Five things Go taught me about Open Source

Velocity Berlin 2019

What makes an open source project popular?

Proposition 1

Velocity is not a programming language conference. You're not here to hear a dork from Australia rabbit on about Go.

Proposition 2

Open source has influenced—at least in part—the career of everyone in this room.

“As of November 2019, my best estimate is between 1.15 and 1.96 million.”

–Russ Cox, https://research.swtch.com/gophercount

Was it a maniacal focus on simplicity?

Was it solid performance?

Was it modern approach to concurrency?

Stoic commitment to backwards compatibility?

Google's money?

The distribution model

Single, static, binary

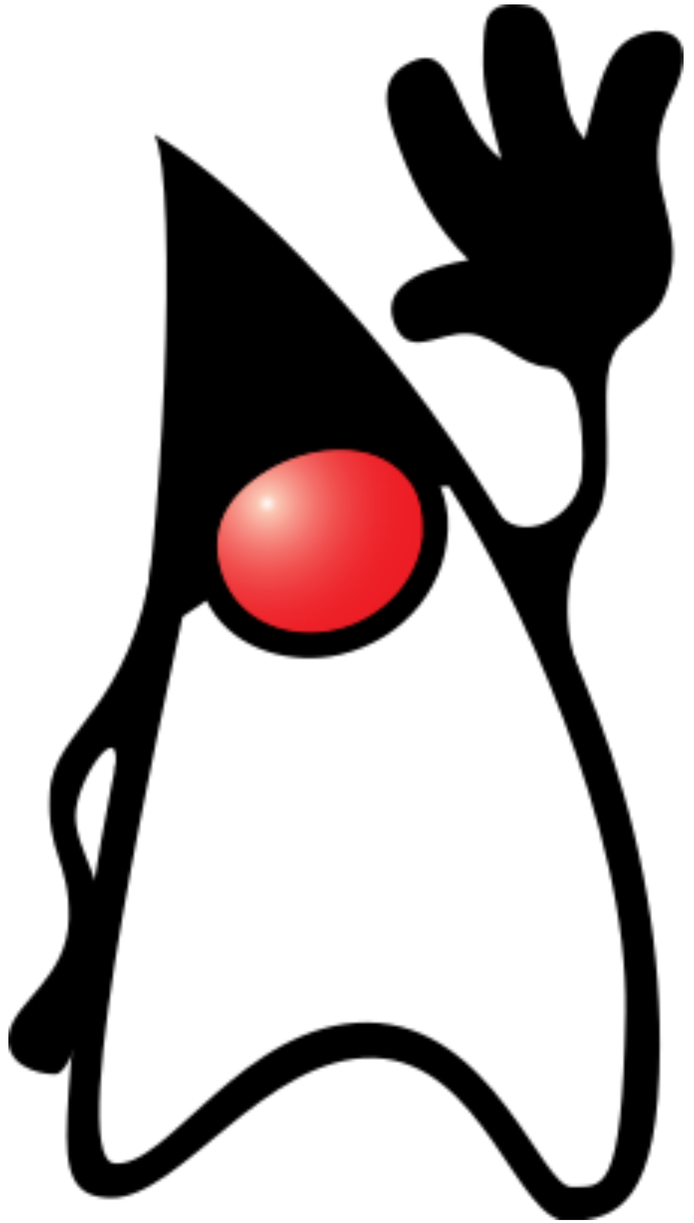
Dead simple cross-compilation

First impressions are *really* important

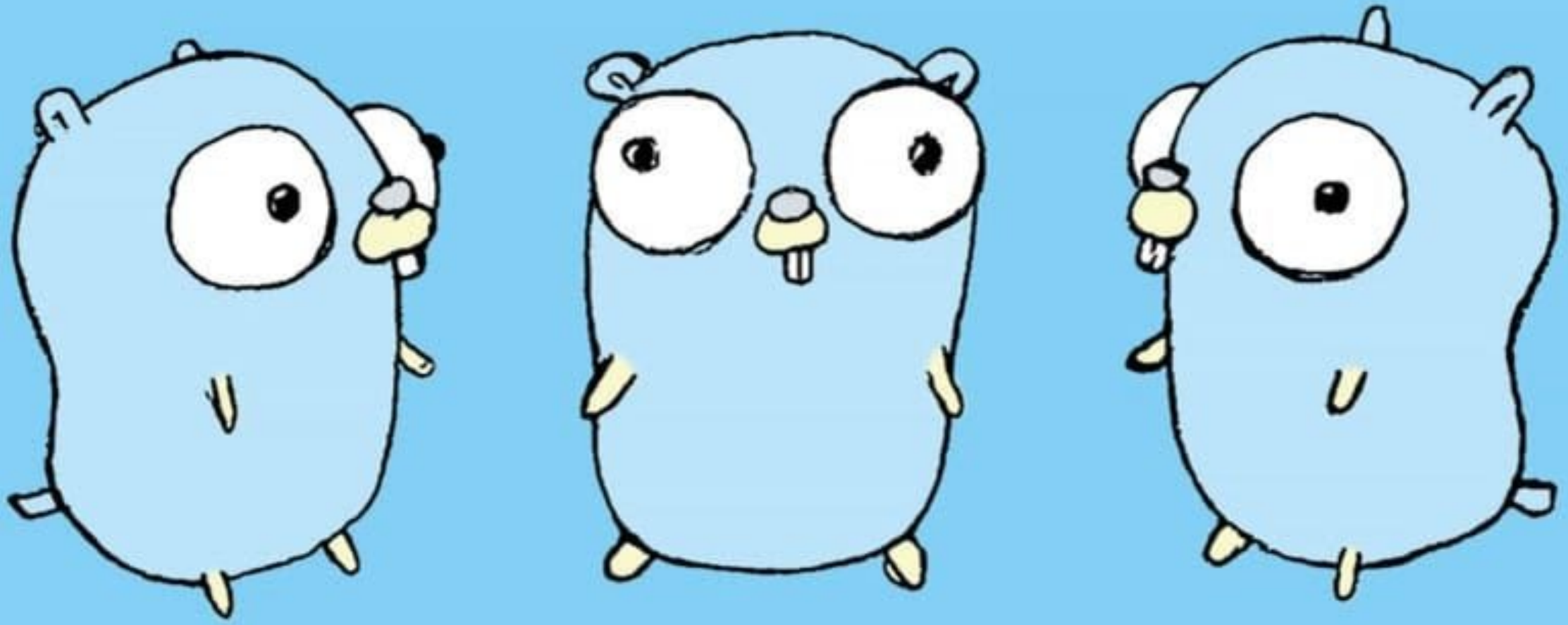
Make it as simple as humanly possible for someone to try your project.

The Gopher

Image: @egonelbre



© Sun Microsystems Inc. BSD Licence



The Go Gopher by Renee French, CC BY 3.0

Gophers reimagine their relationship with the language via their art

@egonelbre

現場で使える
実践テクニック

みんなのGO言語

松木雅幸
mattn
藤原俊一郎
中島大一
牧 大輔
鈴木 健太

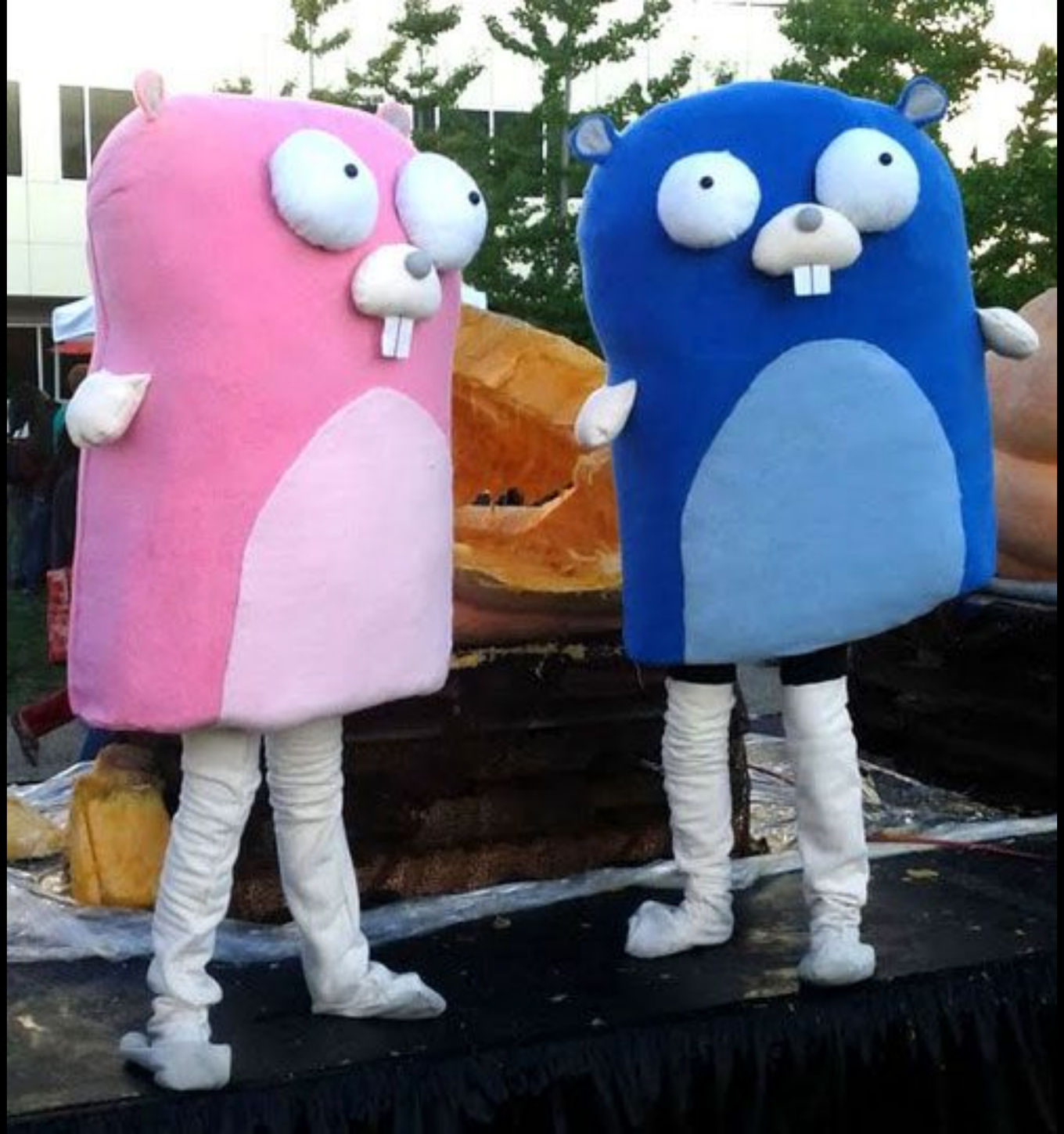
チーム開発のはじめ方
マルチプラットフォームへの対応
アプリ作成のテクニック
コマンドラインツールの作成
reflect／テストツール

技術評論社

Image: Erick Zelaya

image: @ashleymcnamara CC BY-NC-SA 4.0

Image: Luit van Drongelen



Contribution is more than just code

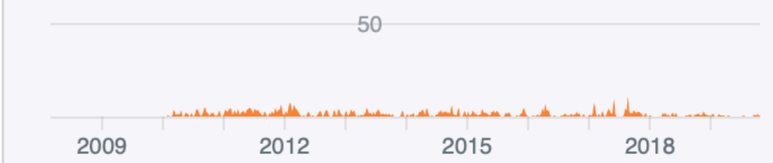
“The actual company values, as opposed to the nice-sounding values, are shown by who gets rewarded, promoted or let go.”

–Patty McCord

If you define merit in your project in terms of who's cut the most code then you explicitly exclude other forms of contribution

github.com/golang/go/graphs/contributors

#13 **alexbrainman**
753 commits 43,141 ++ 23,485 --

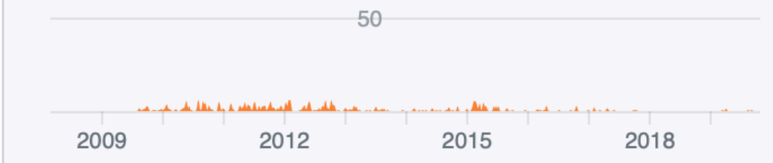


#14 **agl**
523 commits 102,427 ++ 44,044 --

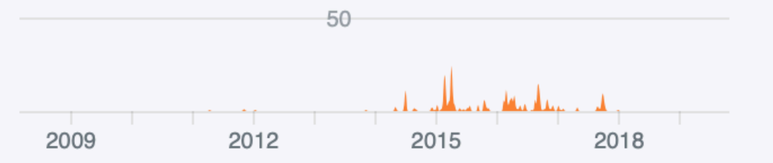
#15 **ken**
508 commits 71,631 ++ 24,971 --

#16 **davecheney**
476 commits 22,309 ++ 14,543 --

#17 **nigeltao**
441 commits 86,398 ++ 75,049 --



#18 **crawshaw**
392 commits 28,571 ++ 16,343 --



Use internal packages to reduce your public API surface

In the beginning, before the `go` tool, before Go 1.0, the Go distribution stored the standard library in a subdirectory called `pkg/` and the commands which built upon it in `cmd/`. This wasn't so much a deliberate taxonomy but a by product of the original `make` based build system. In September 2014, the Go distribution dropped the `pkg/` subdirectory, but then this tribal knowledge had set root in large Go projects and continues to this day.

I tend to view empty directories inside a Go project with suspicion. Often they are a hint that the module's author may be trying to create a taxonomy of packages rather than ensuring each package's name, and thus its enclosing directory, uniquely describes its purpose. While the symmetry with `cmd/` for `package main` commands is appealing, a directory that exists only to hold other packages is a potential design smell.

CATEGORIES

Economy
Go
Hardware Hacking
History
Internets of interest
Photography
Programming
Retrochallenge
Small ideas
Uncategorized
Useless Trivia

There are many ways to contribute to an open source project, make sure you recognise and reward them equally.

A functioning job market

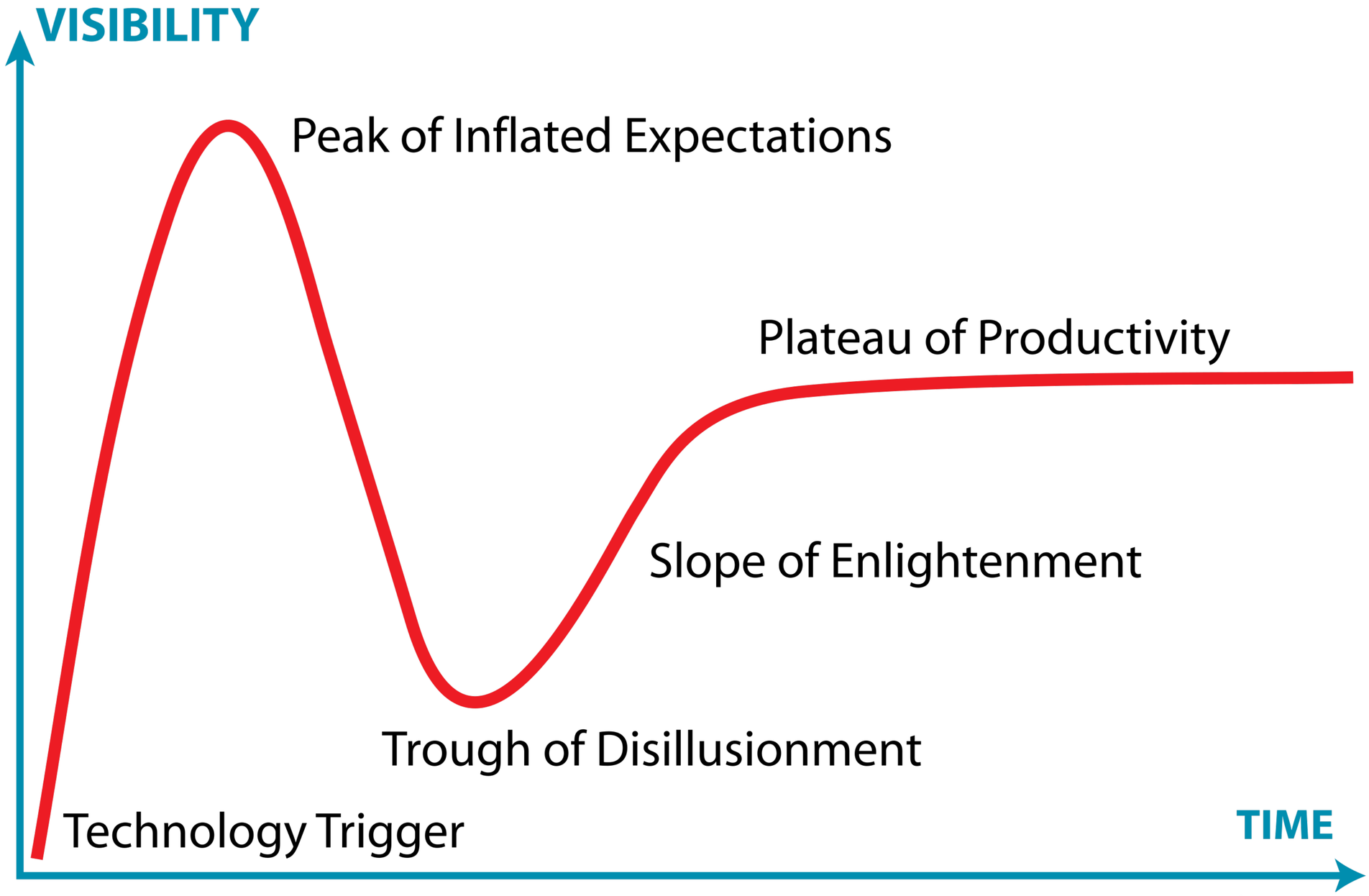


image: wikipedia

Choosing to invest you career in a new technology implicitly means letting your current skills lapse

Go got lucky

Just write a hit song

Codes of conduct

Codes of conduct are just words on paper, but they're *important* words.

Go waited too long to adopt a code of conduct

If you plan to build a community around your project, have a code of conduct in place from the start.

Open governance and succession

The source code of Go may be open, but membership of its governing body is not.

It is not enough for a project to be open source, it must also embrace an open governance model

First impressions matter

Contribution is more that code

Functional job market is key

Write a code of conduct, enforce it

Adopt open governance and plan for sucession