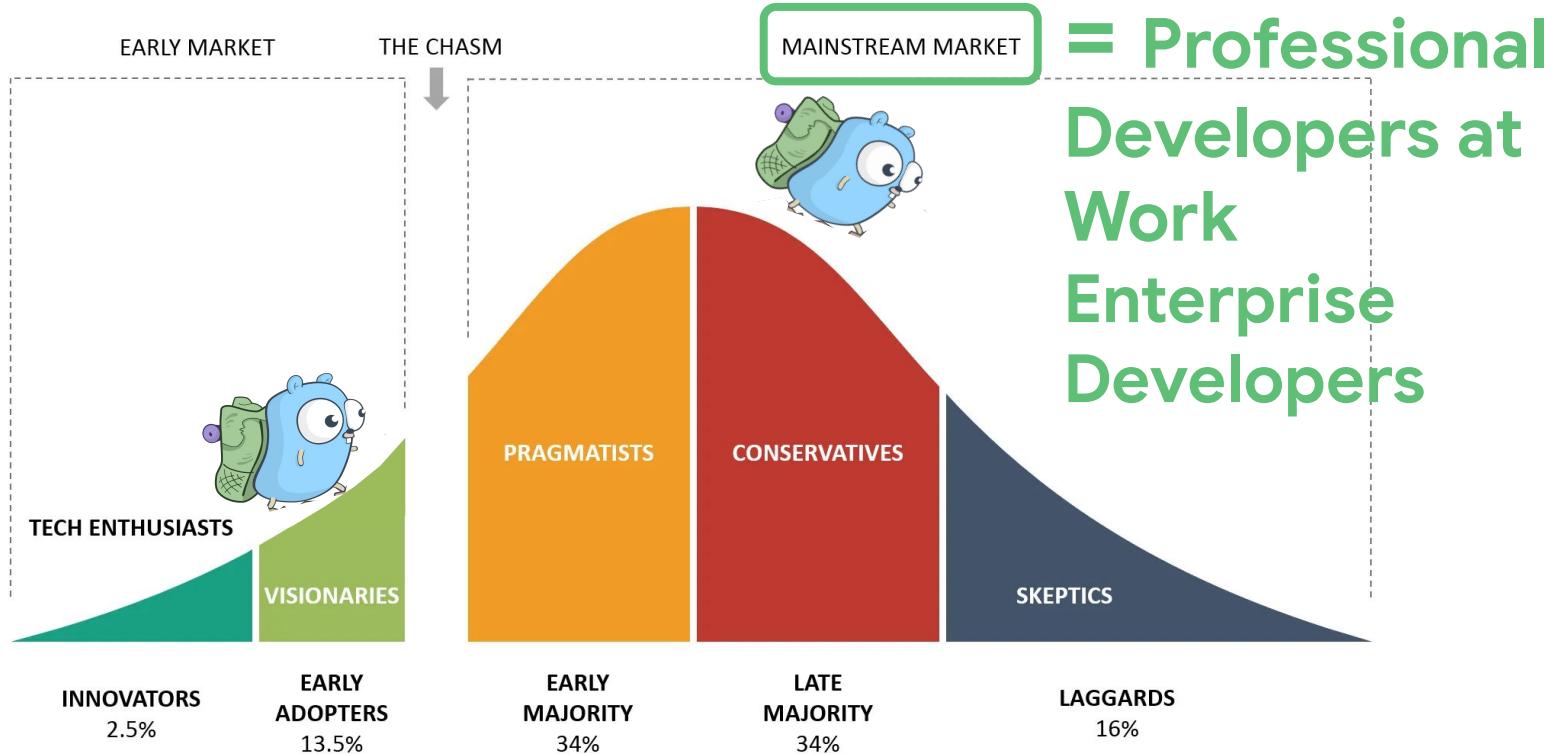
# Crossing the Chasm - Go Programming Language



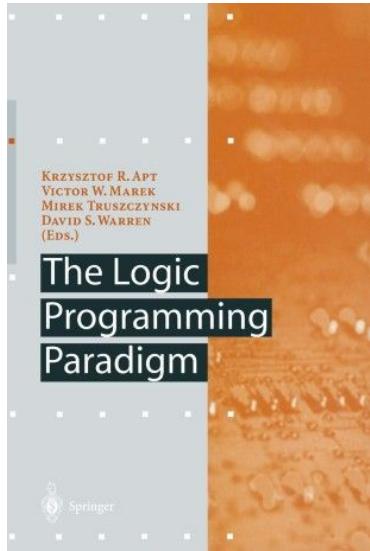
# Crossing the Chasm - Go Programming Language



# Crossing the Chasm - Go Programming Language



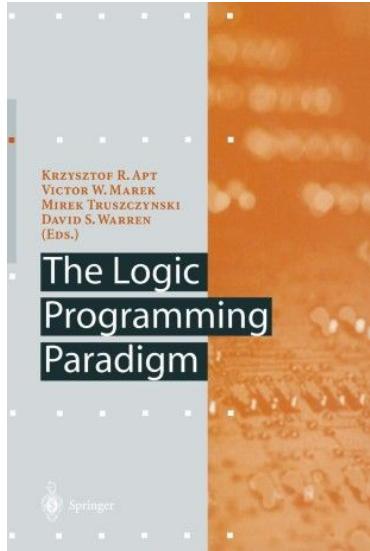
# Four Main Elements That Influence Spread of a New Idea - The Case of Haskell



How enterprises use functional languages, and  
why they don't

Philip Wadler  
Bell Labs, Lucent Technologies  
[wadler@research.bell-labs.com](mailto:wadler@research.bell-labs.com)

# Four Main Elements That Influence Spread of a New Idea - The Case of Haskell



How enterprises use functional languages, and  
why they don't

Philip Wadler  
Bell Labs, Lucent Technologies  
[wadler@research.bell-labs.com](mailto:wadler@research.bell-labs.com)

“Have you used it in  
anger yet?”

# Four Main Elements That Influence Spread of a New Idea

## The Case of Haskell

“The time is a dozen years ago... my fellow postdoc has just scrutinized my new bike. He’s admired the chrome, checked the gears, noted the Kryptonite lock. Now he wants to know **if I’ve used it to serious purpose**. Gleaming chrome is well and good, **but will it run you through the woods?**

**Have you used it in anger yet?**”

-Philip Wadler

# Four Main Elements That Influence Spread of a New Idea

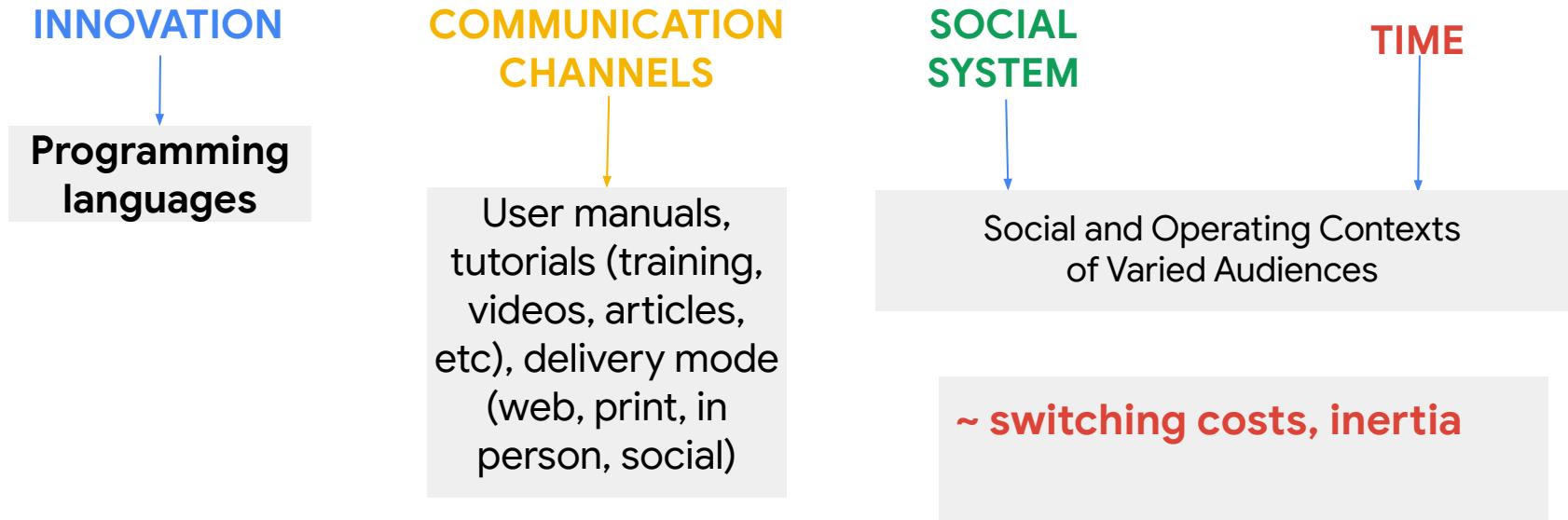
## The Case of Haskell

“Mathematical elegance is well and good, **but will it run that mission-critical system?**”

**COBOL does.**

**And so does Go.**

# Four Main Elements That Influence the Spread of New Ideas



Philip Wadler's "Angry Half Dozen" things for a programming language to be "Enterprise Ready"

# Four Main Elements That Influence the Spread of New Ideas

## INNOVATION

Programming languages

- ~interoperability
- ~extensive libraries
- ~highly portable
- ~stable, easy to install
- ~profilers & debuggers

## COMMUNICATION CHANNELS

User manuals, tutorials (training, videos, articles, etc), delivery mode (web, print, in person, social)

## SOCIAL SYSTEM

Social and Operating Contexts of Varied Audiences

## TIME

~ switching costs, inertia

# Four Main Elements That Influence the Spread of New Ideas

## INNOVATION

Programming languages

- ~interoperability
- ~extensive libraries
- ~highly portable
- ~stable, easy to install
- ~profilers & debuggers

## COMMUNICATION CHANNELS

User manuals, tutorials (training, videos, articles, etc), delivery mode (web, print, in person, social)

Be Accompanied by Training Courses

## SOCIAL SYSTEM

Social and Operating Contexts of Varied Audiences

## TIME

~ switching costs, inertia

# Four Main Elements That Influence the Spread of New Ideas

## INNOVATION

### Programming languages

- ~interoperability
- ~extensive libraries
- ~highly portable
- ~stable, easy to install
- ~profilers & debuggers

## COMMUNICATION CHANNELS

User manuals, tutorials (training, videos, articles, etc), delivery mode (web, print, in person, social)

Be Accompanied by Training Courses

## SOCIAL SYSTEM

Social and Operating Contexts of Varied Audiences

## TIME

~ switching costs, inertia

~ good track record on previous projects

# Four Main Elements That Influence the Spread of New and Persistence of Existing Ideas

INNOVATION

Programming languages

- ~Support interlanguage working.
- ~Possess extensive libraries
- ~Highly Portable
- ~Stable and Easy to Install
- ~Come with Profilers and Debuggers

COMMUNICATION CHANNELS

User manuals, tutorials (training, videos, articles, etc), delivery mode (web, print, in person, social)

Be Accompanied by Training Courses

SOCIAL SYSTEM

Killer App

TIME

~ good track record on previous projects



# The Killer App for Programming Languages

Python — Zope, Django

Rust — Servo

Go — Docker, Kubernetes

Kotlin — Android compatibility

# The Killer App for Programming Languages

Python — Zope, Django

Rust — Servo

Go — Docker, Kubernetes

Kotlin — Android compatibility



Killer App = *jumpstart proof of viability* for the given domain.

# Four Main Elements That Influence the Spread of New and Persistence of Existing Ideas

## INNOVATION

### Programming languages

- ~interoperability
- ~extensive libraries
- ~highly portable
- ~stable, easy to install
- ~profilers & debuggers

## COMMUNICATION CHANNELS

User manuals, tutorials (training, videos, articles, etc), delivery mode (web, print, in person, social)

Be Accompanied by Training Courses

## SOCIAL SYSTEM

Social and Operating Contexts of Varied Audiences

## TIME

~ switching costs, inertia

~ good track record on previous projects

~ killer app, social proof

~ “looking right and left”

# Empirical Analysis of Programming Language Adoption

Leo A. Meyerovich

UC Berkeley \*

lmeyerov@eecs.berkeley.edu

Ariel Rabkin

Princeton University

asrabkin@cs.princeton.edu

## Abstract

Some programming languages become widely popular while others fail to grow beyond their niche or disappear altogether. This paper uses survey methodology to identify the factors that lead to language adoption. We analyze large datasets, including over 200,000 SourceForge projects, 590,000 projects tracked by Ohloh, and multiple surveys of 1,000-13,000 programmers.

aid developers in determining when and whether to bet on a new, experimental language. To date, the language adoption process has not been quantitatively studied in a large scale. This paper addresses that gap. We use a combination of survey research and software repository mining to investigate the factors that influence developer language choice.

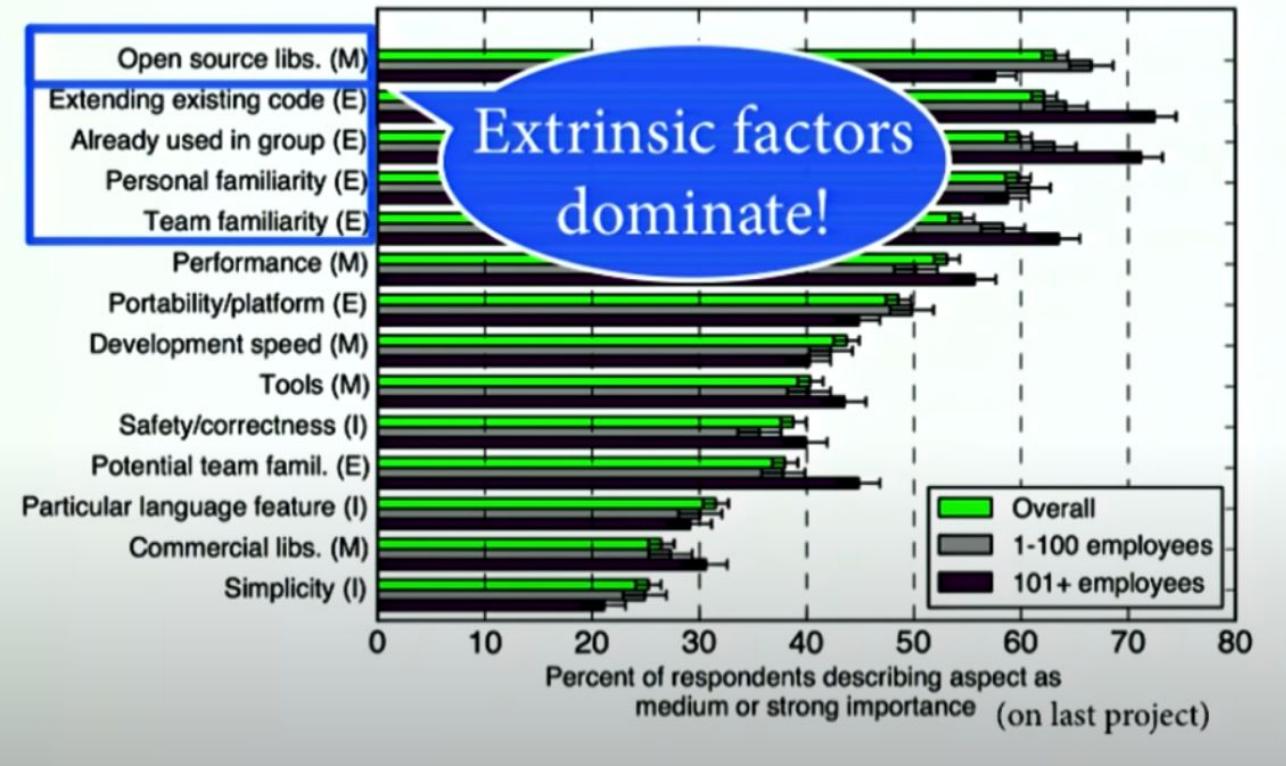
Since little is quantified about the programming language adoption process, we focus on broad research questions:

**Authors:**  [Leo A. Meyerovich](#),  [Ariel S. Rabkin](#) [Authors Info & Affiliations](#)

**Publication:** OOPSLA '13: Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applications • October 2013 • Pages 1–18 • <https://doi.org/10.1145/2509136.2509515>

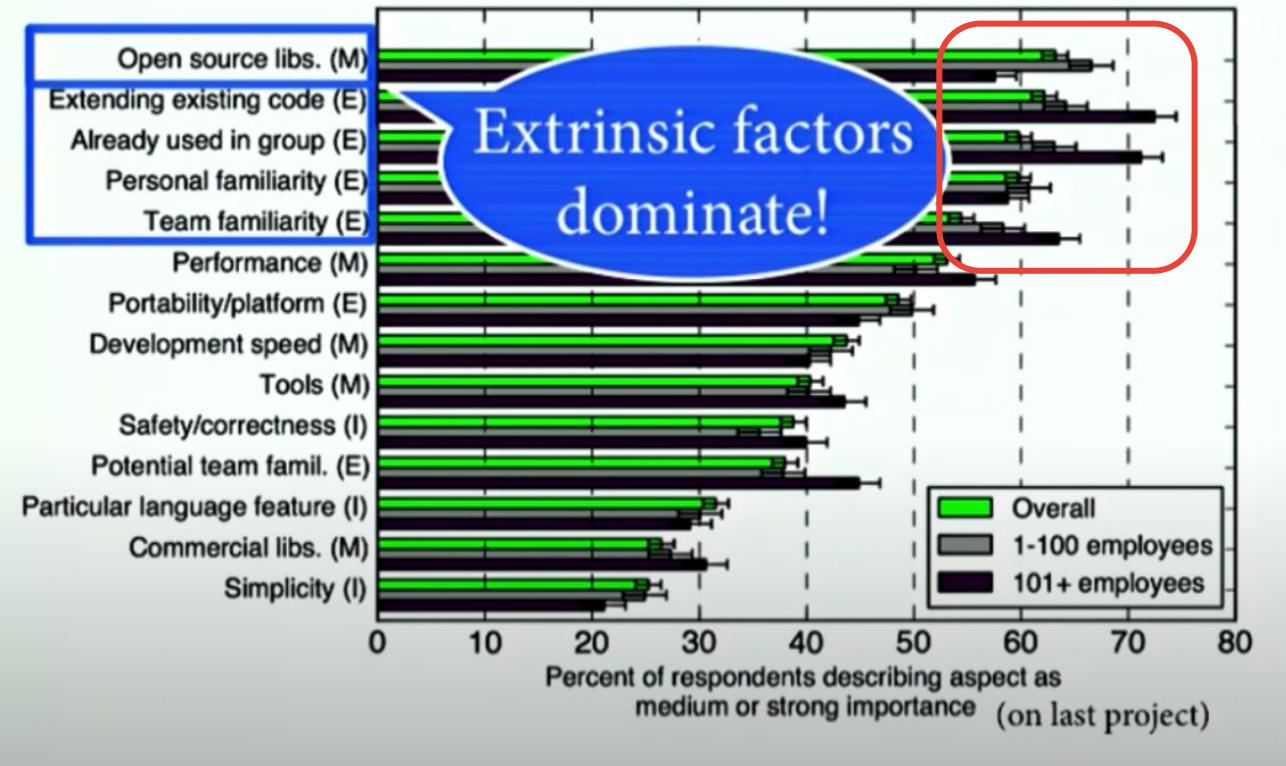
# Factors for Language Choice in Last Project

## Survey of 1,679 Developers



# Factors for Language Choice in Last Project - Size of Company

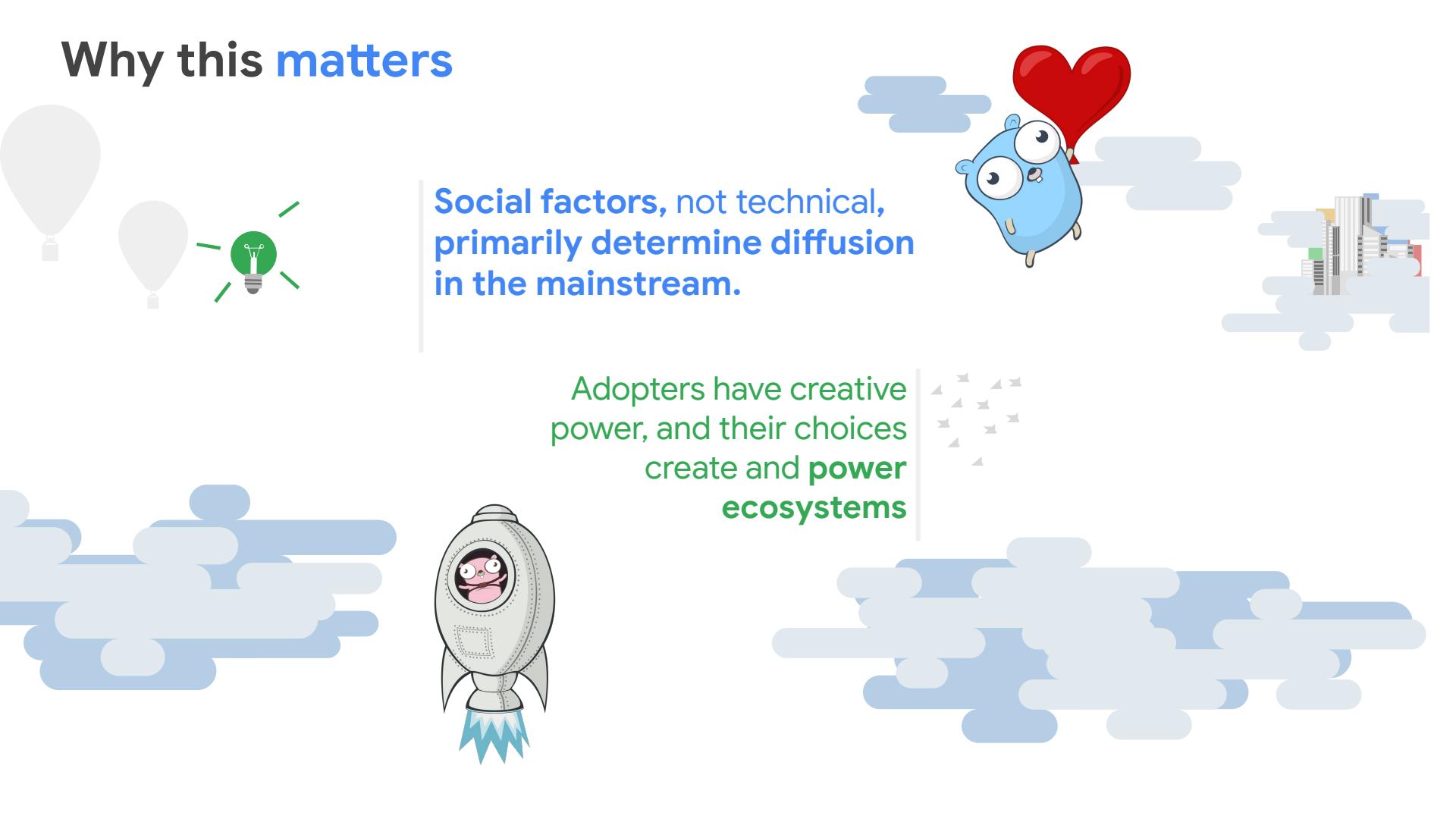
## Survey of 1,679 Developers



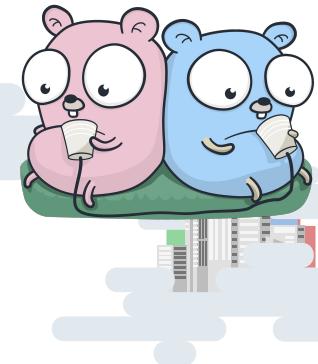
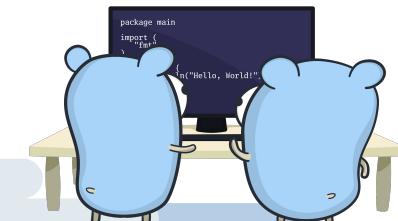
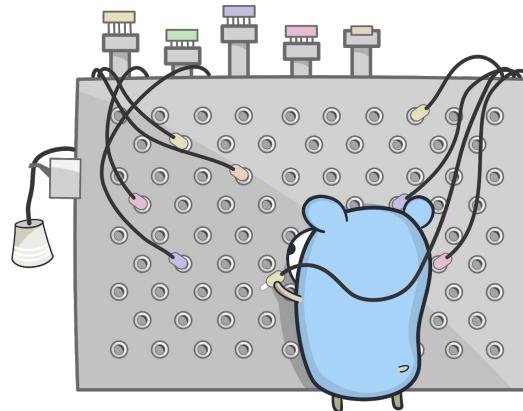
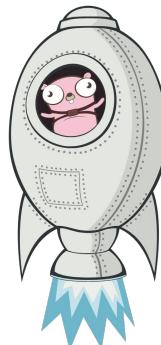
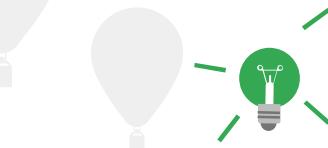
# Why this matters

Social factors, not technical,  
primarily determine diffusion  
in the mainstream.

Adopters have creative  
power, and their choices  
create and power  
ecosystems



# So, what's next to be done for Go? How can you help?



# Go Developer Survey



## Go Developer Survey 2020

The Go Project is conducting research on people's experiences and ideas about Go. Your feedback will help us make improvements to the existing tools and prioritize new features, ensuring a better experience for all Go developers. Once the survey has closed, we will publicly share de-identified and aggregated results with the Go community.

If you use Go, have ever used Go, have ever stopped using Go, or have any interest in the language, please help by sharing your feedback to improve Go for you and your fellow Gophers.

The survey should take about 10 minutes. Your individual responses will not be combined with any other information, and will be handled in accordance with [Google's Privacy Policy](#).

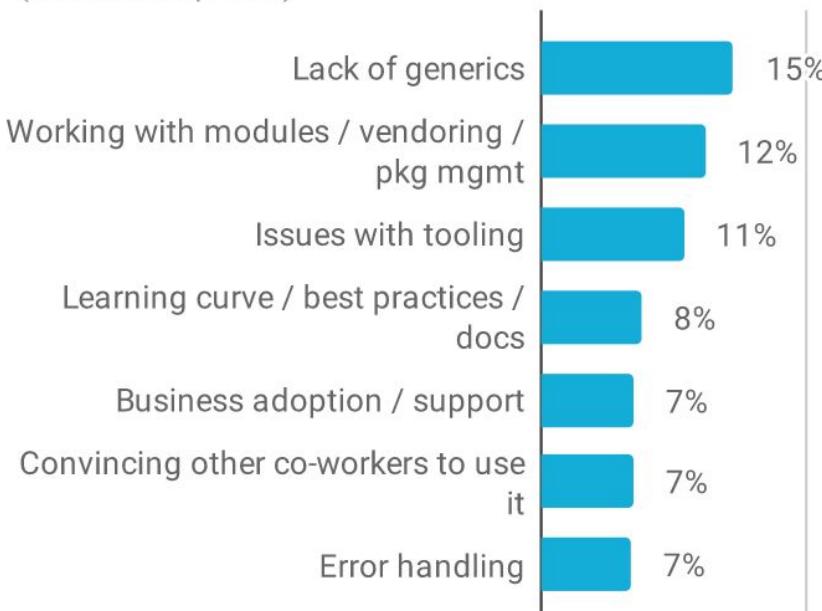
All questions are optional. Please only answer the questions you feel apply to you.

[Continue >>](#)

# Diffusion for the Next Million - Go

What is the biggest challenge you personally face using Go today?

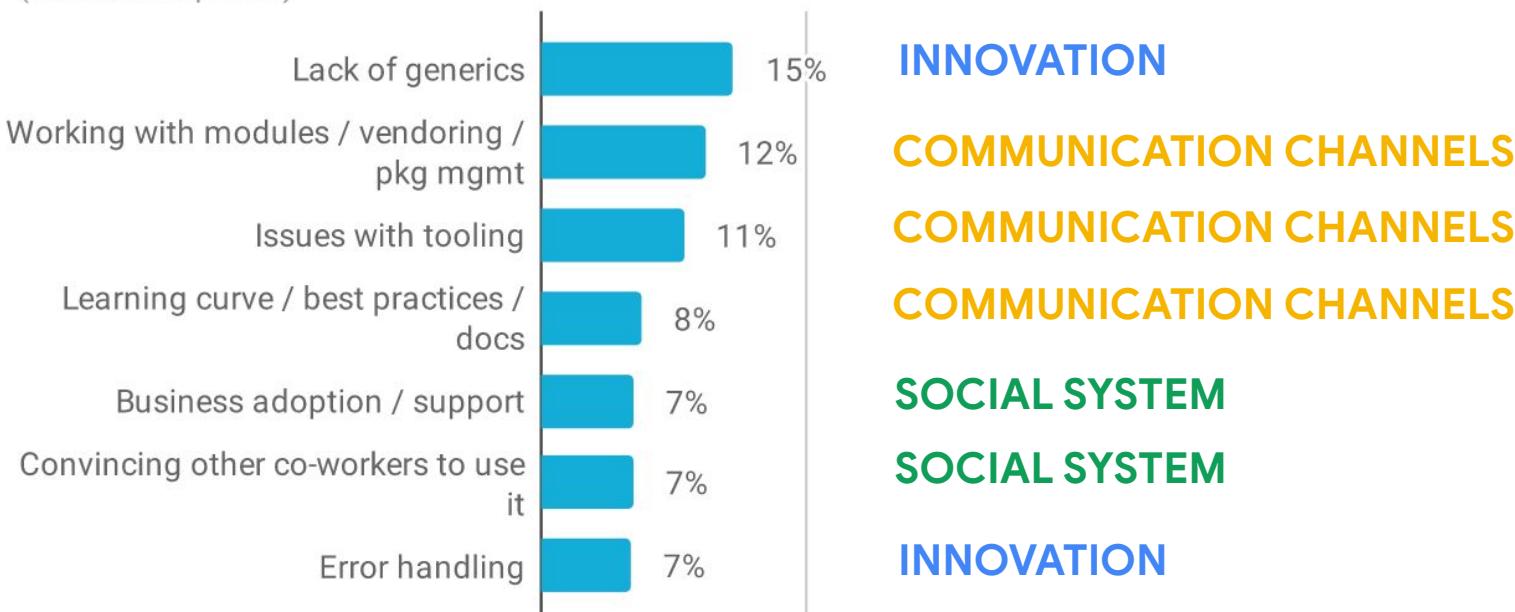
(free-text response)



# Diffusion for the Next Million - Go

What is the biggest challenge you personally face using Go today?

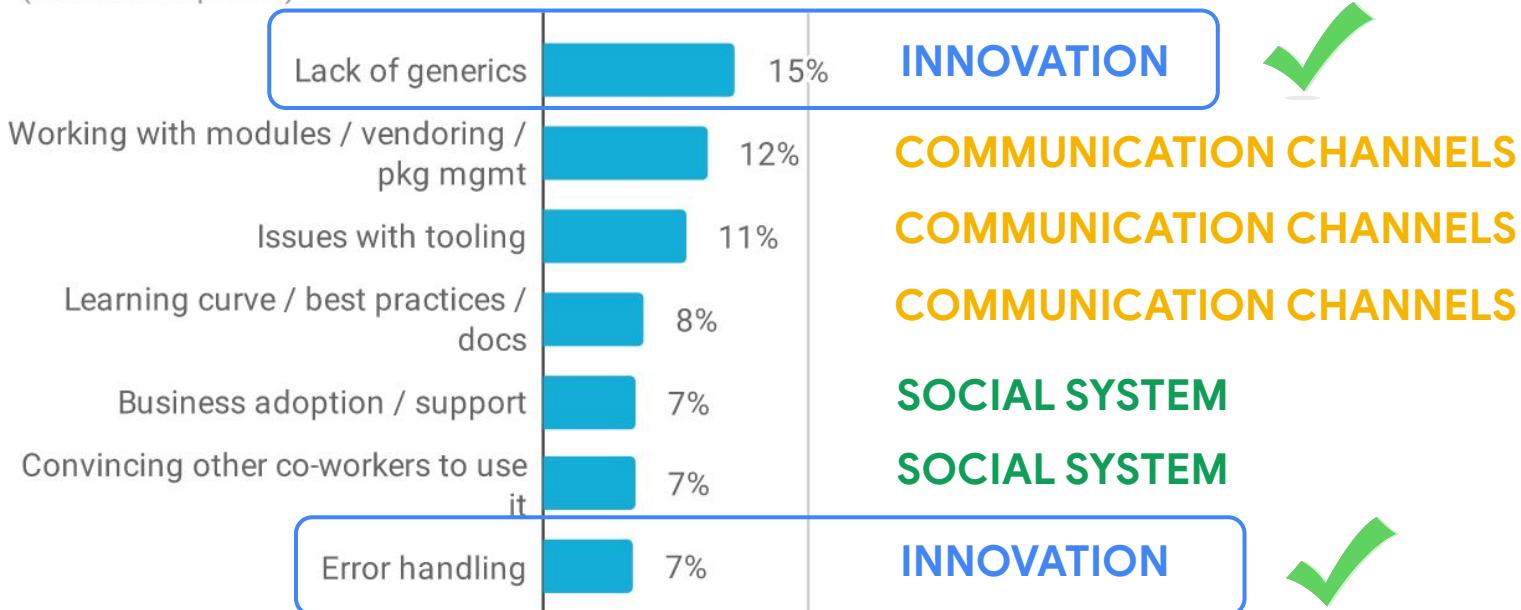
(free-text response)



# Diffusion for the Next Million - Go

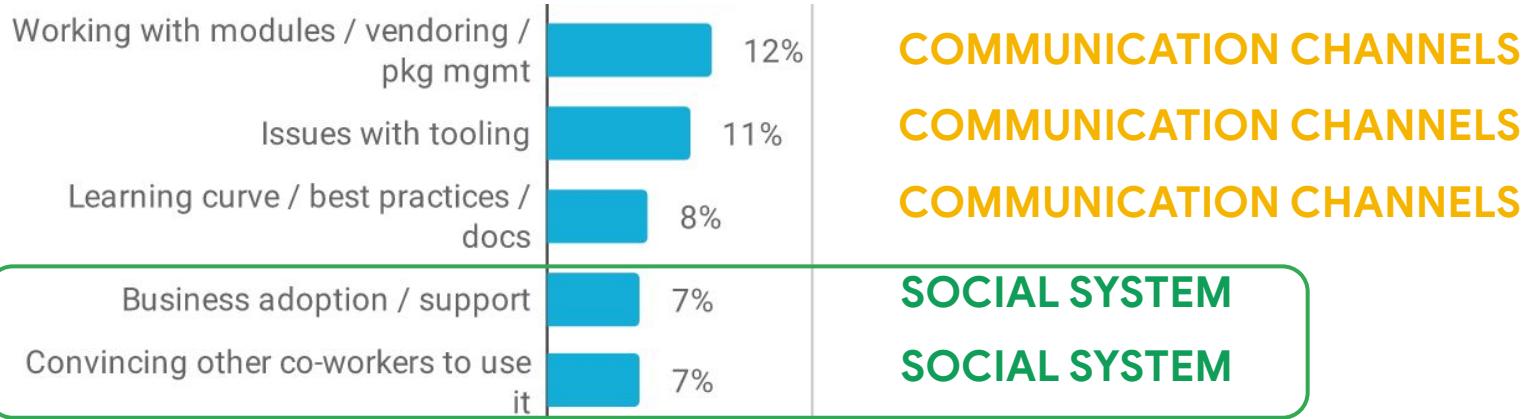
What is the biggest challenge you personally face using Go today?

(free-text response)



# Diffusion for the Next Million - Go

What is the biggest challenge you personally face using Go today?



# Diffusion for the Next Million - Go.dev

GO Search for a package

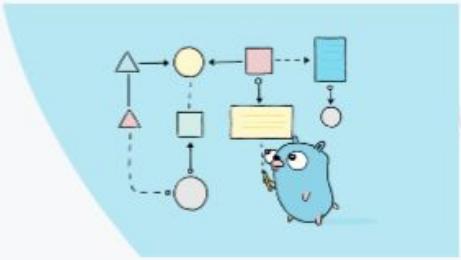
Why Go Getting Started Discover Packages About

RECENTLY UPDATED

## Using Go at Google

Go was created at Google in 2007, and since then, engineering teams across Google have adopted Go to build products and services at massive scale.

Learn more →



Case studies Use cases

---

Business adoption / support	<div style="width: 70%; height: 20px; background-color: #00AEEF;"></div> 7%
Convincing other co-workers to use it	<div style="width: 70%; height: 20px; background-color: #00AEEF;"></div> 7%

**SOCIAL SYSTEM**  
**SOCIAL SYSTEM**

# Diffusion for the Next Million - Go.dev

 <p><b>American Express Uses Go for Payments &amp; Rewards</b></p> <p>Go provides American Express with the speed and scalability it needs for both its payment and rewards networks.</p> <p><a href="#">View case study</a></p>	 <p><b>Capital One - A Serverless and Go Journey</b></p> <p>At the time, no single team member knew Go, but within a month, everyone was writing in Go and we were building out the endpoint layer. The Go code was easy to use, and the really cool concept behind Go (how Go handles native concurrency, garbage collection, and of course safety+speed.) that helped engage us during the build. Also, who can beat that cute mascot!</p> <p><a href="#">View blog post</a></p>	 <p><b>Graceful upgrades in Go</b></p> <p>Cloudflare speeds up and protects millions of websites, APIs, SaaS services, and other properties connected to the internet. "Go is at the heart of Cloudflare's services." By enabling a fast migration for high-latency HTTP connections, our entire DNS infrastructure, SSL, load testing and more."</p> <p><a href="#">View blog post</a></p>	 <p><b>How Facebook built an entity framework in Go</b></p> <p>Learn about a Facebook engineering teams decision to write a new entity framework (ORM) in Go.</p> <p><a href="#">View blog post</a></p>	 <p><b>Using Go at Google</b></p> <p>Go was created at Google in 2007, and since then, engineering teams across Google have adopted Go to build products and services at massive scale.</p> <p><a href="#">View case study</a></p>	 <p><b>Biglice - A cluster computing system in Go</b></p> <p>At GRAIL, we use the Go programming language for most of our bioinformatics, data processing, and machine learning tasks. Go's simplicity makes it easy for newcomers to learn; its transparent runtime semantics makes it easy to reason about performance; and its ability to control data layout and allocation makes it possible to write highly performant data processing code.</p> <p><a href="#">View blog post</a></p>	 <p><b>PayPal Taps Go to Modernize and Scale</b></p> <p>Go's value in producing clean, efficient code that readily scales as software deployment scales made the language a strong fit to support PayPal's goals.</p> <p><a href="#">View case study</a></p>	 <p><b>Salesforce - From Python/C to Go</b></p> <p>One of the big advantages is that Go's cross-platform features make porting code easy.</p> <p><a href="#">View blog post</a></p>	 <p><b>Target - Recommending Go</b></p> <p>We loved the simplified syntax, strong standard library, great external community, and well-built and maintained libraries. We loved how fast multiple times faster Go is than Python, and how quickly we could build when deploying containers. As we continue to grow and scale our technical solutions to our guests, we find the concurrency primitives in Go particularly useful.</p> <p><a href="#">View blog post</a></p>
 <p><b>Cockroach Labs - Why We Chose to Build Our Database with Go</b></p> <p>Go's performance benefits, garbage collection, and low barrier to entry made it a great fit for CockroachDB.</p> <p><a href="#">View blog post</a></p>	 <p><b>How Curve is getting ahead with Golang</b></p> <p>Curve shares how Go's efficiency, standard library, and thriving community help them move banking to the cloud.</p> <p><a href="#">View blog post</a></p>	 <p><b>Dropbox - Open sourcing our Go libraries</b></p> <p>About a year ago, we decided to migrate our performance-critical backends from Python to Go to leverage better concurrency support and faster execution speed... At this point, we have successfully moved major parts of our infrastructure to Go.</p> <p><a href="#">View blog post</a></p>	 <p><b>MercadoLibre Grows with Go</b></p> <p>Go provides clean, efficient code that readily scales as MercadoLibre's online commerce grows, and increases developer productivity by allowing their engineers to serve their ever-increasing audience while writing less code.</p> <p><a href="#">View case study</a></p>	 <p><b>How Microsoft Embraces Go</b></p> <p>Learn about how Microsoft has helped support Go and how it uses Go to power pieces of its cloud infrastructure.</p> <p><a href="#">View blog post</a></p>	 <p><b>Netflix - Application data caching using SSDs</b></p> <p>The decision to use Go was deliberate, because we needed something that had lower latency than Java (where garbage collection pauses are an issue) and is more productive for developers than C, while also handling tens of thousands of client connections. Go fits this space well.</p> <p><a href="#">View blog post</a></p>	 <p><b>Twitch - Go's march to low latency GC</b></p> <p>We use Go at Twitch for many of our busiest systems. Its simplicity, safety, performance, and readability make it a good tool for the problems we encounter with serving live video and chat to our millions of users.</p> <p><a href="#">View blog post</a></p>	 <p><b>Twitter - 5 billion sessions a day in realtime</b></p> <p>We now see about five billion sessions per day and growing. Hundreds of millions of devices send millions of events every second to the Answers endpoint. During the time that it took you to read to here, the Answers back-end has received and processed about 10,000,000 analytics events.</p> <p><a href="#">View blog post</a></p>	 <p><b>Uber - GPU-power analytics engine in Go</b></p> <p>AresDB (written in Go) is widely used at Uber to power our real-time data analysis dashboards, enabling us to make data-driven decisions at scale about myriad aspects of our business.</p> <p><a href="#">View blog post</a></p>

Business adoption / support

Convincing other co-workers to use it

7%

7%

**SOCIAL SYSTEM**

**SOCIAL SYSTEM**



# Diffusion for the Next Million - pkg.go.dev

Black Lives Matter

Support the Equal Justice Initiative

GO Search for a package

Discover Packages > github.com/sirupsen/logrus

**logrus** package module

Version v1.7.0 LATEST | May 28, 2020 | MIT | 18 Imports | 20000+ Imported by

Jump to ...

**README**

**Logrus** build passing godoc reference

Logrus is a structured logger for Go (golang), completely API compatible with the standard library logger.

Logrus is in maintenance-mode. We will not be introducing new features. It's simply too hard to do in a way that won't break many people's projects, which is the last thing you want from your Logging library (again...).

This does not mean Logrus is dead. Logrus will continue to be maintained for security, (backwards compatible) bug fixes, and performance (where we are limited by the interface).

I believe Logrus' biggest contribution is to have played a part in today's widespread use of structured logging in Golang. There doesn't seem to be a reason to do a major, breaking iteration into Logrus V2, since the fantastic Go community has built those

Expand ▾

**Documentation**

**Overview**

Package logrus is a structured logger for Go, completely API compatible with the standard library logger.

GO Search for a package

Discover Packages > github.com/sirupsen/logrus

**logrus** package module

← Go to main page

v1 – github.com/sirupsen/logrus

v1.7.0 – May 28, 2020

v1.6.0 – May 2, 2020

v1.5.0 – Mar 22, 2020

v1.4.2 – May 18, 2019

v1.4.1 – Apr 2, 2019

v1.4.0 – Mar 11, 2019

v1.3.0 – Jan 2, 2019

v1.2.0 – Nov 1, 2018

v1.1.1 – Oct 8, 2018

v1.1.0 – Sep 25, 2018

v1.0.6 – Jul 21, 2018

Business adoption / support

7%

Convincing other co-workers to use it

7%

**SOCIAL SYSTEM**  
**SOCIAL SYSTEM**



# Diffusion for the Next Million - pkg.go.dev

Discover Packages > [github.com/sirupsen/logrus](https://pkg.go.dev/github.com/sirupsen/logrus) 

## logrus

package

module

Version v1.7.0

LATEST



May 28, 2020



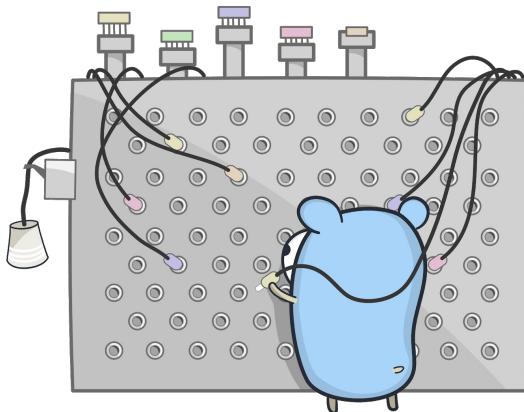
MIT



18 Imports



20000+ Imported by



### Imports

[golang.org/x/sys/unix](https://golang.org/x/sys/unix)

### Standard library Imports

bufio  
bytes  
context  
encoding/json  
fmt  
io  
log  
os  
reflect  
runtime  
sort  
strconv  
strings  
sync  
sync/atomic  
time  
unicode/utf8

Known importers: 46952

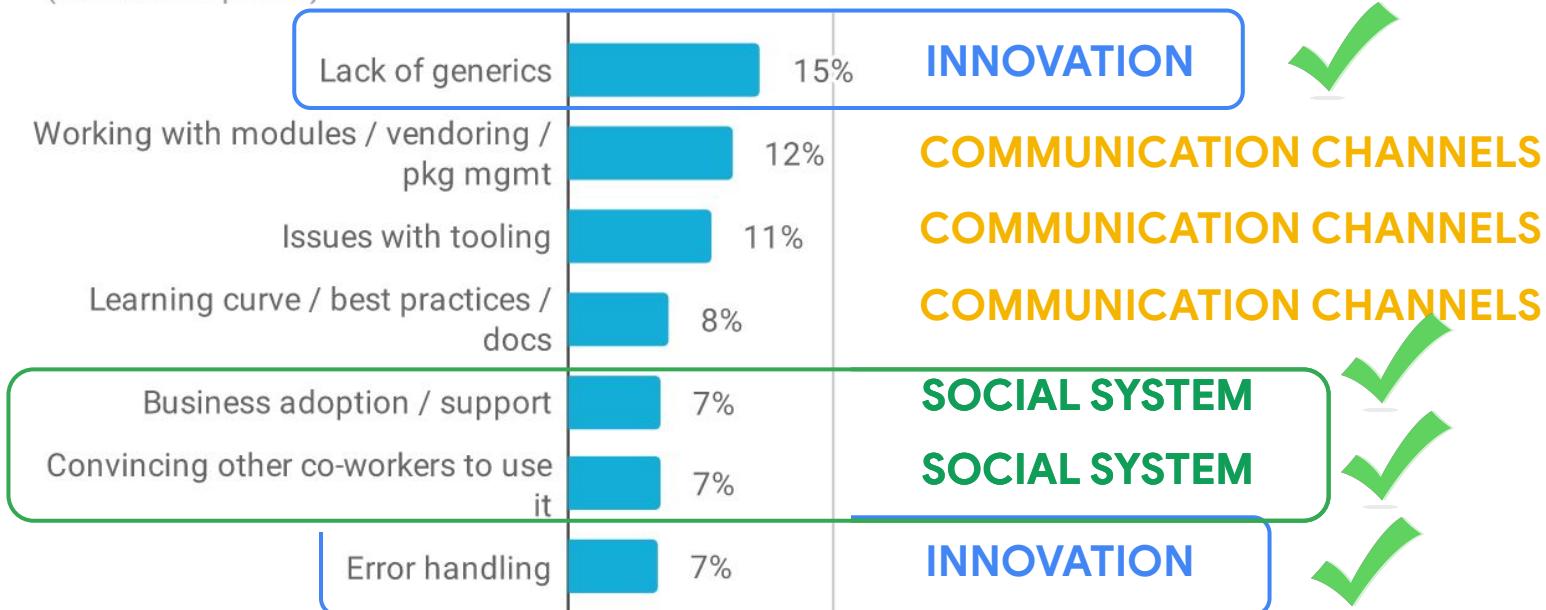
[admiralty.io/multicluster-scheduler/cmd/agent](https://admiralty.io/multicluster-scheduler/cmd/agent)

- ▶ [agones.dev/agones/](https://agones.dev/agones/) (20)
- ▶ [alexejk.io/portal](https://alexejk.io/portal) (4)
- ▶ [arcadium.dev/pkg/log/logrus](https://arcadium.dev/pkg/log/logrus)
- ▶ [astuart.co/slot](https://astuart.co/slot)
- ▶ [atec.pub/canon/fetch](https://atec.pub/canon/fetch)
- ▶ [bind.ch/git/phil/dbx](https://bind.ch/git/phil/dbx)
- ▶ [bitbucket.butik.ru/WB/goerrors](https://bitbucket.butik.ru/WB/goerrors)
- ▶ [bitbucket.org/4s/go-logging](https://bitbucket.org/4s/go-logging)
- ▶ [bitbucket.org/ROKT/](https://bitbucket.org/ROKT/) (7)
- ▶ [bitbucket.org/Shekelbrainweigh/listensor/logging](https://bitbucket.org/Shekelbrainweigh/listensor/logging)
- ▶ [bitbucket.org/Szkil/ZLogger](https://bitbucket.org/Szkil/ZLogger)
- ▶ [bitbucket.org/TIKI-Institut/ai-common-go/](https://bitbucket.org/TIKI-Institut/ai-common-go/) (9)
- ▶ [bitbucket.org/Tensor146/tribo/](https://bitbucket.org/Tensor146/tribo/) (8)
- ▶ [bitbucket.org/advbet/](https://bitbucket.org/advbet/) (5)
- ▶ [bitbucket.org/airenas/listgo/internal/pkg/cmdapp](https://bitbucket.org/airenas/listgo/internal/pkg/cmdapp)
- ▶ [bitbucket.org/andre\\_puankare/book/](https://bitbucket.org/andre_puankare/book/) (2)
- ▶ [bitbucket.org/atomgun/](https://bitbucket.org/atomgun/) (2)
- ▶ [bitbucket.org/axbannaz/sizetest/go/logger](https://bitbucket.org/axbannaz/sizetest/go/logger)
- ▶ [bitbucket.org/bavusua/store\\_tools/](https://bitbucket.org/bavusua/store_tools/) (3)
- ▶ [bitbucket.org/bexstech/bexs-devops-exam/backend/src/backend](https://bitbucket.org/bexstech/bexs-devops-exam/backend/src/backend)
- ▶ [bitbucket.org/binarysoftware/framework/](https://bitbucket.org/binarysoftware/framework/) (2)
- ▶ [bitbucket.org/blackxcloudeng/](https://bitbucket.org/blackxcloudeng/) (24)
- ▶ [bitbucket.org/browserlondon/SwanToolkit](https://bitbucket.org/browserlondon/SwanToolkit)

# Diffusion for the Next Million - Go

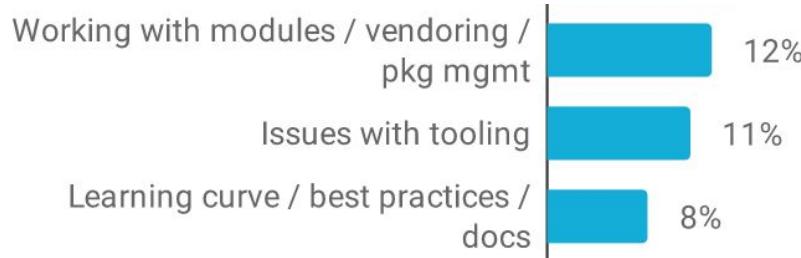
What is the biggest challenge you personally face using Go today?

(free-text response)



# Diffusion for the Next Million - Go

What is the biggest challenge you personally face using Go today?



**COMMUNICATION CHANNELS**

**COMMUNICATION CHANNELS**

**COMMUNICATION CHANNELS**

# Diffusion for the Next Million - Go Existing Communication Channels -



Documents Packages

The Project

Help

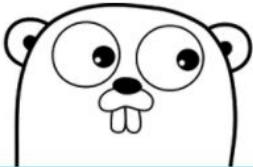
Blog

Play

Search



Go is an open source programming language that makes it easy to build **simple, reliable, and efficient** software.



Download Go

Binary distributions available for Linux, macOS, Windows, and more.

Featured articles

Featured video

Try Go

Open in Playground

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

import "fmt"

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

Hello, World!

Run

Share

Tour

A Tour of Go

Hello, 世界

Welcome to a tour of the Go programming language.

The tour is divided into a list of modules that you can access by clicking on A Tour of Go on the top left of the page.

You can also view the table of contents at any time by clicking on the menu on the top right of the page.

Throughout the tour you will find a series of slides and exercises for you to complete.

You can navigate through them using

"previous" or PageUp to go to the previous page,  
"next" or PageDown to go to the next page.

The tour is interactive. Click the Run button now (or press Shift + Enter) to compile and run the program on a remote server. The result is displayed below the code.

These example programs demonstrate different aspects of Go. The programs in the tour are meant to be starting points for your own experimentation.

Edit the program and run it again.

When you click on Format (shortcut: Ctrl + Enter), the text in the editor is formatted using the golint tool. You can switch syntax highlighting on and off by clicking on the syntax button.

When you're ready to move on, click the right arrow below or type the PageDown key.

```
hello.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Hello, 世界")
7 }
```

Imports off Syntax off

Reset Format Run



# Diffusion for the Next Million - Go

## Existing Communication Channels -



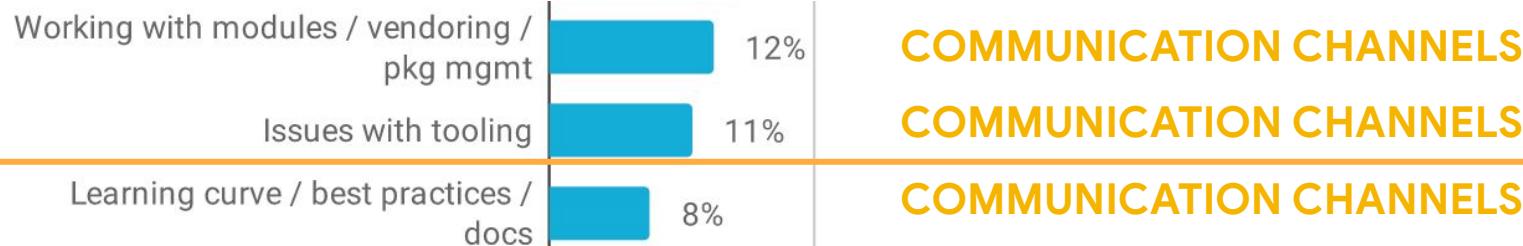
The Go Playground interface is shown. The top bar includes navigation icons (back, forward, search, home) and a URL field showing "play.golang.org". Below the bar, the title "The Go Playground" is displayed next to buttons for "Run", "Format", "Imports", "Share", and a dropdown menu set to "Hello, playground". The main area is a code editor containing the following Go code:

```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     fmt.Println("Hello, playground")
9 }
10
11
12
13
14
15
16
17
18
19
20
21
22
```

# Diffusion for the Next Million - Go

## Communication Challenges

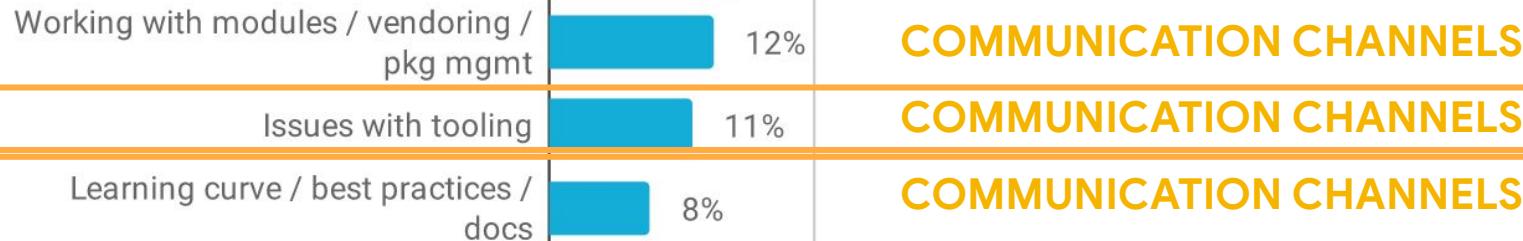
What is the biggest challenge you personally face using Go today?



# Diffusion for the Next Million - Go

## Communication Challenges

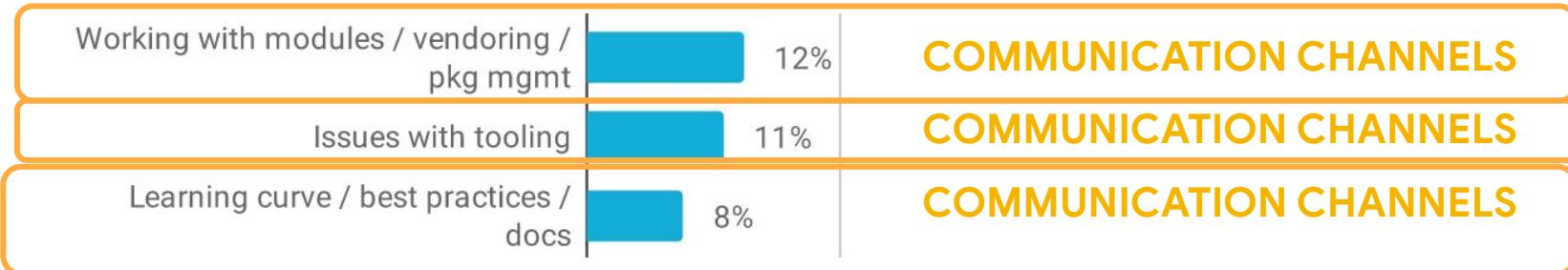
What is the biggest challenge you personally face using Go today?



# Diffusion for the Next Million - Go

## Communication Challenges

What is the biggest challenge you personally face using Go today?



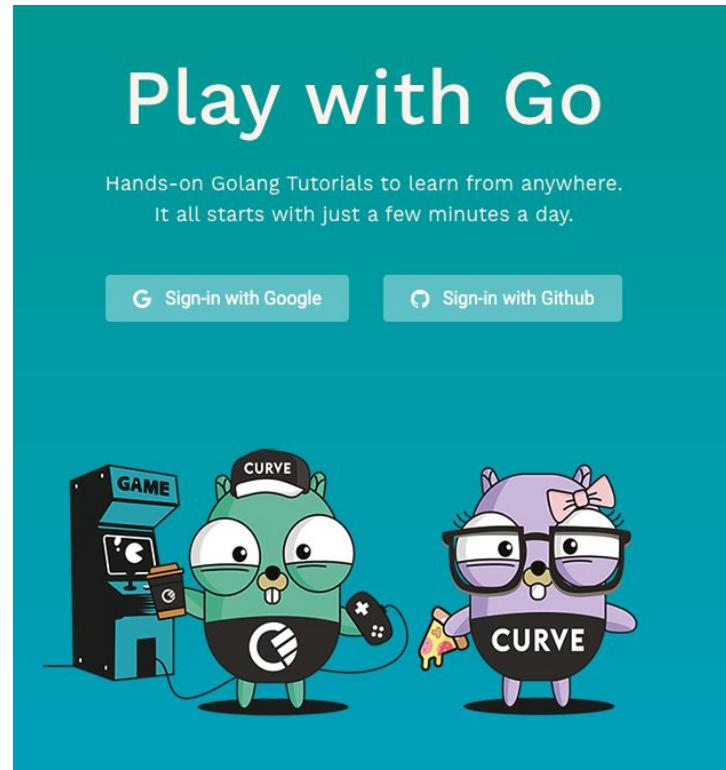
# Diffusion for the Next Million - Go Solution: Participatory Interactive Environment

play-with-go.dev

-For students, trainers, content creators, teachers.

-For the community, by the community.

-Open Source (contributions welcome)



# Diffusion for the Next Million - Go Solution: Participatory Interactive Environment

## Play-with-go.dev

- Write your own interactive tutorials
- For the community, by the community
- head over to [#play-with-go](#) on Gophers Slack

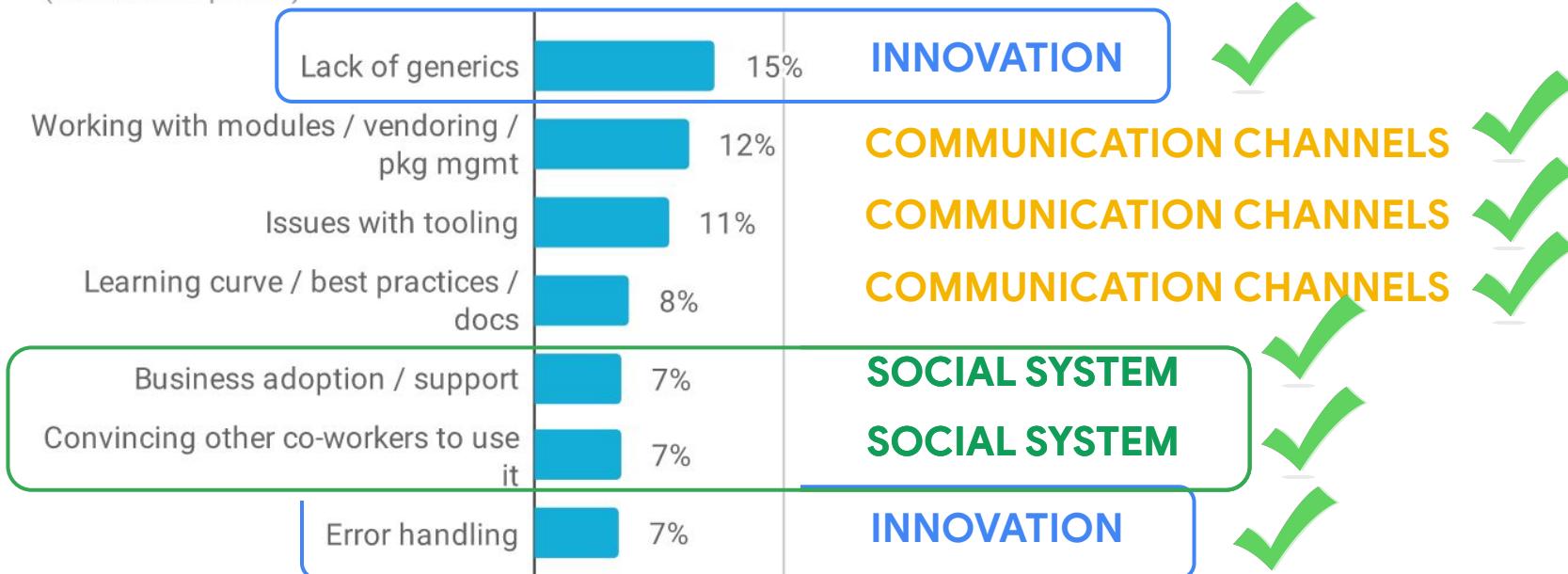
The screenshot shows the homepage of play-with-go.dev. The header features a teal background with yellow text. It says "Hi Friend! Welcome to Play with Go" and "Join these challenges and learn how to build and share a containerized app and so much more. Sounds fun? Get started now!" Below the header, there are six tutorial cards arranged in a 3x2 grid. Each card has a white background and a teal border. The cards are:

- Using Staticcheck: static analysis for the win**  
With supporting text below as a natural lead-in to additional content.  
[Start tutorial](#)
- Retracting Module Versions**  
With supporting text below as a natural lead-in to additional content.  
[Start tutorial](#)
- A simple Go modules guide**  
With supporting text below as a natural lead-in to additional content.  
[Start tutorial](#)
- Installing Go programs standalone**  
With supporting text below as a natural lead-in to additional content.  
[Start tutorial](#)
- Developer tools as module dependencies**  
With supporting text below as a natural lead-in to additional content.  
[Start tutorial](#)
- Tutorial: Go fundamentals**  
With supporting text below as a natural lead-in to additional content.  
[Start tutorial](#)
- Tutorial: Get started with Go**  
With supporting text below as a natural lead-in to additional content.  
[Start tutorial](#)
- An introduction to play-with-go.dev guides**  
With supporting text below as a natural lead-in to additional content.  
[Start tutorial](#)

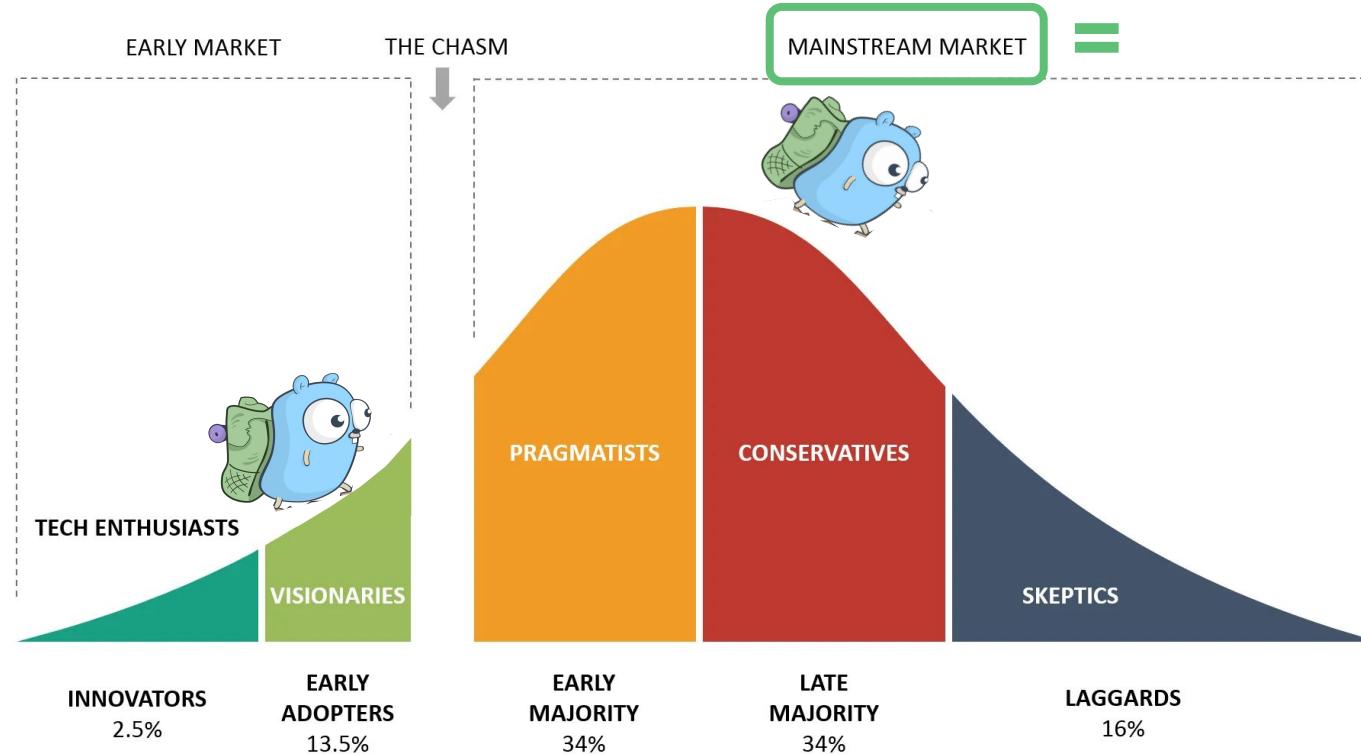
# Diffusion for the Next Million - Go

What is the biggest challenge you personally face using Go today?

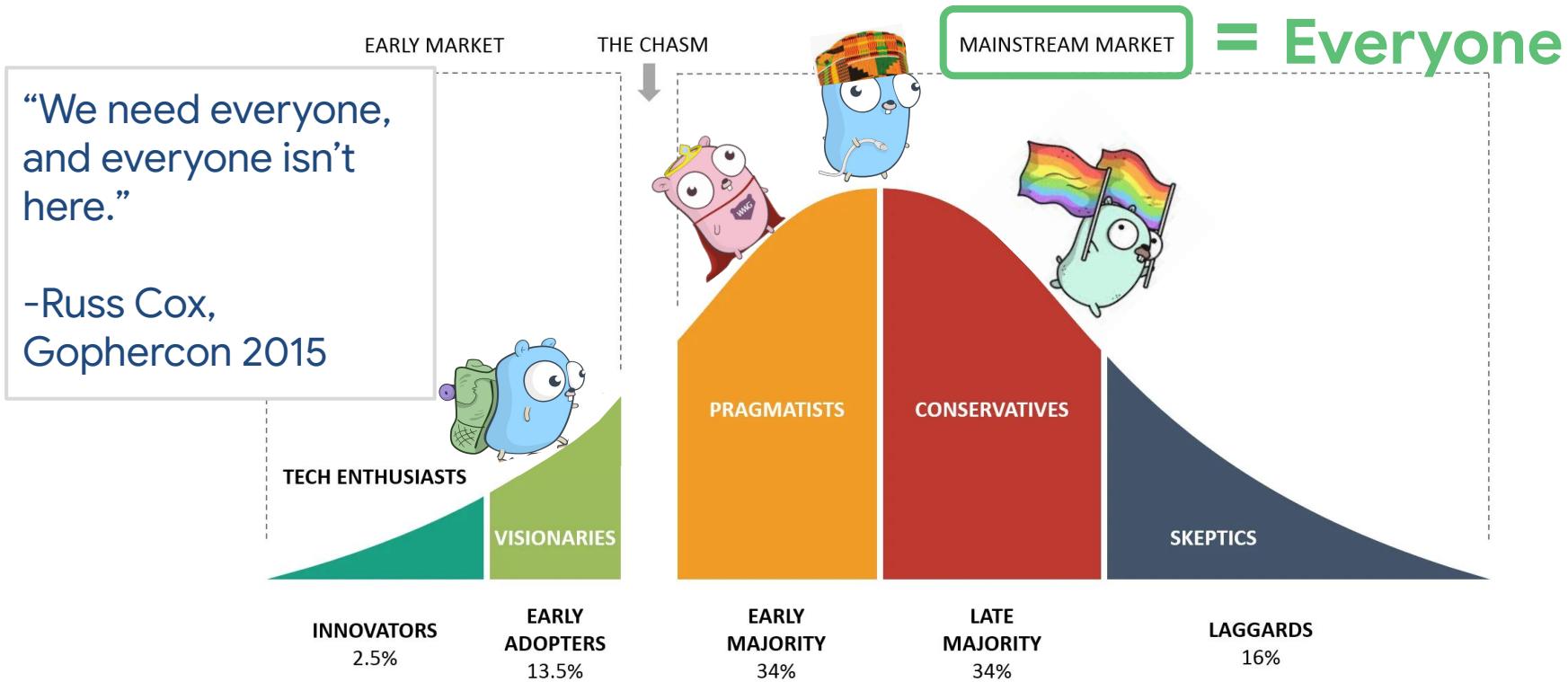
(free-text response)



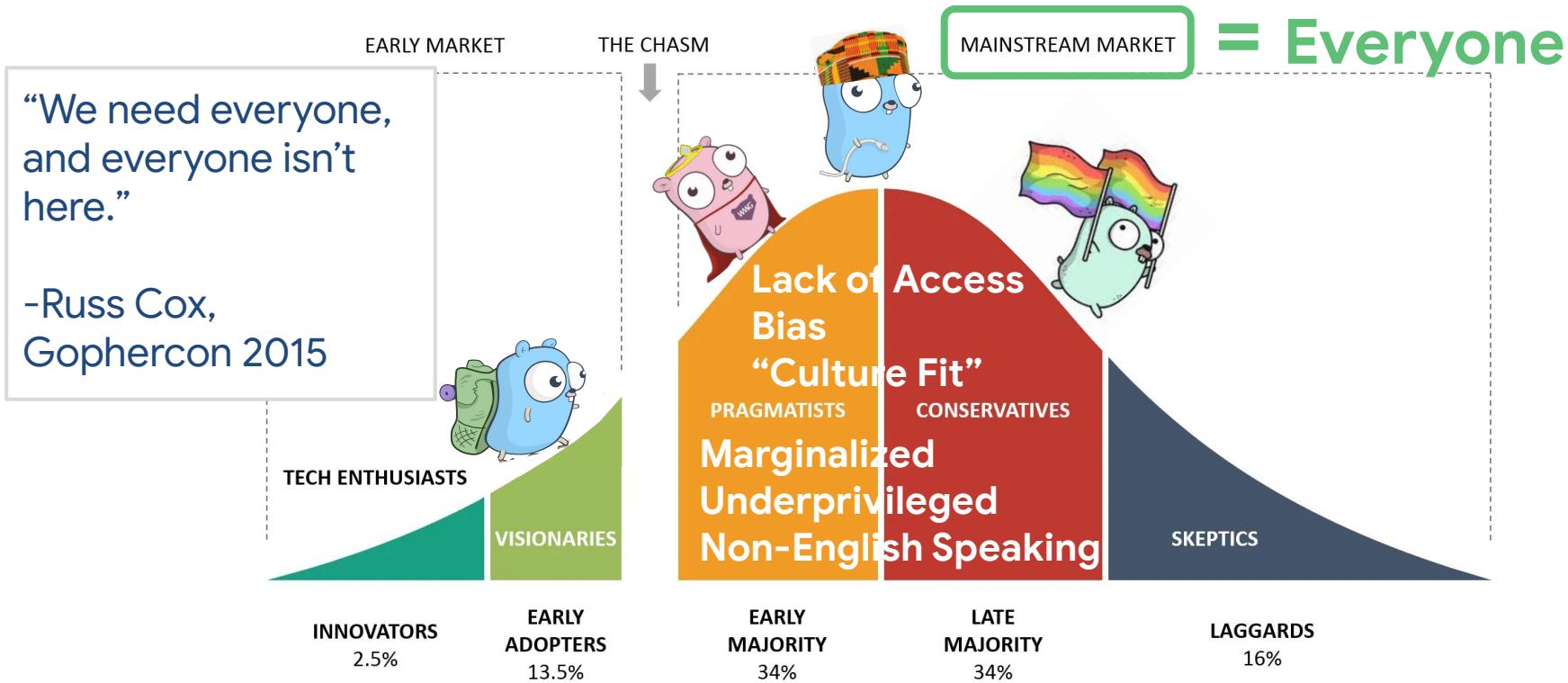
# Crossing the Chasm - Go Programming Language



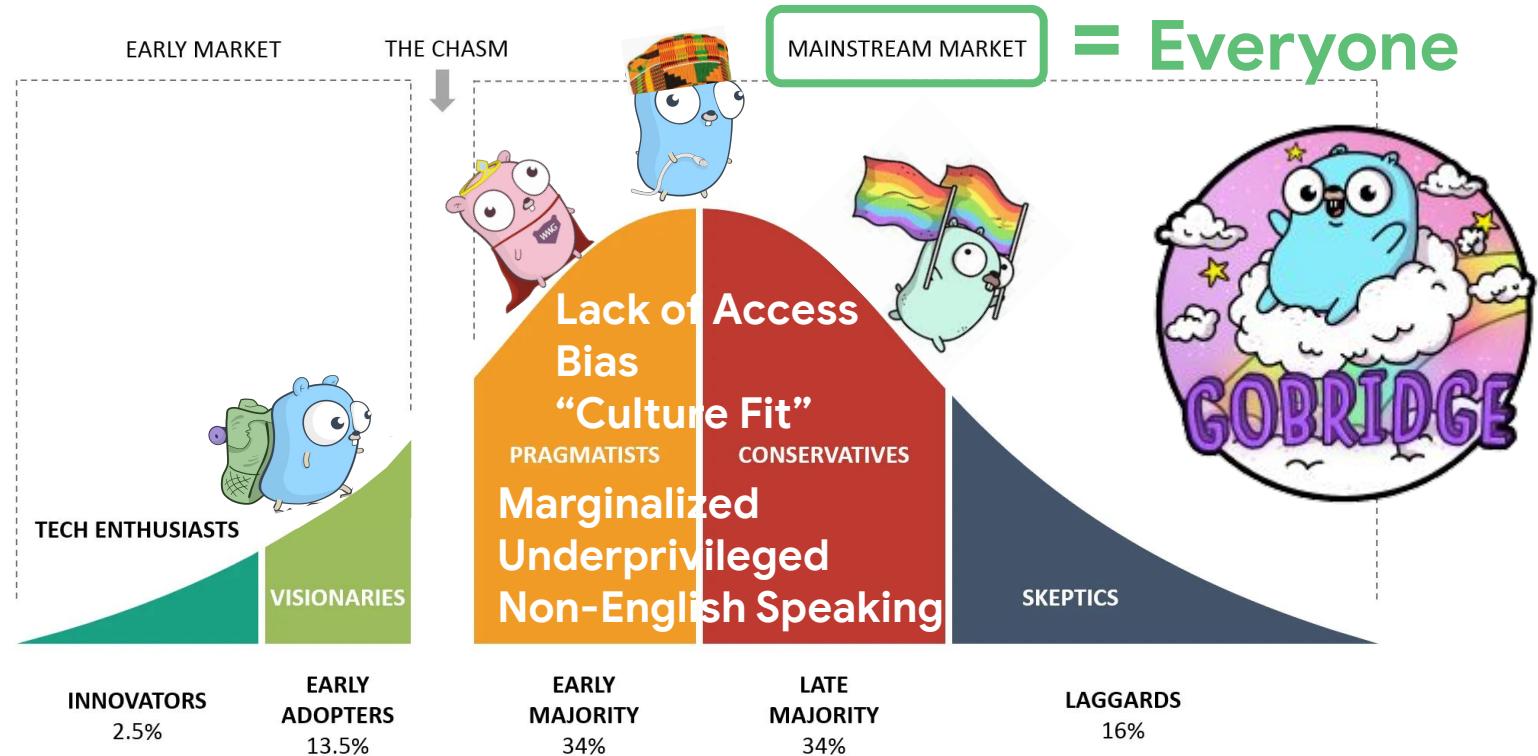
# Crossing the Chasm - Go Programming Language



# Crossing the Chasm - Go Programming Language



# Crossing the Chasm - Go Mentoring Platform



# Future of Go's Ecosystem



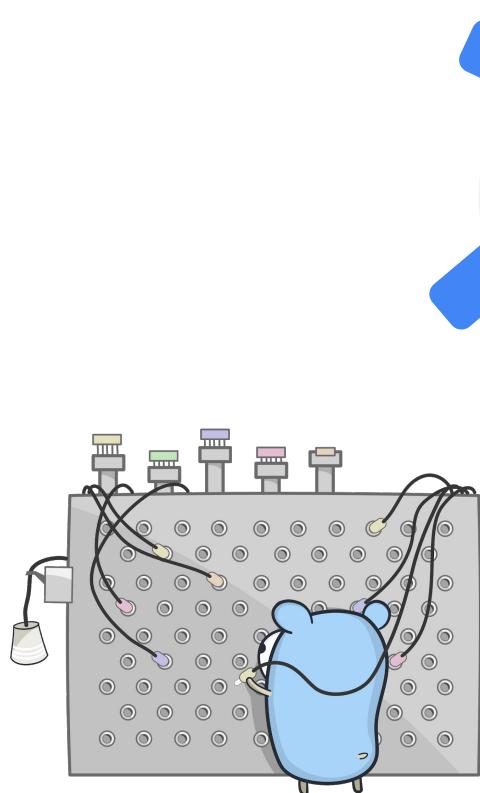
# Thank You



Carmen Andoh, Google  @carmatrocitY

# Diffusion in an [Open Source] Ecosystem

INNOVATION



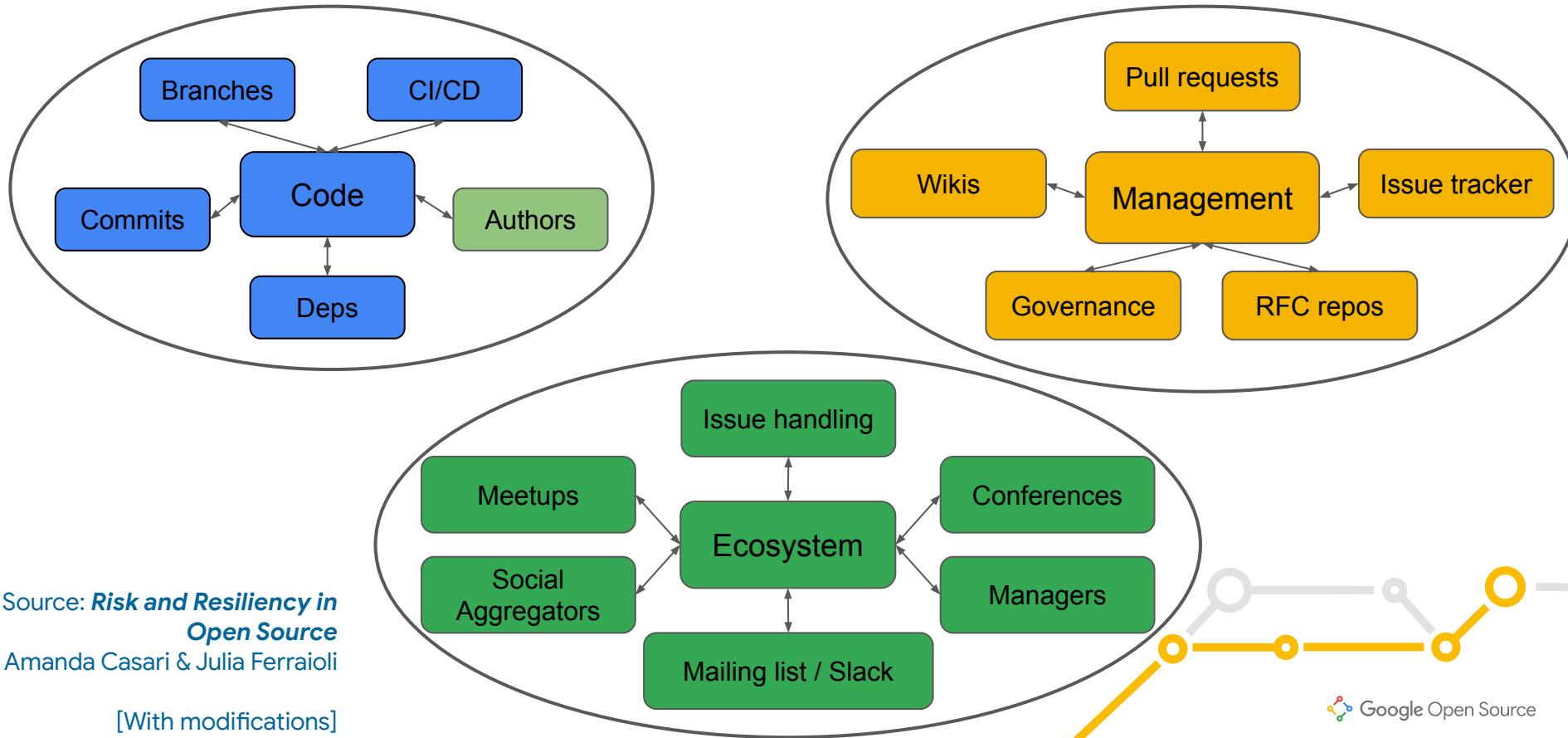
COMMUNICATION CHANNELS



SOCIAL SYSTEM



# Open Source Ecosystems as Complex Networks



# Diffusion for the Next Million - Stack Overflow

The screenshot shows two Stack Overflow tag pages side-by-side. On the left is the 'go' tag page, which has 49922 questions and 30 asked today. On the right is the 'django' tag page, which has 59.1k watchers and 49.9k questions. Both pages provide a brief description of the language.

stackoverflow

go

an open-source programming language. It is statically-typed, with a syntax loosely derived from C, adding automatic memory management, typ...

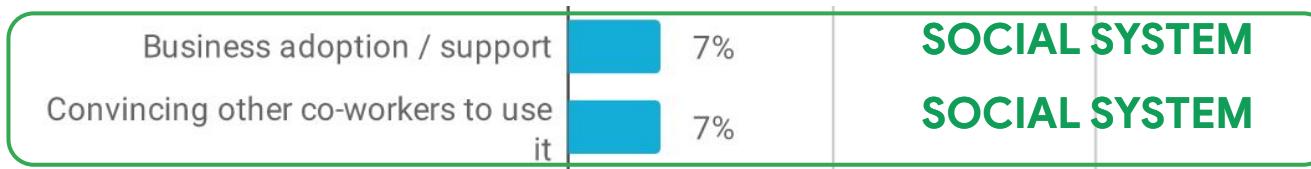
49922 questions      30 asked today, 199 this week

go

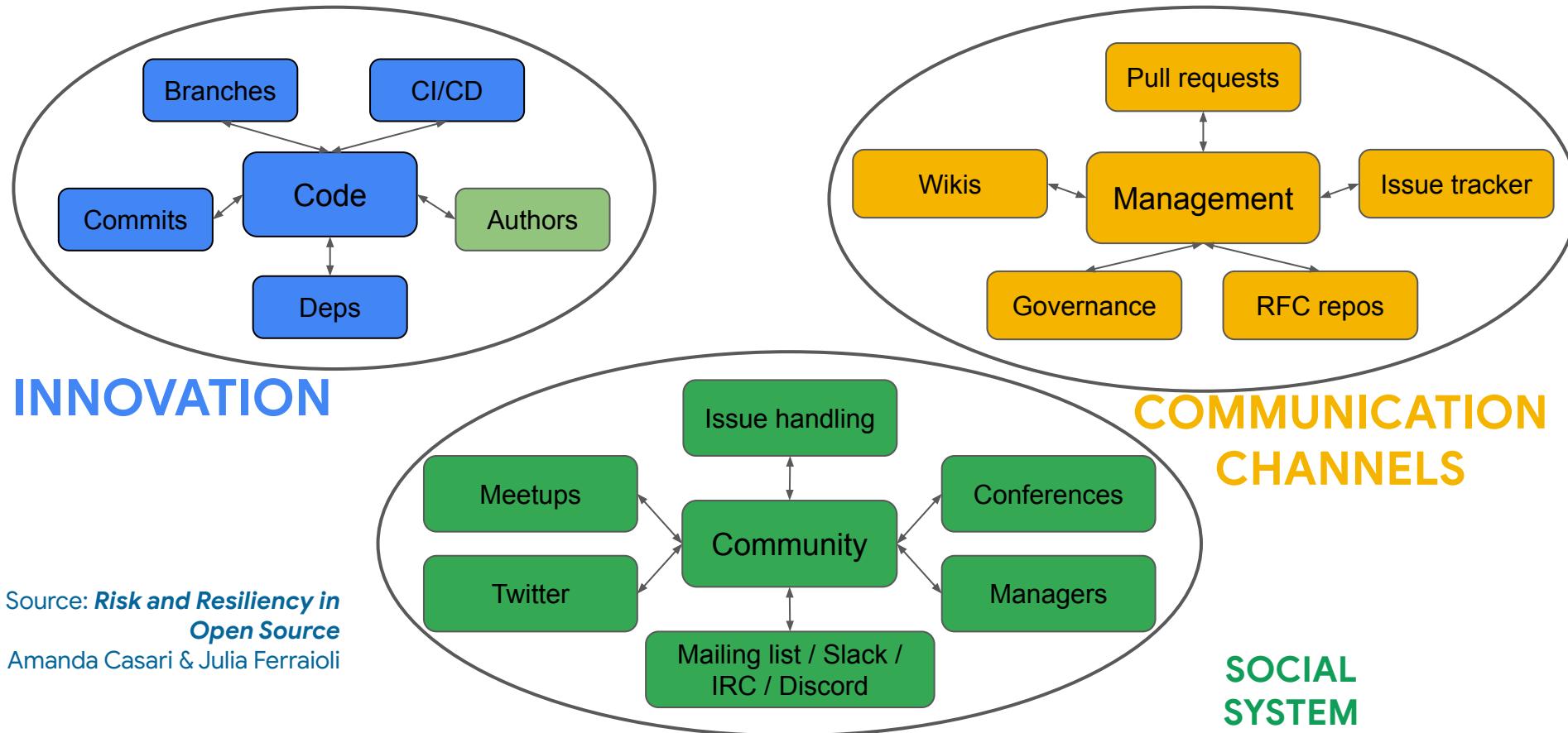
59.1k watchers 49.9k questions

Go is an open-source programming language. It is statically-typed, with a syntax loosely derived from C, adding automatic memory management, type safety, some dynamic typing capabilities, additional built-in types such as variable-length arrays (called slices) and key-value maps, and a large standard library. [View tag](#)

django

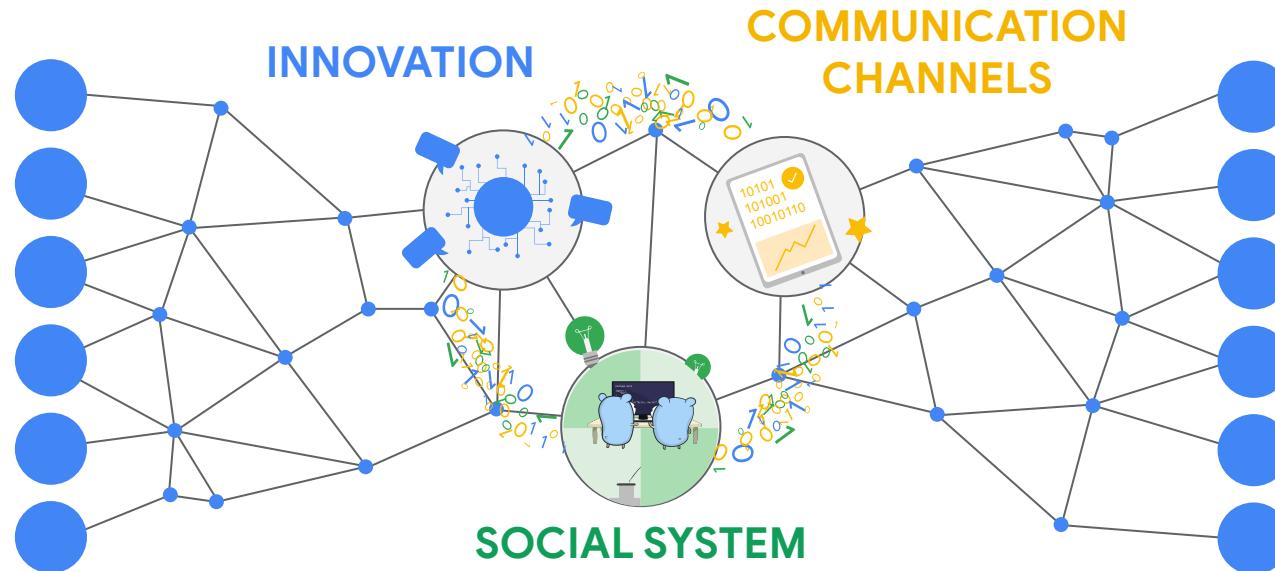


# Open Source Ecosystems as Complex Networks



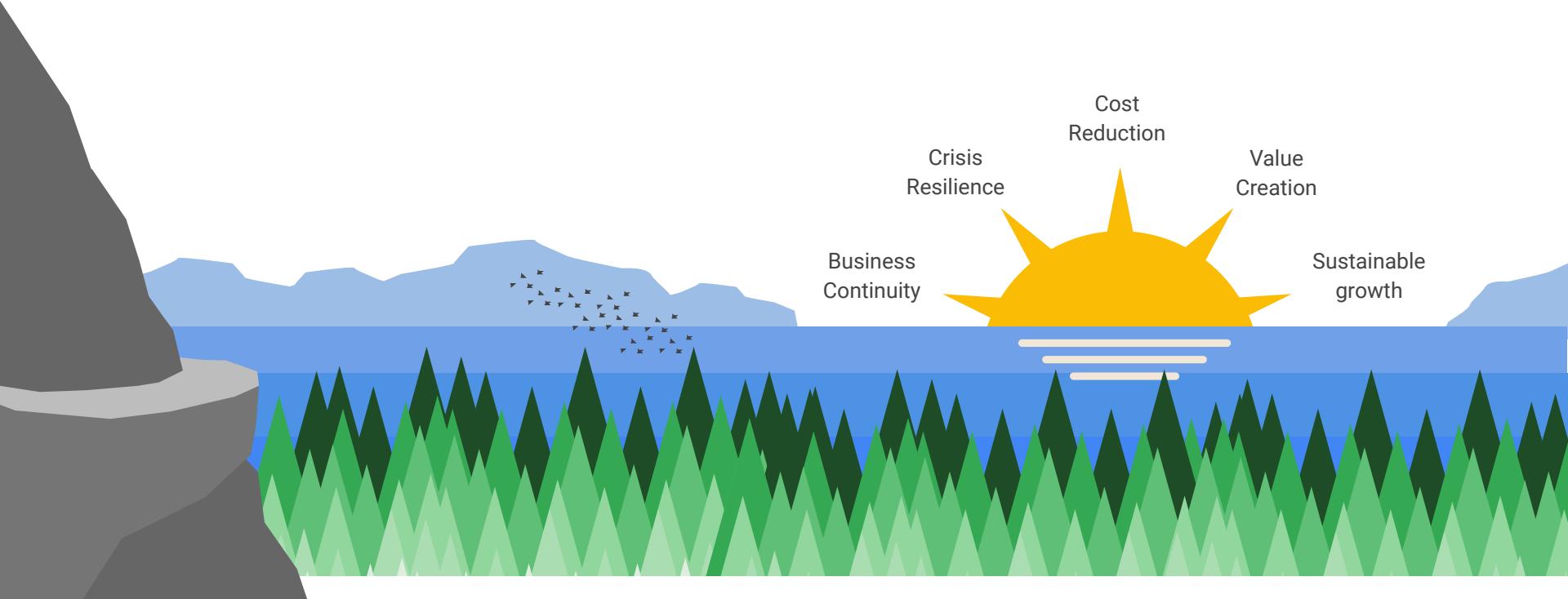
# Measuring and Mapping Risk and Resilience in Ecosystems

## University of Vermont Complex Systems Center





# Crossing the Chasm: Go for the Next Million Users



# Socio-PLT: Sociological Programming Language Theory

[What is Socio-PLT?](#) | [Video](#) | [Further Resources](#) | Created by [Leo Meyerovich](#) and [Ari Rabkin](#) (EECS @ Berkeley)

**Visualizations:** [Language/statement ranking](#) [Statement correlation](#) [Language correlation](#) [Statement clustering](#) [Language clustering](#)

**Why does one language succeed and another one fail?** To answer questions like this, we are examining sociological aspects of programming language theory: socio-PLT. This varies from establishing first principles (see [our survey of sociological research](#)) to building socially-optimized languages. The interactive visualizations here show some of our recent efforts for a **quantitative analysis of programming language perceptions**.

We're examining large-scale activity, and a lot of it:

- Thousands of respondents from Berkeley's recent [massive open online course on software as a service](#), most of whom are working developers that already completed at least a bachelors in computer science.
- A decade of [SourceForge repository data](#), including over 300,000 projects
- Two years of the the [Hammer Principle online survey](#) on programming languages, including over 13,000 respondents