RELEASE STRATEGY

Mahmoud Masih Tehrani
TEHLUG 2017 Sep 7

MAHMOUD MASIH TEHRANI

- Php & Golang developer
- about 10 years use GNU/Linux
- I love
  Nature ,Sport ,Theatre ,Free
  Software & Programing
- Username : mahm0ud22
- Email:
  mahmud.tehrani@gmail.com

PLEASE READ!

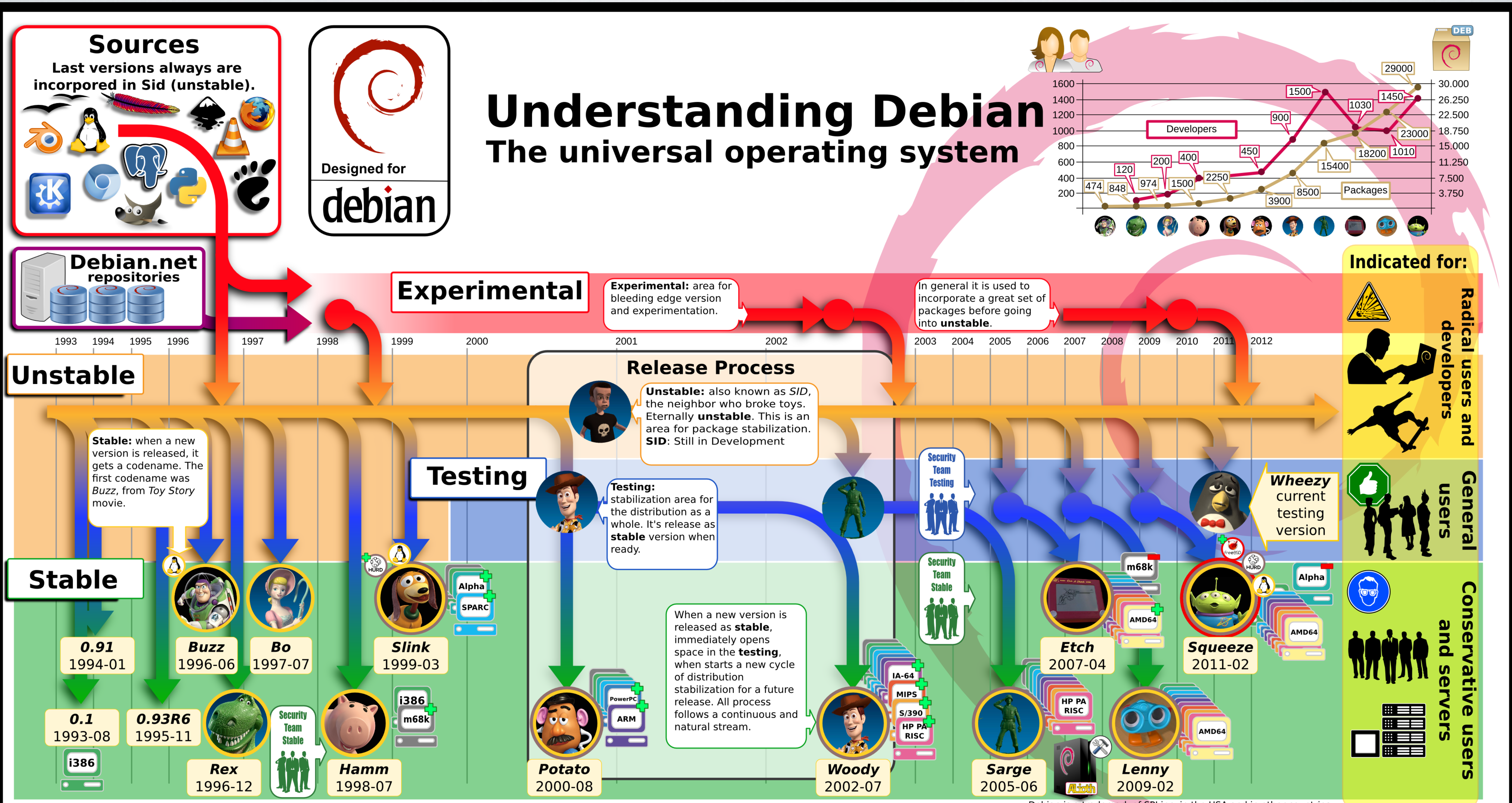
- Please read agile team workflow
- https://www.slideshare.net/mahm0ud22/agile-team-workflow

RELEASE

- Debian
- Arch
- Ubuntu

Understanding Debian

The universal operating system



Sources

Last versions always are incorporated in Sid (unstable).

Experimental: area for bleeding edge version and experimentation.

In general it is used to incorporate a great set of packages before going into **unstable**.

Release Process

Unstable: also known as *SID*, the neighbor who broke toys. Eternally **unstable**. This is an area for package stabilization. **SID**: Still in Development

Stable: when a new version is released, it gets a codename. The first codename was *Buzz*, from *Toy Story* movie.

Testing: stabilization area for the distribution as a whole. It's release as **stable** version when ready.

When a new version is released as **stable**, immediately opens space in the **testing**, when starts a new cycle of distribution stabilization for a future release. All process follows a continuous and natural stream.

Wheezy current testing version

DEBIAN

- Sid
- Unstable
- Testing
- Stable
- Archive

ARCH LINUX

Arch Linux uses a **rolling release model**, such that a regular system update is all that is needed to obtain the latest Arch software; the installation images released by the Arch team are simply up-to-date snapshots of the main system components.

ROLLING RELEASE

In software development, a **rolling release**, **rolling update**, or **continuous delivery** is the concept of frequently delivering updates to applications. This is in contrast to a *standard* or *point release* development model which uses software versions that must be reinstalled over the previous version.

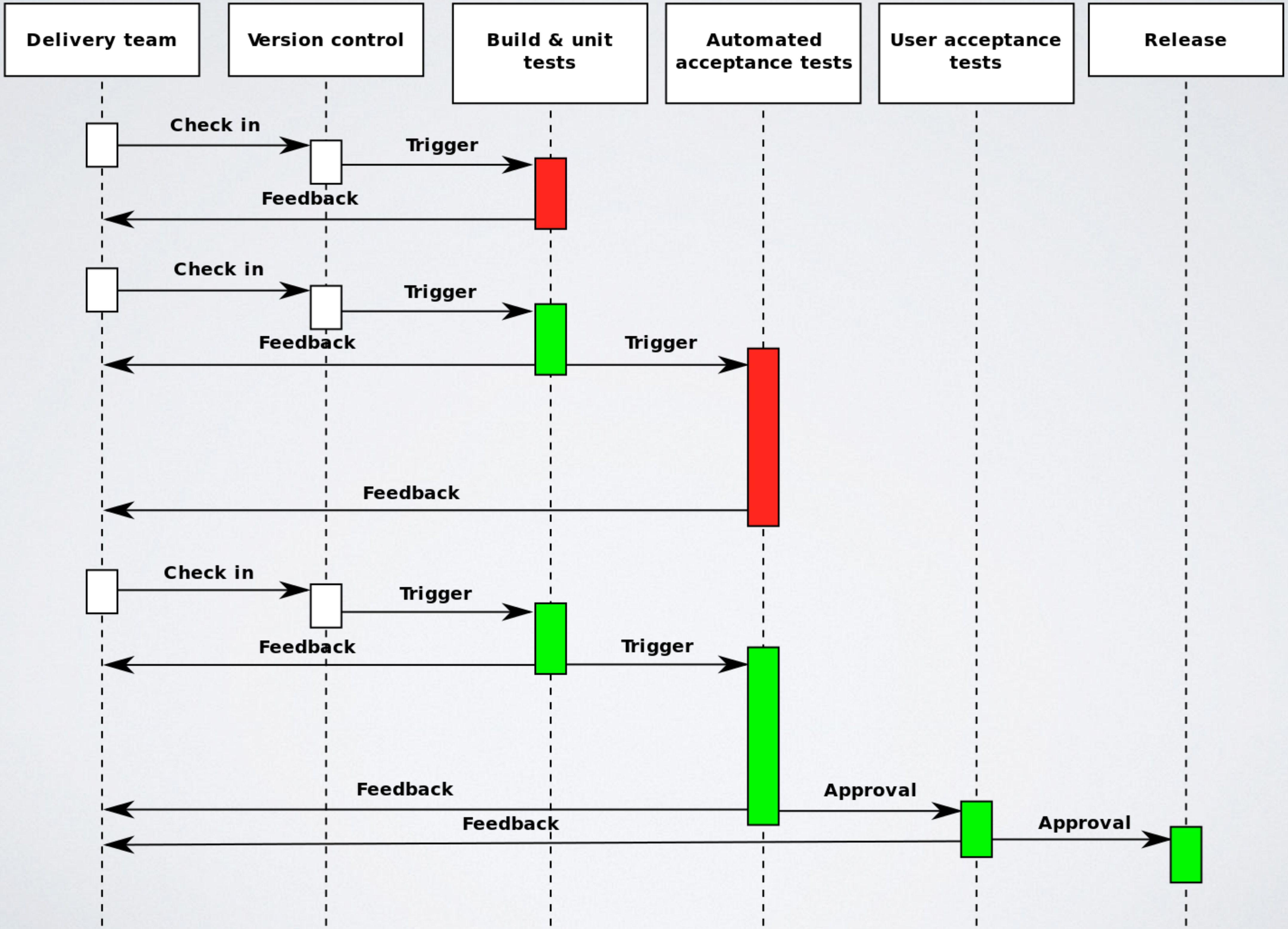
Rolling release development models are one of many types of software release life cycles. Although a rolling release model can be used in the development of any piece or collection of software, it is often seen in use by Linux distributions, see rolling distribution.

A rolling release is typically implemented using small and frequent updates. However, simply having updates does not automatically mean that a piece of software is using a rolling release cycle; for this, the philosophy of developers **must be to work with one code branch, versus discrete versions.**

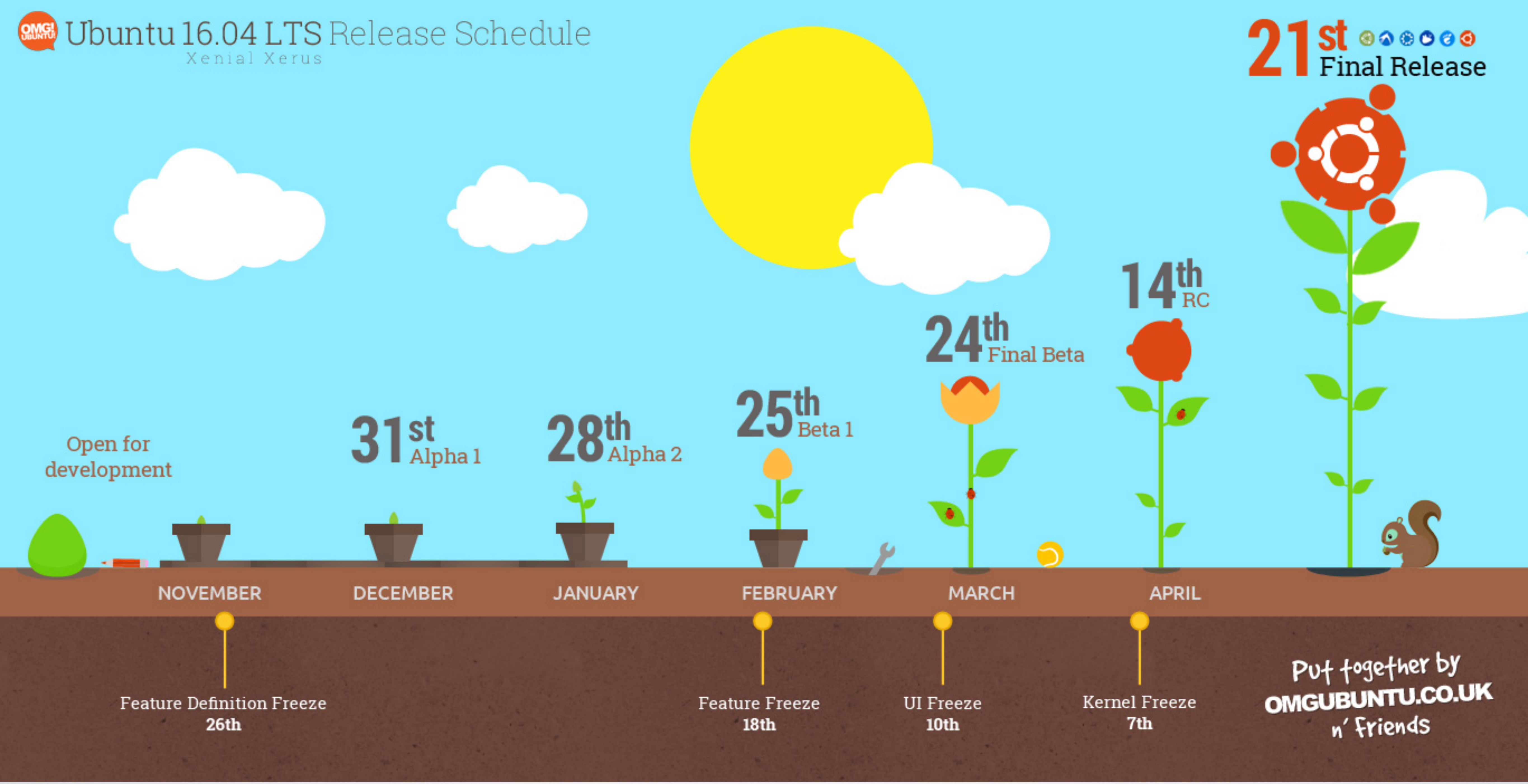
CONTINUOUS DELIVERY

Continuous delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time.[1] It aims at building, testing, and releasing software faster and more frequently. The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production. A straightforward and repeatable deployment process is important for continuous delivery.

CONTINUOUS DELIVERY



UBUNTU


OMG! UBUNTU!
Ubuntu 16.04 LTS Release Schedule
Xenial Xerus
21st Final Release
14th RC
24th Final Beta
25th Beta 1
28th Alpha 2
31st Alpha 1
Open for development
NOVEMBER
DECEMBER
JANUARY
FEBRUARY
MARCH
APRIL
Feature Definition Freeze 26th
Feature Freeze 18th
UI Freeze 10th
Kernel Freeze 7th
Put together by OMGUBUNTU.CO.UK n' Friends

SOFTWARE RELEASE LIFE CYCLE

- Pre-Alpha
- Alpha
- Beta
- RC (Release Condidate)

PRE-ALPHA

Pre-alpha refers to all activities performed during the software project before formal testing. These activities can include requirements analysis, software design, software development, and unit testing. In typical open source development, there are several types of pre-alpha versions. ***Milestone*** **versions include specific sets of functions and are released as soon as the functionality is complete.**

ALPHA

The alpha phase of the release life cycle is the first phase to begin **software testing** (alpha is the first letter of the Greek alphabet, used as the number 1). In this phase, developers generally test the software using **white-box techniques**. Additional validation is then performed using **black-box or gray-box** techniques, by another testing team. Moving to black-box testing inside the organization is known as *alpha release*.[1]

Alpha software can be unstable and could cause crashes or data loss. Alpha software may not contain all of the features that are planned for the final version.[2] In general, external availability of alpha software is uncommon in proprietary software, while open source software often has publicly available alpha versions. **The alpha phase usually ends with a feature freeze, indicating that no more features will be added to the software. At this time, the software is said to be feature complete.**

BETA

Beta phase generally begins when the software is feature complete but likely to contain a number of **known or unknown bugs**.[4] Software in the beta phase will generally **have many more bugs** in it than completed software, as well as **speed/ performance issues** and may still cause crashes or data loss.The focus of beta testing is reducing impacts to users, often incorporating usability testing.The process of delivering a beta version to the users is called *beta release* and this is typically the first time that the software is available outside of the organization that developed it. Beta version software is often useful for demonstrations and previews within an organization and to prospective customers. Some developers refer to this stage as a *preview*, *preview release*, *prototype*, *technical preview* / *technology preview* (*TP*),[5] or *early access*. Some software is kept in perpetual beta, where new features and functionality are continually added to the software without establishing a firm "final" release.

Beta testers **are people who actively report issues of beta software. They are usually customers or representatives of prospective customers of the organization that develops the software.** Beta testers tend to volunteer their services free of charge but often receive versions of the product they test, discounts on the release version, or other incentives.

As the Internet has facilitated rapid and inexpensive distribution of software, companies have begun to take a looser approach to use of the word "beta".[6] In February 2005, ZDNet published an article about the recent phenomenon of a beta version often staying for years and being used as if it were in production level, disparagingly called "perpetual beta". It noted that Gmail and Google News, for example, had been in beta for a long period of time and were not expected to drop the beta status despite the fact that they were widely used; however, Google News did leave beta in January 2006, followed by Google Apps, including Gmail, in July 2009.[7] This technique may allow a developer to delay offering full support and responsibility for remaining issues. In the context of Web 2.0,

OPEN AND CLOSED BETA

Developers release either a *closed beta* also called *private beta* or an *open beta* also called *public beta*; closed beta versions are released to a restricted group of individuals for a user test by invitation, while open beta testers are from a larger group, or anyone interested. **Private beta** could be suitable for the software that is capable to deliver value, but is not ready to be used by everyone either due to scaling issues, lack of documentation or still missing vital features. The testers report any bugs that they find, and sometimes suggest additional features they think should be available in the final version. Examples of a major **public beta** test include the following:

- Early customers purchased a "pioneer edition" of the WordVision word processor for the IBM PC for $49.95. In 1984, Stephen Manes wrote that "in a brilliant marketing coup, Bruce and James Program Publishers managed to get people to *pay* for the privilege of testing the product."[9]
- In September 2000 a *boxed version* of Apple's Mac OS X Public Beta operating system was released.[10]
- Microsoft's release of *community technology previews* (*CTP*s) for Windows Vista, between September 2005 and May 2006.[11]
- Throughout 2009 to 2011, Minecraft was in public beta.
- On December 29, 2014, all owners of *Halo: The Master Chief Collection* for the Xbox One were able to download and play the Beta of *Halo 5: Guardians* for free through January 18, 2015. Users of the Beta were reminded via an in-game popup that the

RELEASE CANDIDATE

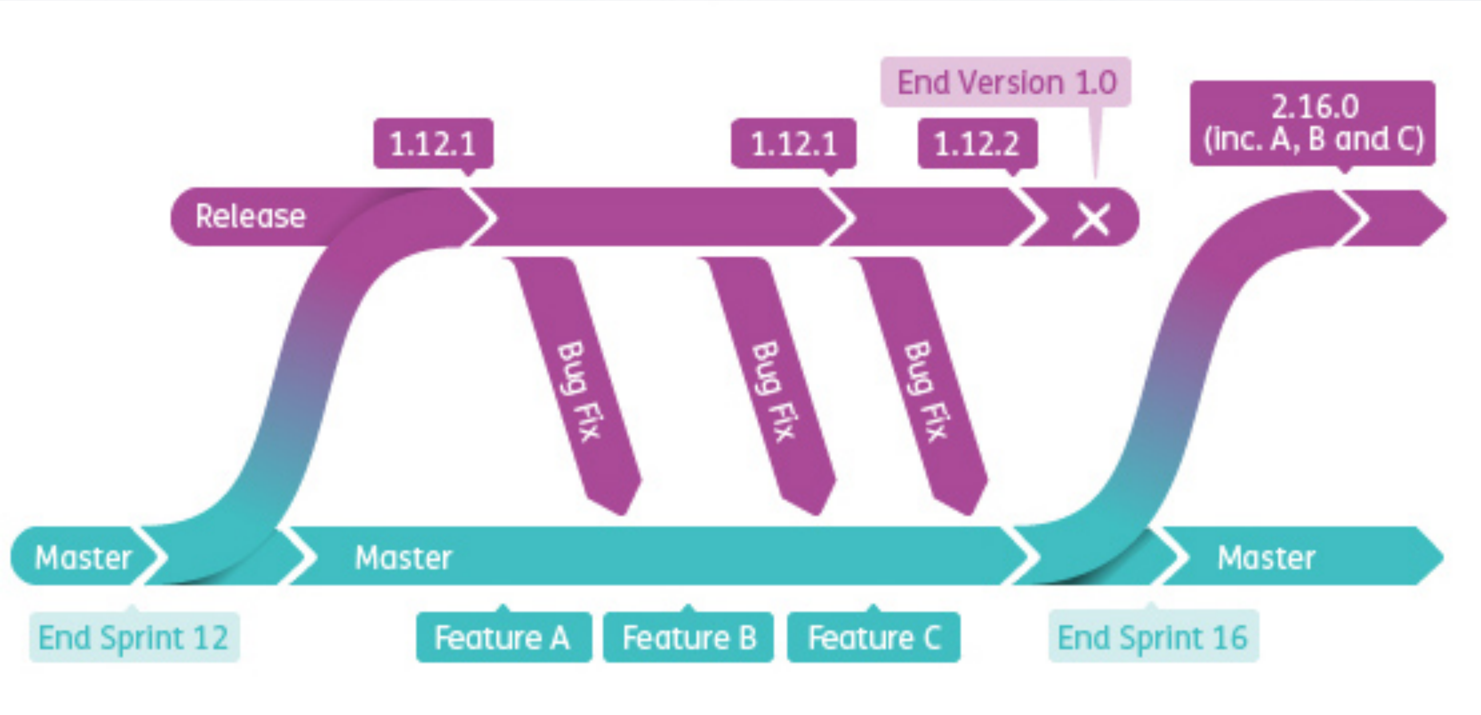
A *release candidate* (*RC*), also known as "going silver", is a beta version with **potential to be a final product, which is ready to release unless significant bugs emerge.** In this stage of product stabilization, all product features have been designed, coded and tested through one or more beta cycles with no known showstopper-class bugs. A release is called *code complete* when the development team agrees that no entirely new source code will be added to this release. There could still be source code changes to fix defects, changes to documentation and data files, and peripheral code for test cases or utilities. Beta testers, if privately selected, will often be credited for using the release candidate as though it were a finished product. Beta testing is conducted in a client's or customer's location and to test the software from a user's perspective.

FREEZE

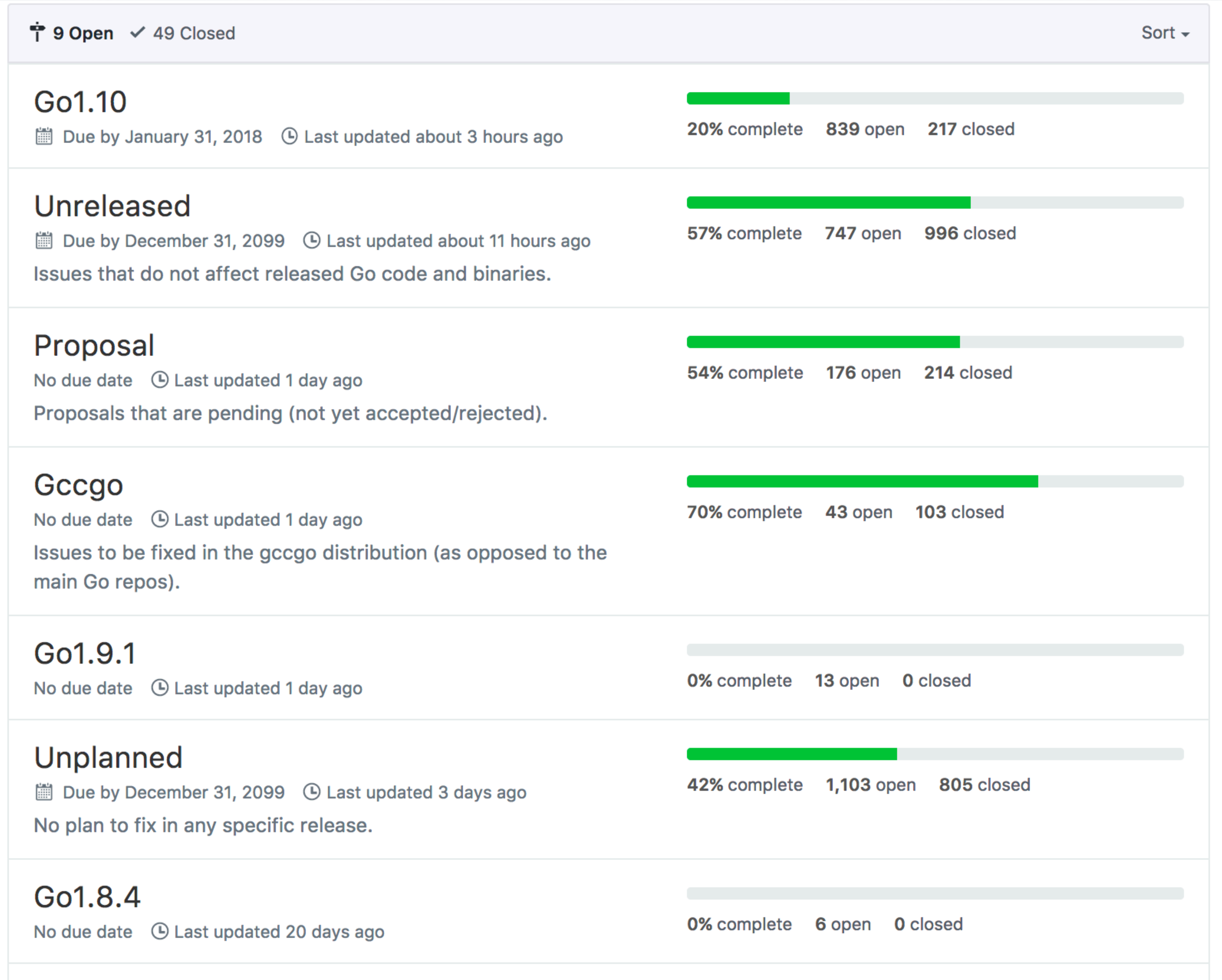
<code_freeze>101</code_freeze>

RELEASE

- Sid
- Testing
- Stable



GOLANG



9 Open ✓ 49 Closed

Sort

Go1.10

Due by January 31, 2018 Last updated about 3 hours ago

20% complete 839 open 217 closed

Unreleased

Due by December 31, 2099 Last updated about 11 hours ago

Issues that do not affect released Go code and binaries.

57% complete 747 open 996 closed

Proposal

No due date Last updated 1 day ago

Proposals that are pending (not yet accepted/rejected).

54% complete 176 open 214 closed

Gccgo

No due date Last updated 1 day ago

Issues to be fixed in the gccgo distribution (as opposed to the main Go repos).

70% complete 43 open 103 closed

Go1.9.1

No due date Last updated 1 day ago

0% complete 13 open 0 closed

Unplanned

Due by December 31, 2099 Last updated 3 days ago

No plan to fix in any specific release.

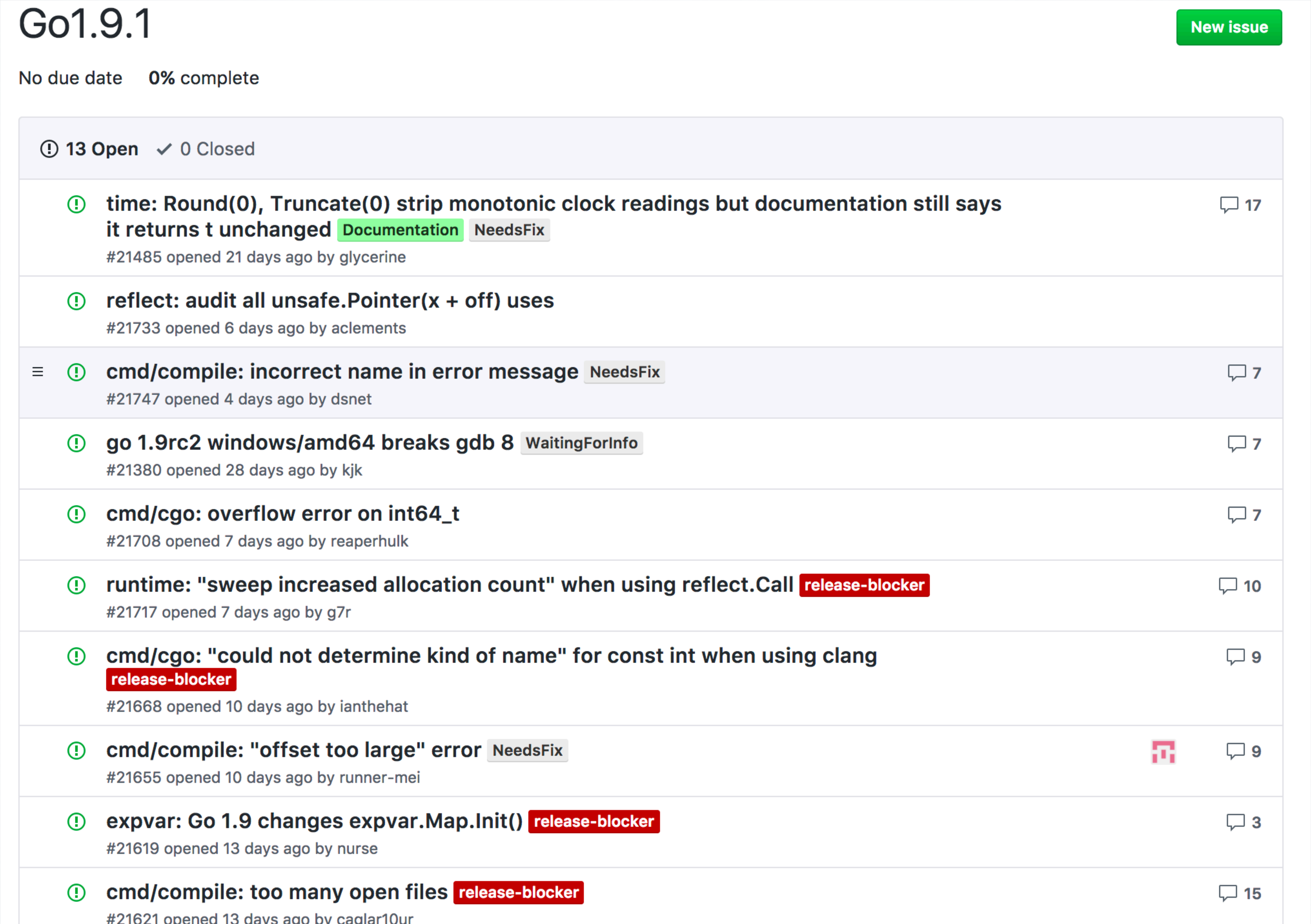
42% complete 1,103 open 805 closed

Go1.8.4

No due date Last updated 20 days ago

0% complete 6 open 0 closed

GOLANG



Go1.9.1

No due date **0%** complete

13 Open ✓ 0 Closed

time: Round(0), Truncate(0) strip monotonic clock readings but documentation still says it returns t unchanged Documentation NeedsFix 17
#21485 opened 21 days ago by glycerine

reflect: audit all unsafe.Pointer(x + off) uses
#21733 opened 6 days ago by aclements

cmd/compile: incorrect name in error message NeedsFix 7
#21747 opened 4 days ago by dsnet

go 1.9rc2 windows/amd64 breaks gdb 8 WaitingForInfo 7
#21380 opened 28 days ago by kjk

cmd/cgo: overflow error on int64_t 7
#21708 opened 7 days ago by reaperhulk

runtime: "sweep increased allocation count" when using reflect.Call release-blocker 10
#21717 opened 7 days ago by g7r

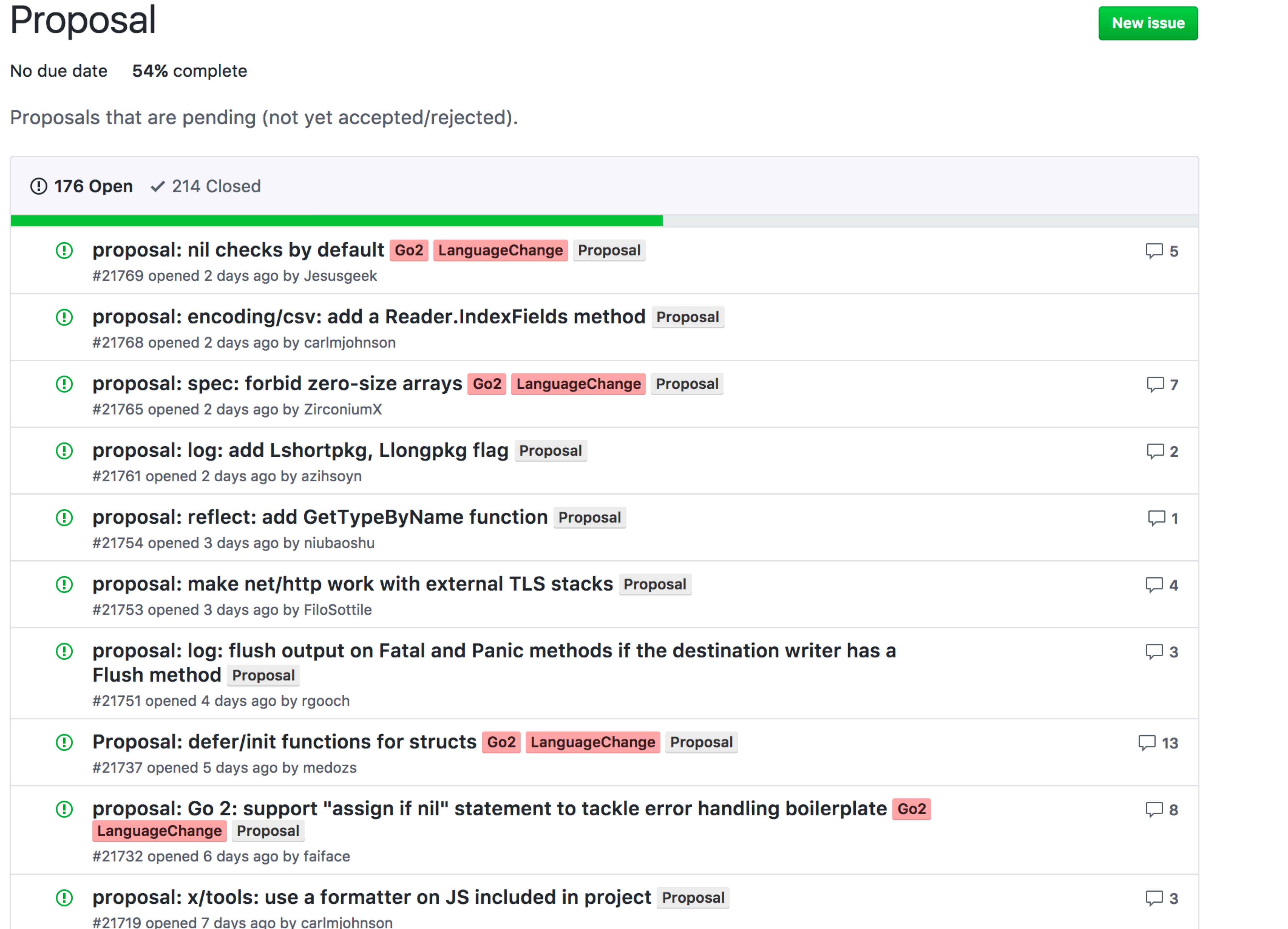
cmd/cgo: "could not determine kind of name" for const int when using clang release-blocker 9
#21668 opened 10 days ago by ianthehat

cmd/compile: "offset too large" error NeedsFix 9
#21655 opened 10 days ago by runner-mei

expvar: Go 1.9 changes expvar.Map.Init() release-blocker 3
#21619 opened 13 days ago by nurse

cmd/compile: too many open files release-blocker 15
#21621 opened 13 days ago by caglar10ur

GOLANG



Proposal

New issue

No due date **54%** complete

Proposals that are pending (not yet accepted/rejected).

176 Open ✓ 214 Closed

proposal: nil checks by default Go2 LanguageChange Proposal — 5
#21769 opened 2 days ago by Jesusgeek

proposal: encoding/csv: add a Reader.IndexFields method Proposal
#21768 opened 2 days ago by carlmjohnson

proposal: spec: forbid zero-size arrays Go2 LanguageChange Proposal — 7
#21765 opened 2 days ago by ZirconiumX

proposal: log: add Lshortpkg, Llongpkg flag Proposal — 2
#21761 opened 2 days ago by azihsoyn

proposal: reflect: add GetTypeByName function Proposal — 1
#21754 opened 3 days ago by niubaoshu

proposal: make net/http work with external TLS stacks Proposal — 4
#21753 opened 3 days ago by FiloSottile

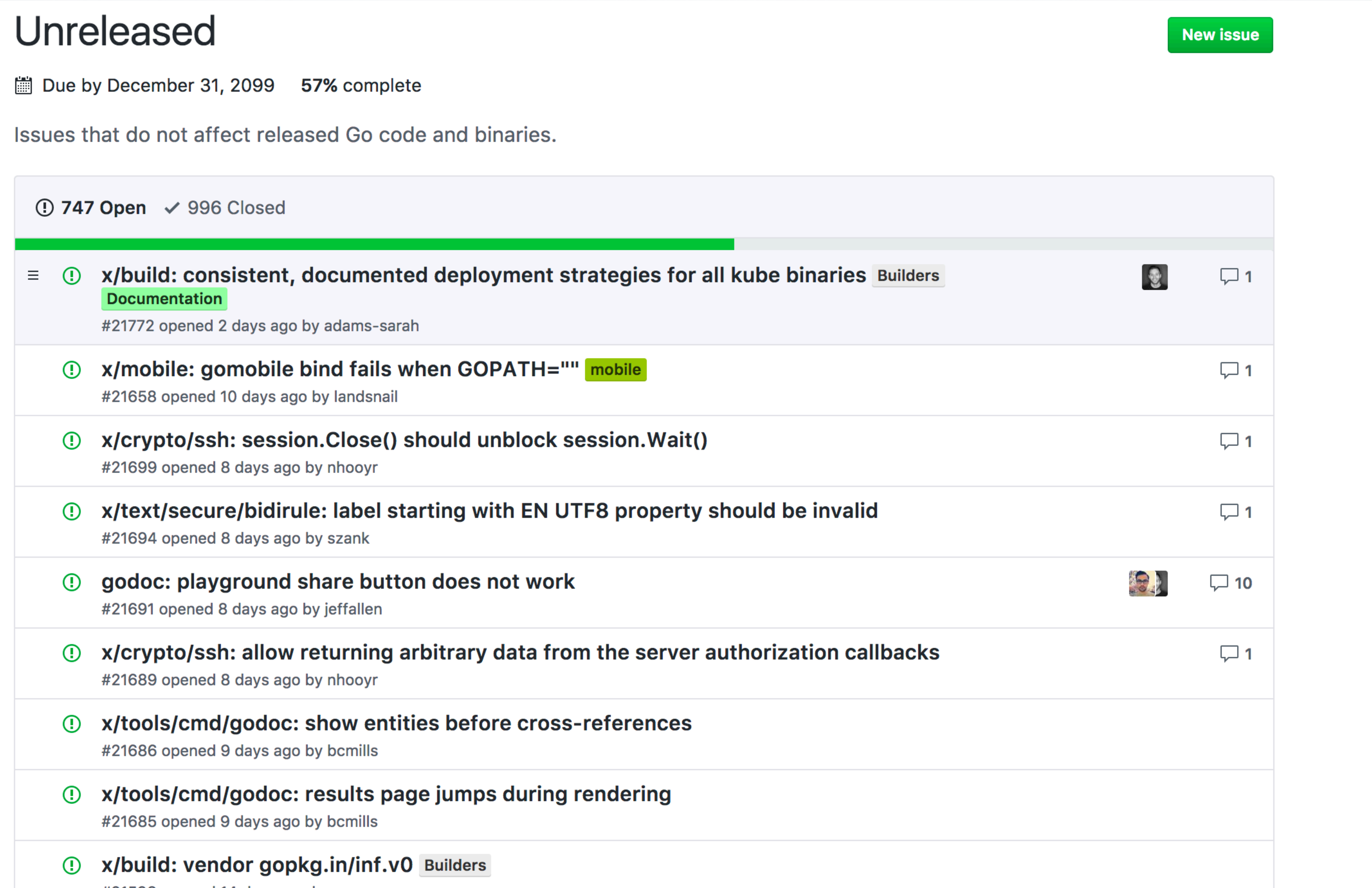
proposal: log: flush output on Fatal and Panic methods if the destination writer has a Flush method Proposal — 3
#21751 opened 4 days ago by rgooch

Proposal: defer/init functions for structs Go2 LanguageChange Proposal — 13
#21737 opened 5 days ago by medozs

proposal: Go 2: support "assign if nil" statement to tackle error handling boilerplate Go2 LanguageChange Proposal — 8
#21732 opened 6 days ago by faiface

proposal: x/tools: use a formatter on JS included in project Proposal — 3
#21719 opened 7 days ago by carlmjohnson

GOLANG



Unreleased

New issue

Due by December 31, 2099   **57%** complete

Issues that do not affect released Go code and binaries.

ⓘ **747 Open**   ✓ 996 Closed

ⓘ **x/build: consistent, documented deployment strategies for all kube binaries** Builders Documentation
#21772 opened 2 days ago by adams-sarah — 1

ⓘ **x/mobile: gomobile bind fails when GOPATH=""** mobile
#21658 opened 10 days ago by landsnail — 1

ⓘ **x/crypto/ssh: session.Close() should unblock session.Wait()**
#21699 opened 8 days ago by nhooyr — 1

ⓘ **x/text/secure/bidirule: label starting with EN UTF8 property should be invalid**
#21694 opened 8 days ago by szank — 1

ⓘ **godoc: playground share button does not work**
#21691 opened 8 days ago by jeffallen — 10

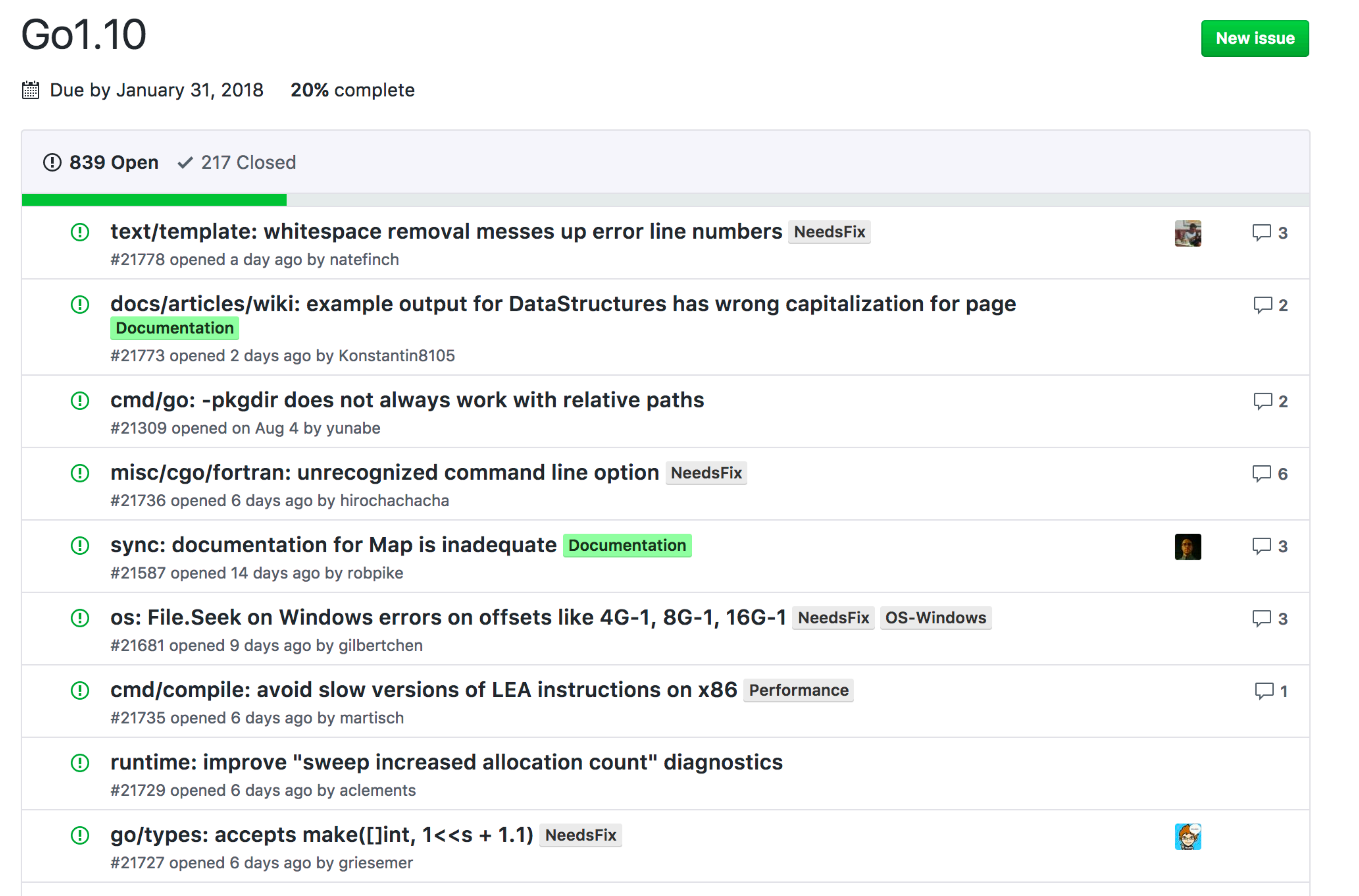
ⓘ **x/crypto/ssh: allow returning arbitrary data from the server authorization callbacks**
#21689 opened 8 days ago by nhooyr — 1

ⓘ **x/tools/cmd/godoc: show entities before cross-references**
#21686 opened 9 days ago by bcmills

ⓘ **x/tools/cmd/godoc: results page jumps during rendering**
#21685 opened 9 days ago by bcmills

ⓘ **x/build: vendor gopkg.in/inf.v0** Builders

GOLANG



Go1.10

Due by January 31, 2018 **20%** complete

839 Open ✓ 217 Closed

text/template: whitespace removal messes up error line numbers NeedsFix 🗨 3
#21778 opened a day ago by natefinch

docs/articles/wiki: example output for DataStructures has wrong capitalization for page Documentation 🗨 2
#21773 opened 2 days ago by Konstantin8105

cmd/go: -pkgdir does not always work with relative paths 🗨 2
#21309 opened on Aug 4 by yunabe

misc/cgo/fortran: unrecognized command line option NeedsFix 🗨 6
#21736 opened 6 days ago by hirochachacha

sync: documentation for Map is inadequate Documentation 🗨 3
#21587 opened 14 days ago by robpike

os: File.Seek on Windows errors on offsets like 4G-1, 8G-1, 16G-1 NeedsFix OS-Windows 🗨 3
#21681 opened 9 days ago by gilbertchen

cmd/compile: avoid slow versions of LEA instructions on x86 Performance 🗨 1
#21735 opened 6 days ago by martisch

runtime: improve "sweep increased allocation count" diagnostics
#21729 opened 6 days ago by aclements

go/types: accepts make([]int, 1<<s + 1.1) NeedsFix
#21727 opened 6 days ago by griesemer

THANK YOU

- Next Presentation is
- Swift for backend
- Distributed hash table
- Any question?