



# EXPLORING OUR BUGS

**BEN COTTON**

Fedora Program Manager

✉ [bcotton@fedoraproject.org](mailto:bcotton@fedoraproject.org)

🐦 [@funnelfiasco](https://twitter.com/funnelfiasco)

[m] [@funnelfiasco:matrix.org](https://matrix.org/#/room/#funnelfiasco:matrix.org)

# HOUSEKEEPING

Pronouns: he/him

License: CC BY-SA 4.0

Nice things: @FunnelFiasco

Not-nice things: /dev/null

Questions at the end

@FunnelFiasco

CC BY-SA 4.0



# WHAT IS THIS TALK?

CC BY-SA 4.0

## A LOOK AT FEDORA BUGS

- From Fedora Linux 19 through Fedora Linux 34
  - Now (sometimes) includes Rawhide bugs!
- Based on curiosity, not convincing
- Asks more questions than it answers

@FunnelFiasco

CC BY-SA 4.0



I started with F19 because that was the first release with an EOL closure type. It represents an obvious “modern era” of our bugs. F34 is the last EOL release at the time of this presentation, so we stop there.

## NOTES ON RAWHIDE

- Rawhide bugs based on closure date
  - Between N-1 and N branch days
- Not included in all charts, tables, etc

@FunnelFiasco

CC BY-SA 4.0

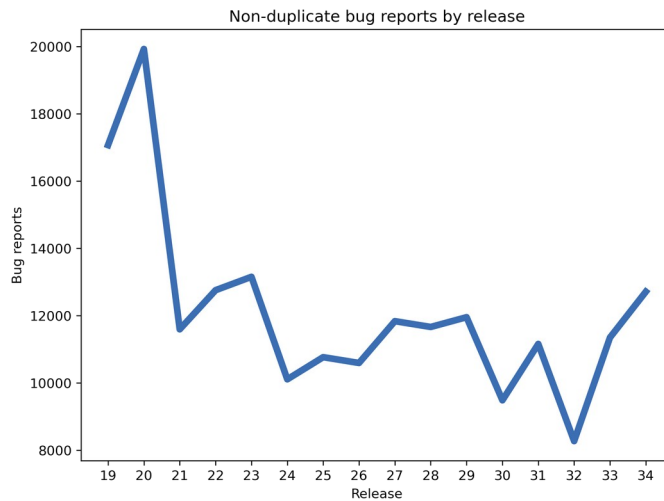


Right now, Rawhide is a separate notebook. As I worked on this talk, I realized there are a lot of places we might want to consider Rawhide bugs alongside released bugs. So I'll merge the notebooks later.

# THE BASICS

CC BY-SA 4.0

## BUG REPORTS PER RELEASE

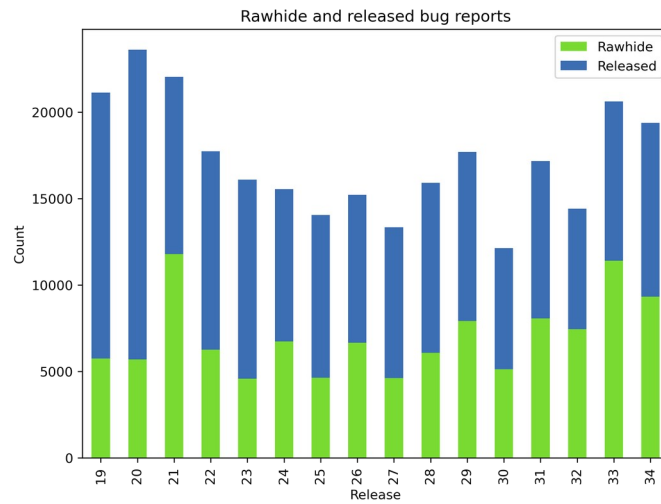


@FunnelFiasco

CC BY-SA 4.0

The obvious place to start is to just look at the number of (non-duplicate) bug reports per release. Karl Fogel says “an accessible bug database is one of the strongest signs that a project should be taken seriously —and the higher the number of bugs in the database, the better the project looks”. The fact that we’ve seen bug reports rise at the same time as we’ve gotten more consistently-good press makes sense.

# PLUS RAWHIDE



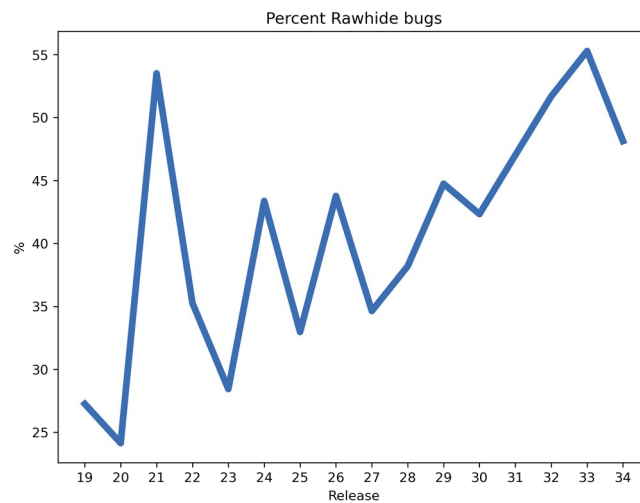
@FunnelFiasco

CC BY-SA 4.0

But how do the Rawhide reports compare to the released?



# RAWHIDE PERCENTAGE



@FunnelFiasco

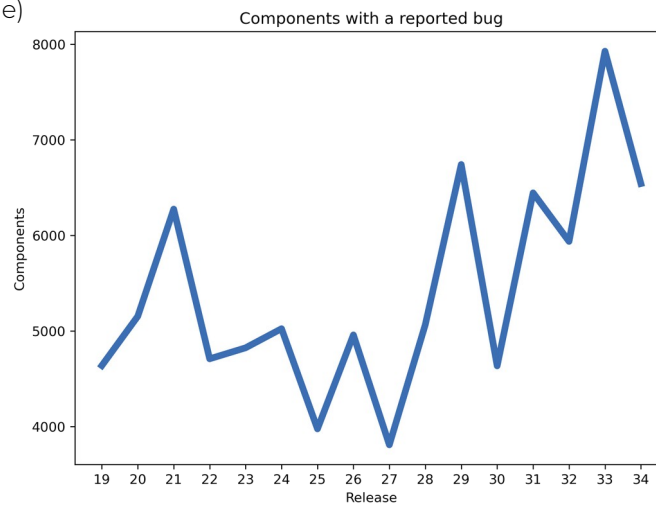
CC BY-SA 4.0



The general trend is for Rawhide to make up a higher percentage of bugs over time. We may be bucking that trend recently.

# COMPONENTS WITH A REPORT

(includes Rawhide)



@FunnelFiasco

CC BY-SA 4.0



# BUGS PER COMPONENT: TOP 10

(includes Rawhide)

Component	Bugs
kernel	11645
selinux-policy	7441
gnome-shell	4296
anaconda	3824
dnf	2662
systemd	2069
firefox	1846
nautilus	1590
xorg-x11-server	1410
PackageKit	1211

@FunnelFiasco

CC BY-SA 4.0



# BUGS PER COMPONENT: BOTTOM 10

(includes Rawhide)

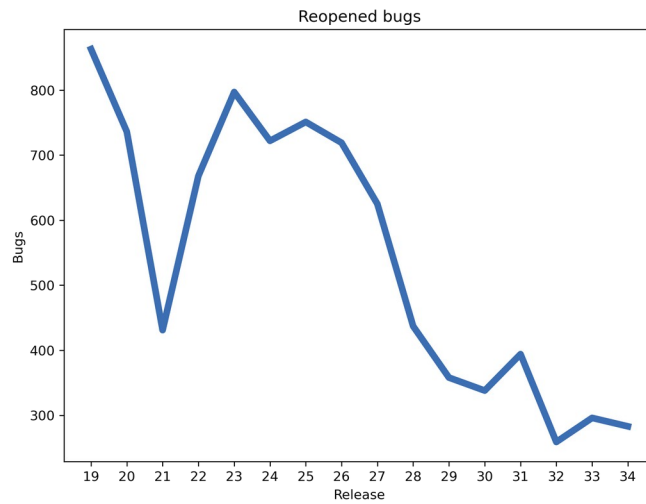
- 22,695 (98.84%) components with < 100 bugs
- 18,894 (82.29%) components with < 10 bugs

@FunnelFiasco

CC BY-SA 4.0



## REOPENED REPORTS



@FunnelFiasco

CC BY-SA 4.0

I also wanted to see how many get reopened. This is an imperfect measure because it's based on the release where the BZ was closed, which isn't necessarily the release where it was reopened.

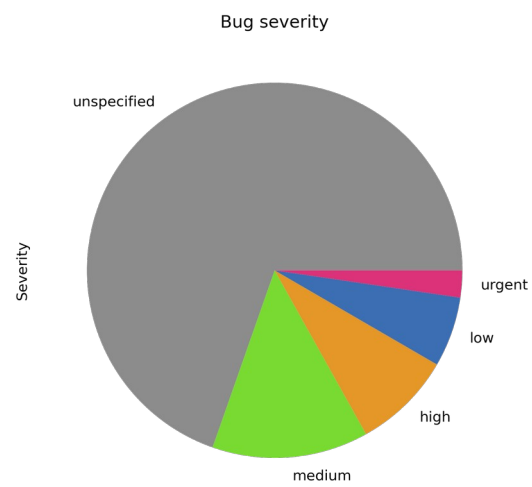
Does this downward trend mean we're getting better at really fixing bugs, or that people have given up when the fix doesn't work?

## SEVERITY

CC BY-SA 4.0

Bugzilla says priority is the developer's rating of when it will be worked on and severity is the user's rating of impact. I am not convinced these are used in the way the documentation thinks they should be used.

# BUG SEVERITY

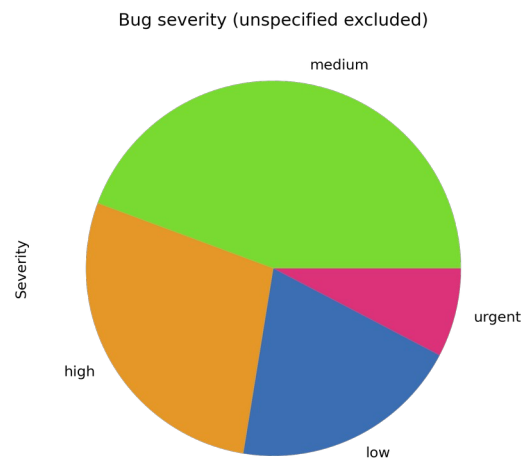


@FunnelFiasco

CC BY-SA 4.0



## BUG SEVERITY (WHEN SPECIFIED)



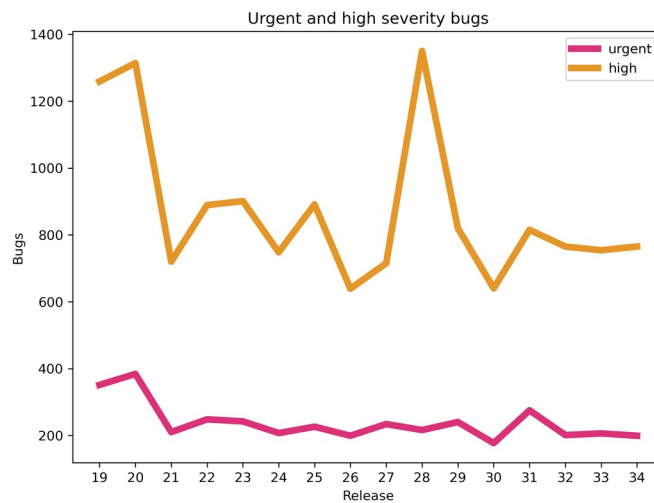
@FunnelFiasco

CC BY-SA 4.0

Removing the unspecified, we see a distribution that makes sense. Urgent bugs represent a small fraction. High is a larger segment, and medium is the highest. The reason I expect medium to be larger than low is that low-severity bugs probably go under-reported.



## SEVERITY OVER TIME



@FunnelFiasco

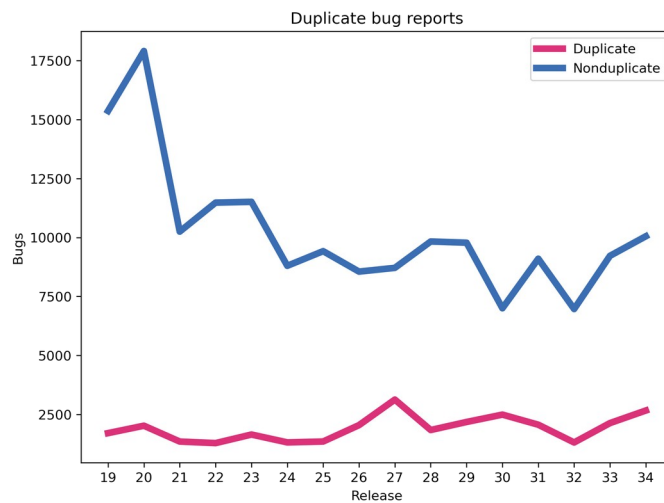
CC BY-SA 4.0

In general, our urgent and high severity bugs have stayed pretty steady in absolute numbers. I think this is pretty intuitive: our user base is big enough that we get reports for most high and urgent severity bugs, even if higher usage leads to higher low and medium severity reports.

# DUPLICATE BUGS

CC BY-SA 4.0

# DUPLICATE BUG REPORTS

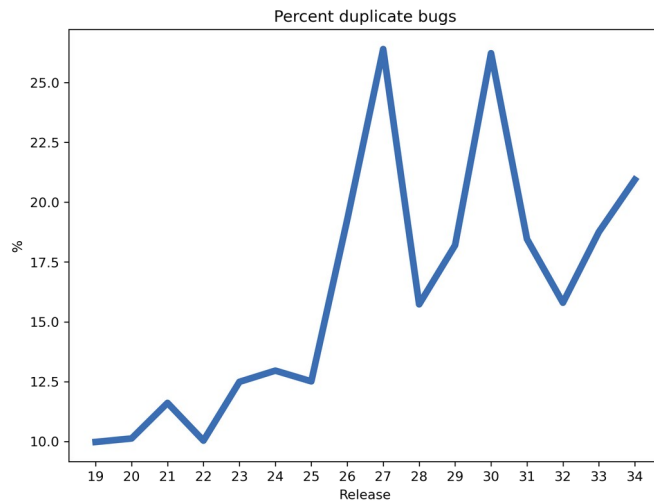


@FunnelFiasco

CC BY-SA 4.0

One thing I wanted to look at was the number of duplicate bug reports. This slide shows the duplicate and non-duplicate bug reports over time.

## % DUPLICATE BUG REPORTS



@FunnelFiasco

CC BY-SA 4.0

...this is reflected clearly when looking at the percentage of duplicates by release.

## FEWEST DUPLICATES PER COMPONENT

Component	Duplicates
ansible	0.67%
xen	0.84%
389-ds-base	0.85%
btrfs-progs	1.11%
synergy	1.26%
kubernetes	1.27%
mediawiki	1.32%
gtk2	1.35%
procps-ng	1.45%
autofs	1.49%

@FunnelFiasco

CC BY-SA 4.0



Not much movement from this time last year. Most of the components kept their relative positions. Some had an increase in percentage of duplicates while others had a decrease.

## OOPS! ALL DUPLICATES!

- 66 components have a 100% duplicate count

@FunnelFiasco

CC BY-SA 4.0



Three more components than last time had 100% duplicate count.

## MOST DUPLICATES PER COMPONENT

Component	Duplicates
tpm2-arbrmd	79.75%
open-vm-tools	76.40%
pypoppler	75.00%
bpytop	75.00%
libgnomekbd	74.14%
glib-networking	71.67%
imsettings	70.43%
kf5-globalaccel	70.21
plasma-drkonqi	69.86%
dleyna-renderer	67.78%

@FunnelFiasco

CC BY-SA 4.0



Most components saw a drop.

# BUG RESOLUTION

CC BY-SA 4.0



## CATEGORIZING CLOSURES

- Happy resolutions: CURRENTRELEASE, ERRATA, NEXTRELEASE, RAWHIDE, UPSTREAM
- Sad user resolutions: CANTFIX, DEFERRED, EOL, WONTFIX
- Sad maintainer resolutions: INSUFFICIENT\_DATA, NOTABUG, WORKSFORME
- DUPLICATE is excluded here

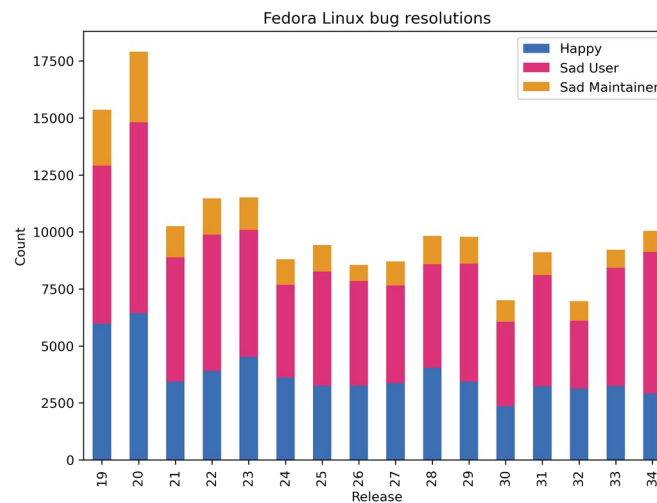
@FunnelFiasco

CC BY-SA 4.0



We can argue about which closure types belong in which category, but this seems reasonable based on how I've seen them used.

# CLOSURES BY CATEGORY

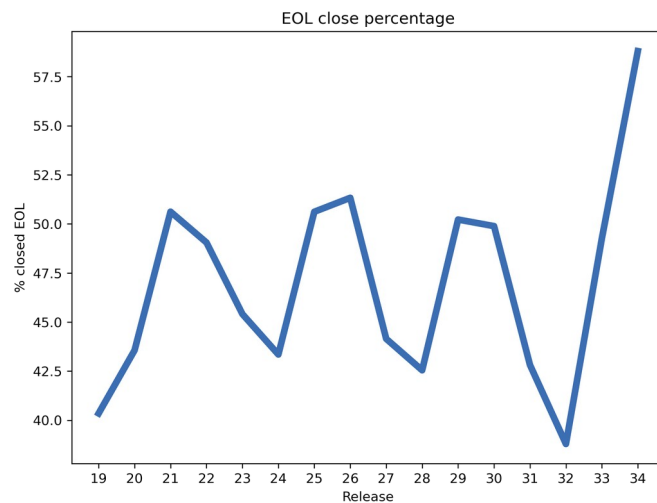


@FunnelFiasco

CC BY-SA 4.0

The happy closures tend to be smaller than I'd like. The sad user closures dominate the sad maintainer closures.

## EOL CLOSURE PERCENTAGE

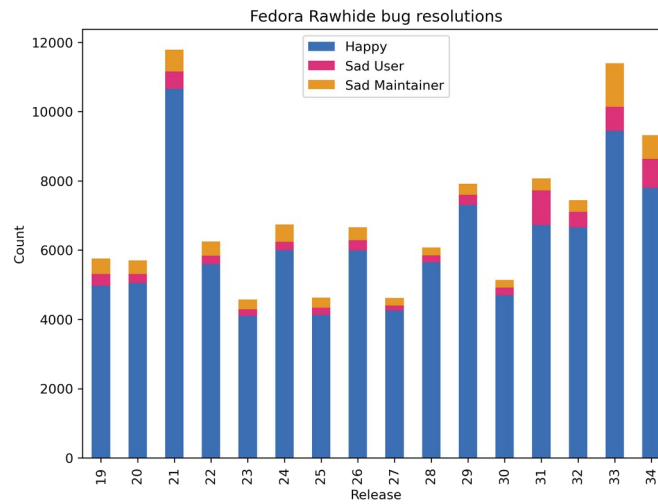


@FunnelFiasco

CC BY-SA 4.0

Every six months, we see complaints about how “Fedora never fixes my bugs, I’m done!” So I wanted to look at the closure percentage. Turns out it’s generally 40-50% of bugs that get closed EOL. What’s interesting is the apparent periodicity to the chart. I don’t have a good explanation for that. But we’ve gotten really bad the last few releases.

# RAWHIDE CLOSURES BY CATEGORY



@FunnelFiasco

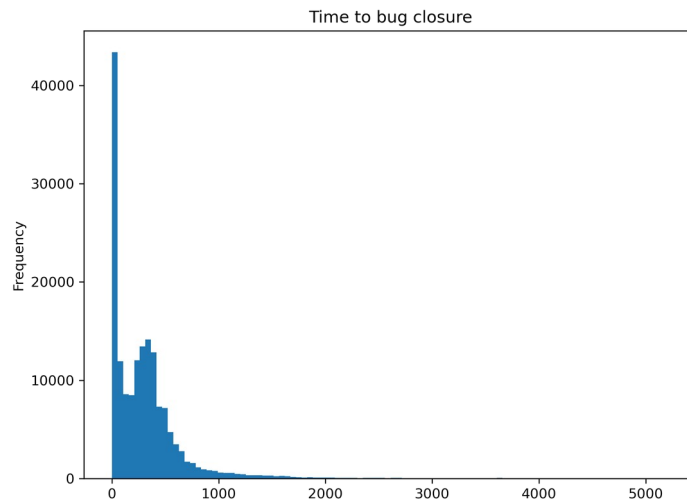
CC BY-SA 4.0

Rawhide, as you'd expect, is mostly happy closures.

# TIME TO RESOLUTION (TTR)

CC BY-SA 4.0

# TTR



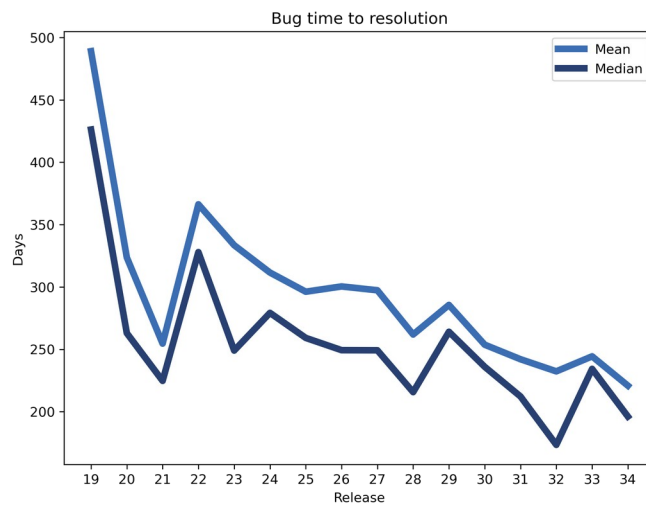
@FunnelFiasco

CC BY-SA 4.0



How quickly are bugs resolved one way or another?

## MEAN/MEDIAN TTR

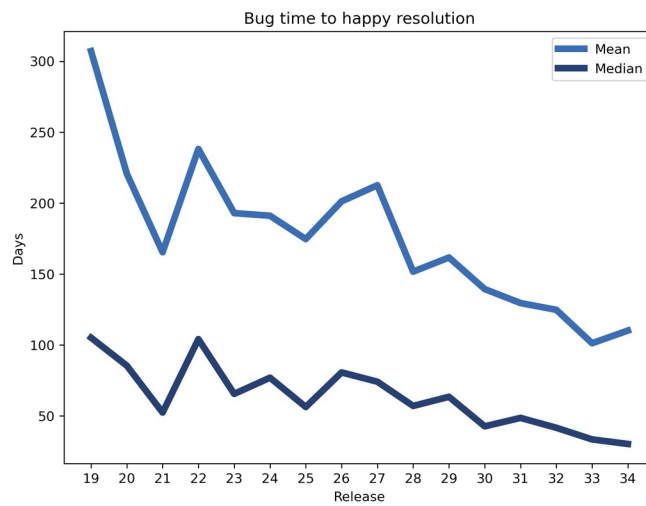


@FunnelFiasco

CC BY-SA 4.0

Looking at the mean and median, the trend is for faster resolution over time, however, we did regress a bit recently.

# MEAN/MEDIAN HAPPY TTR BY RELEASE



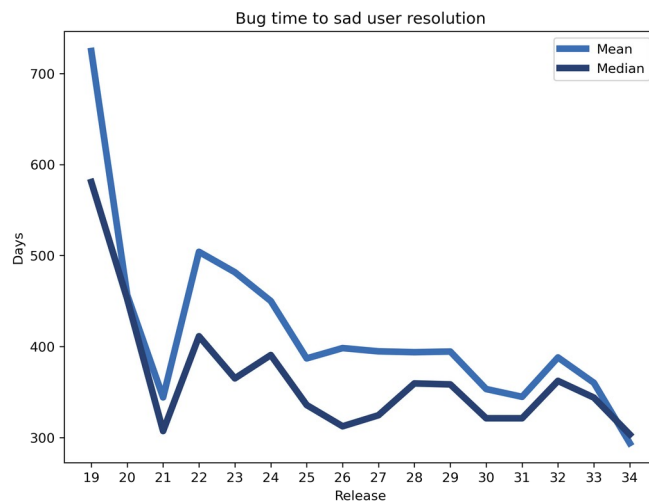
@FunnelFiasco

CC BY-SA 4.0

We're improving at fixing bugs!



## ME{AN,EDIAN} SAD USER TTR BY RELEASE



@FunnelFiasco

CC BY-SA 4.0

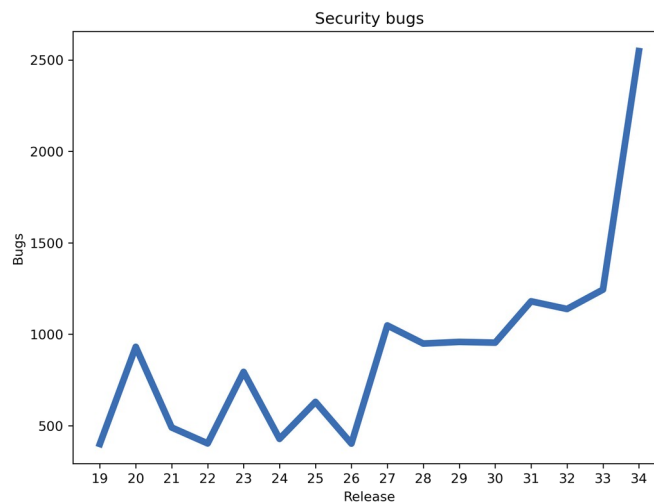
We're getting much faster at breaking the user's heart. This is good, actually, as it keeps things from languishing.

The large drop in sad user TTR from F19 to F20 is probably due to a large backlog of EOL closures.

# SECURITY

CC BY-SA 4.0

# SECURITY REPORTS

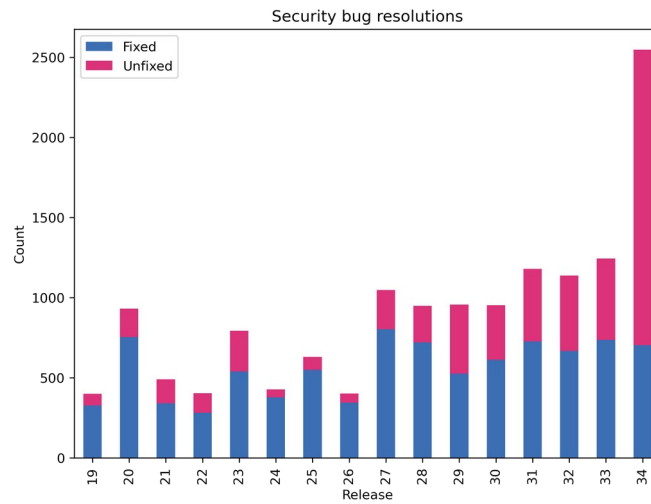


@FunnelFiasco

CC BY-SA 4.0

We see a big jump in security reports. But don't worry, it's actually because Red Hat's Product Security Team has gotten more involved in filing bugs.

# SECURITY RESOLUTIONS

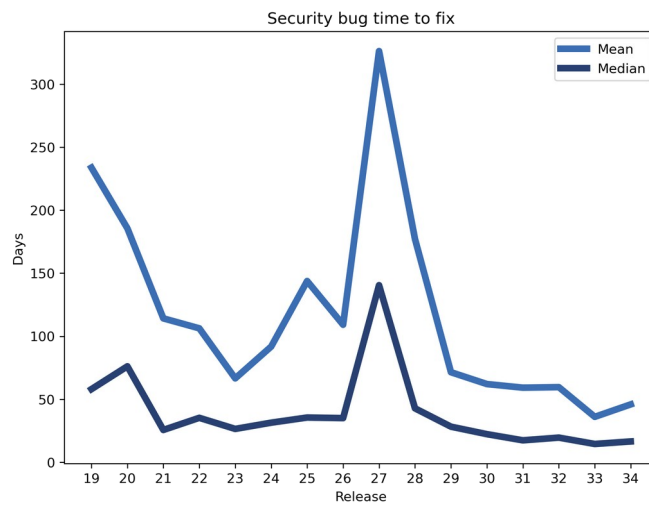


@FunnelFiasco

CC BY-SA 4.0

Fixed is the “happy” resolution types. Unfixed is both “sad user” and “sad maintainer”.

# SECURITY TIME TO FIX



@FunnelFiasco

CC BY-SA 4.0

This is specifically for the “happy” resolutions.

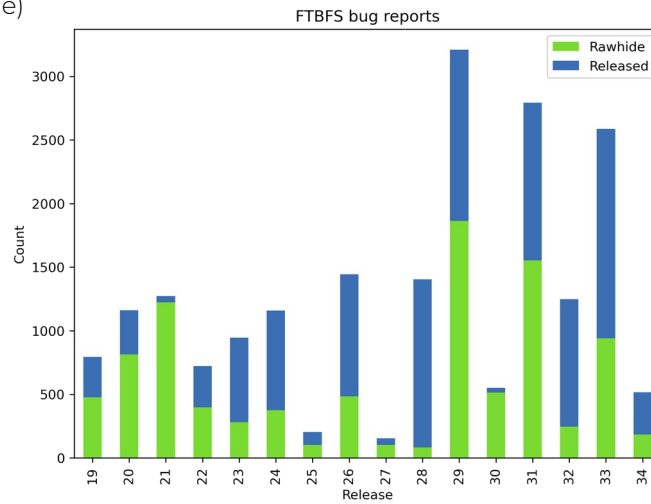
# PROCESS BUGS

CC BY-SA 4.0

Let's look at bug reports that are a part of various development processes.

# FAILS TO BUILD FROM SOURCE (FTBFS)

(includes Rawhide)



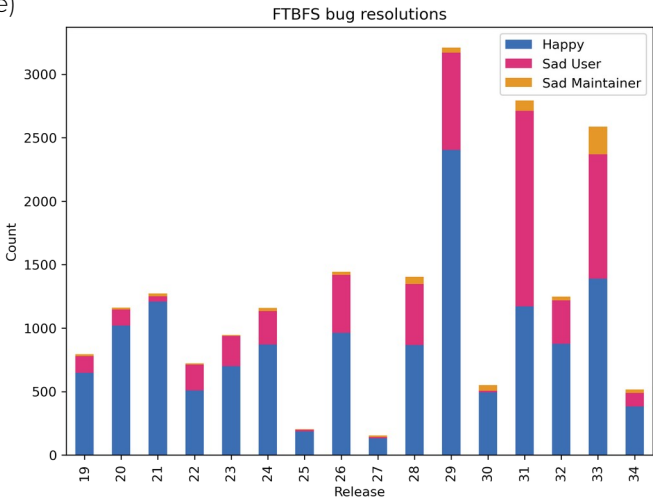
@FunnelFiasco

CC BY-SA 4.0

These bugs are when packages don't build, often as a result of updates to compiler or library packages.

# FTBFS RESOLUTIONS

(includes Rawhide)



@FunnelFiasco

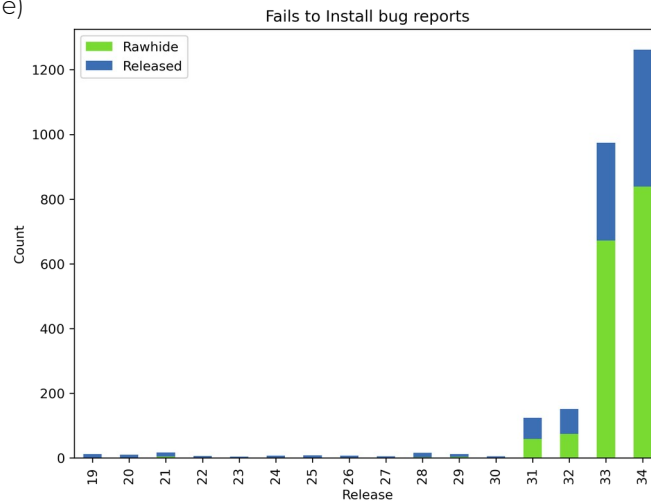
CC BY-SA 4.0





# FAILS TO INSTALL (FTI)

(includes Rawhide)



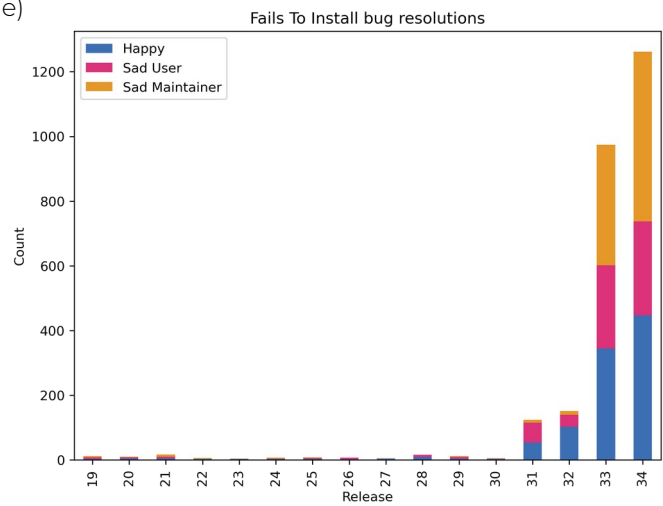
@FunnelFiasco

CC BY-SA 4.0

Fails to install bugs are when a package builds but won't install for some reason. We apparently only recently starting tracking this meaningfully. The first tracking bug appeared in F28.

# FTI RESOLUTIONS

(includes Rawhide)

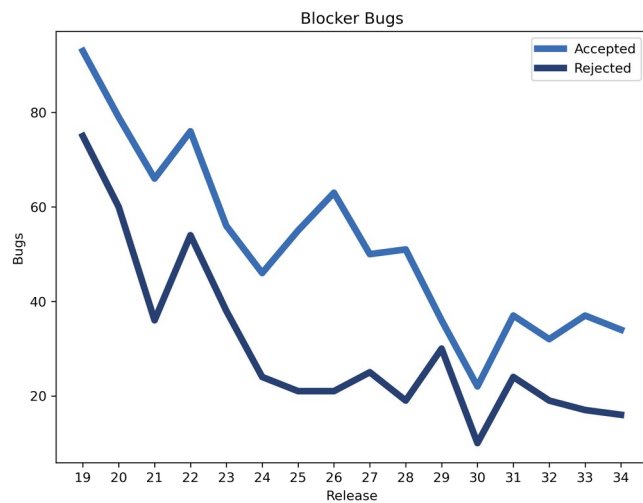


@FunnelFiasco

CC BY-SA 4.0



# BLOCKER BUGS

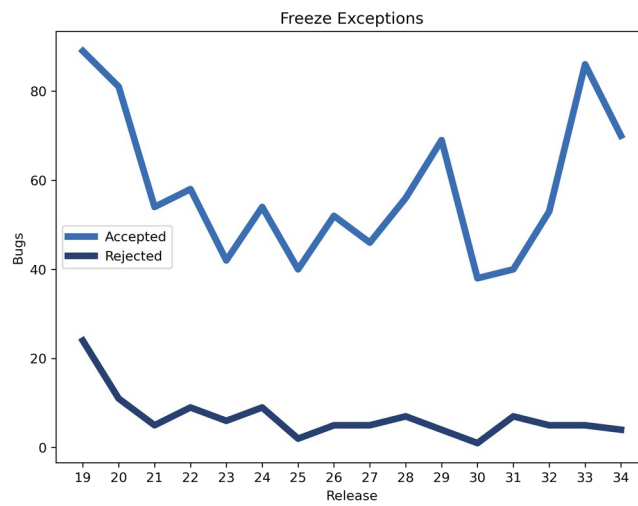


@FunnelFiasco

CC BY-SA 4.0

We accept more blockers than we reject, and we're generally getting fewer of each.

# FREEZE EXCEPTIONS

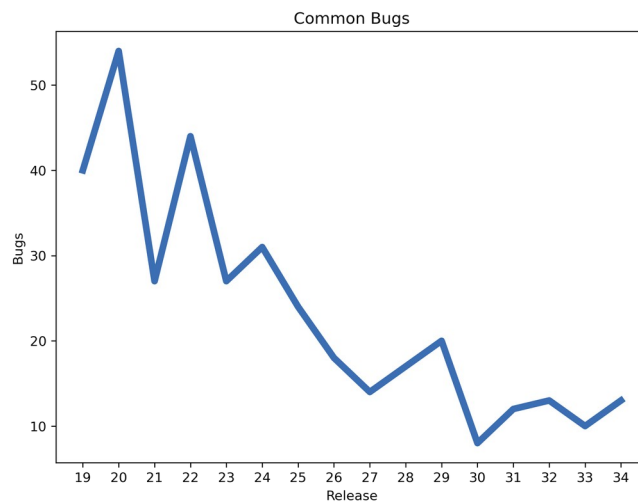


@FunnelFiasco

CC BY-SA 4.0

We've started getting more freeze exception requests, and grant most of them.

# COMMON BUGS



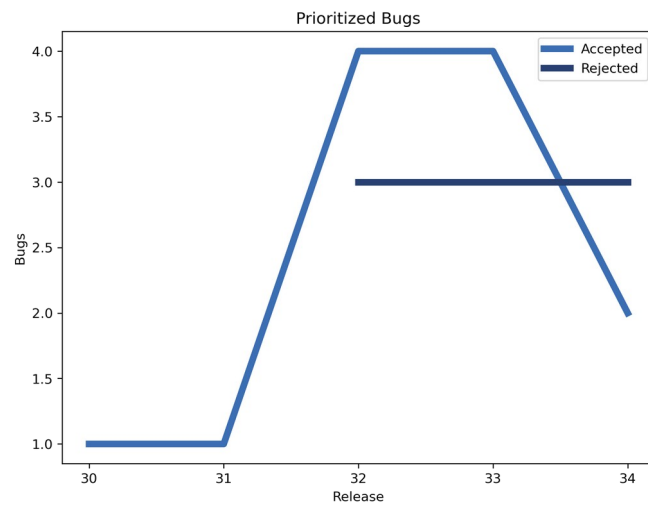
@FunnelFiasco

CC BY-SA 4.0



We have a lot fewer common bugs. Are we better at fixing the ones that people might know about or are we worse at highlighting them?

# PRIORITIZED BUGS



@FunnelFiasco

CC BY-SA 4.0

Okay, this isn't very useful, but it's a reminder to use the process.

# WHAT'S NEXT?

CC BY-SA 4.0

# COMING SOON

- Community Blog post(s)
- Your theories
- Probably more graphs!
- Unified notebook

@FunnelFiasco

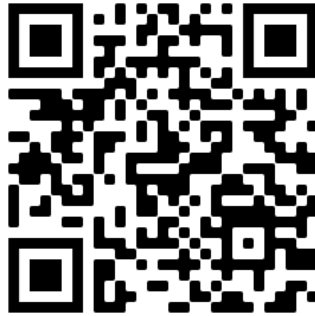
CC BY-SA 4.0





## EXPLORE IT YOURSELF

- <https://pagure.io/fedora-pgm/fedora-bug-data>



@FunnelFiasco

CC BY-SA 4.0

