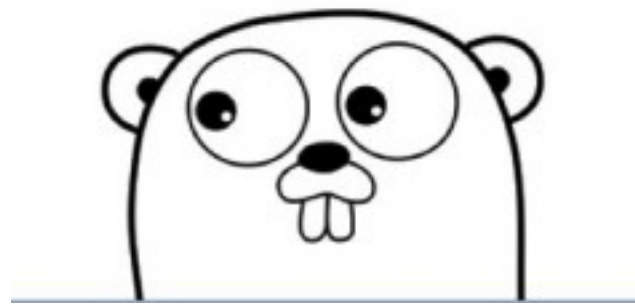




COINVOICE

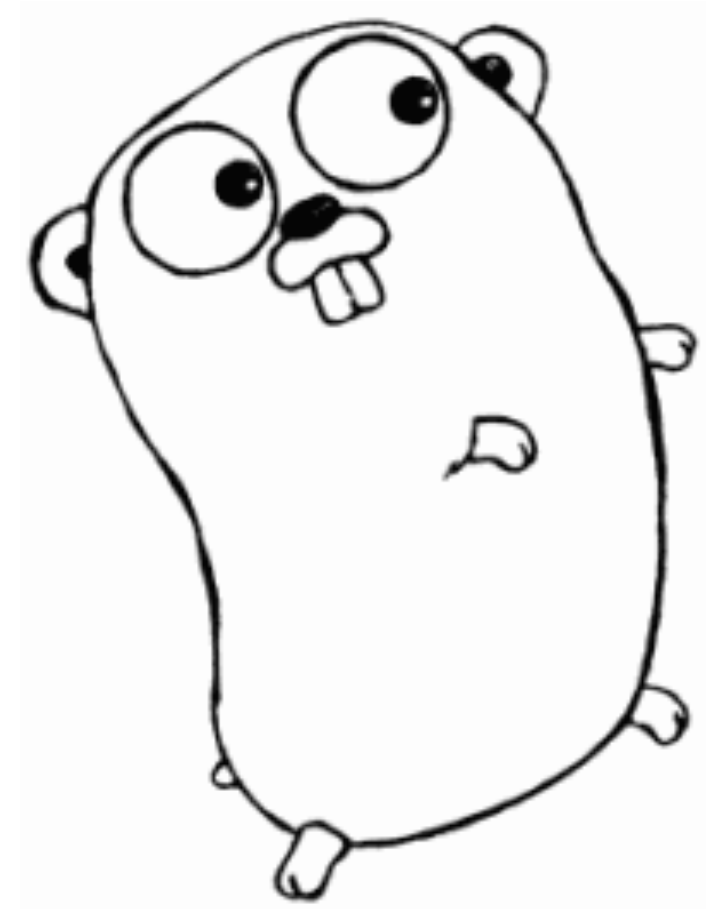


Go for business

Marco Peereboom, CTO

Agenda

- Introduction
- Go at Conformal Systems
- Switching a business from \$LANG
- Economy of Go
- The myth of Object Orientation
- Questions?



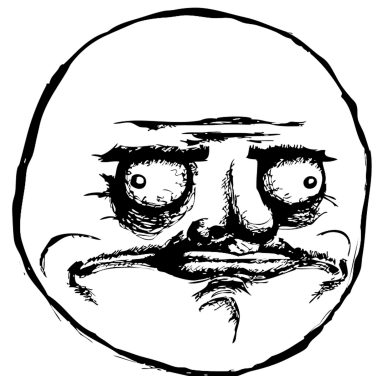
Introduction

- Marco Peereboom
 - CTO
 - 20+ years of industry experience
- Conformal Systems, LLC
 - Fanatical privacy advocates with a crypto problem
 - Coinvoice
 - Cyphertite
 - Alternative bitcoin full node implementation



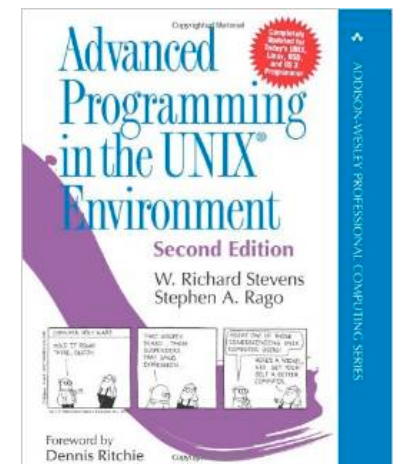
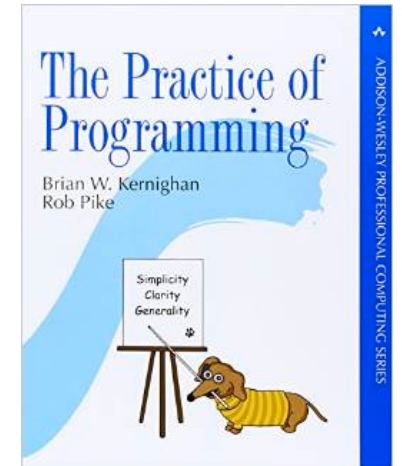
Go at Conformal Systems

- Bunch of hardcore C guys working on Cyphertite
- Scattered use of Python and PHP
- Picked up a book on Go, they had me at defer
- Go is simply put, a better systems programming language and we get to do websites in it too
- Productivity shot up and complexity went down



Switching from C

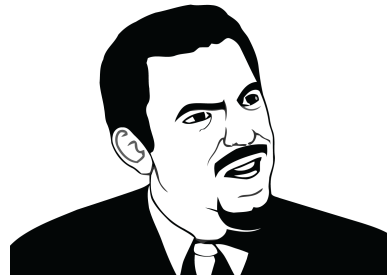
- People magically received Go books in their mail
- Productively developing in less than a week
- Pretty darn good at 4 weeks
- 2 years later the consensus still is: “why didn’t we switch earlier?”
- Respected contributors to the Go community



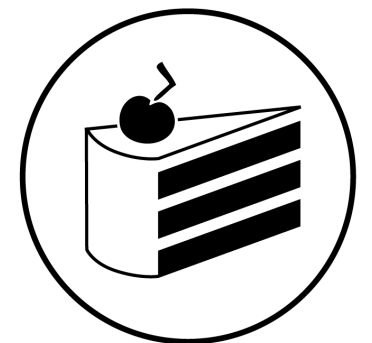
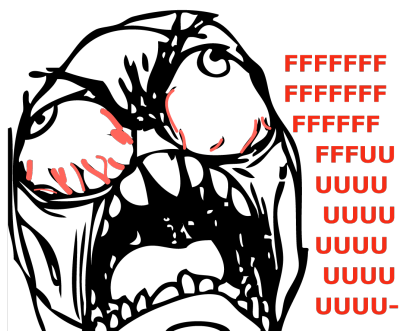
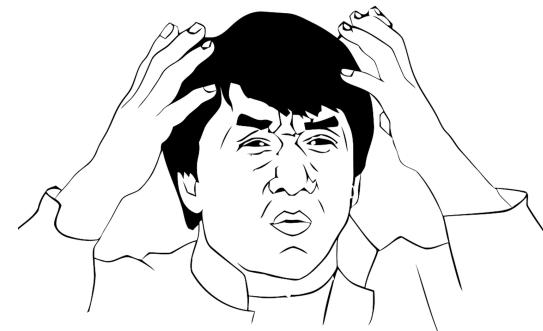
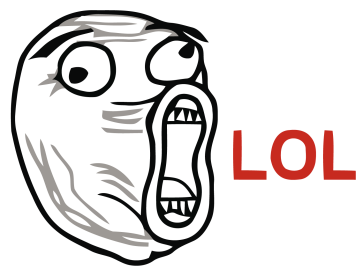
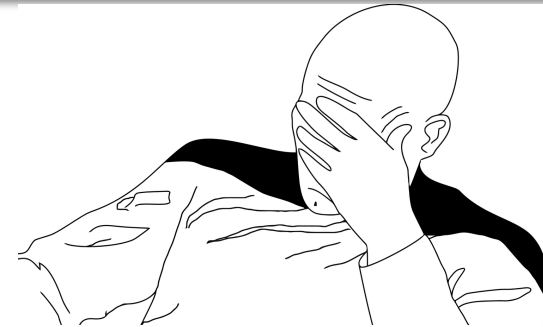
Economy of Go

- The Go developers reinvented the development process
 - obsoleted “worthless” crap like: Makefiles, Doxygen, Ant, etc
 - free stuff: unit test, performance and documentation framework, build and packaging systems
 - eliminates entire teams
- Code is roughly ~25% the size and takes 1/10 of the time to write
 - much minutiae and boilerplate simply removed
 - concurrency, unwinding, garbage collection, %v (reflection), channels, duck typing, etc

Object Orientation



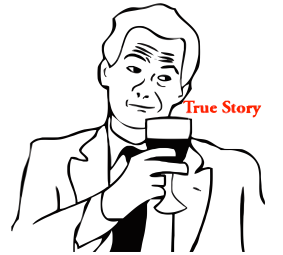
- Standard OO drive!
- You can share code and it is reusable!
- Look at all these design patterns!
- You only write it once!
- Everything is an object!
- Generics allows us to sort bananacars!
- Exceptions are enterprise quality!
- It totally has IDEs!



the cake is a lie !

The so called real world

- Complexity is the enemy of good
 - Refactoring objects usually translates into work for everyone
 - People spend vast amounts of time learning failed paradigms and are unwilling/incapable of admitting that mistake (and their livelihood may depend on it)
- Perfect is the enemy of done
 - Lots of time spent up front to make things “perfect”
 - Always gets abandoned because it can’t be perfect and that silly release thing
- Less is way more
 - Give clever people clever tools and they will do clever things with them
 - Take those toys away and make them use good primitives and watch code be better



Questions?

<https://conformal.com/>

<https://coinvoice.com/>

<https://github.com/conformal>

