Version 1.0, October 2024

Characterization of the Backlog Bugzilla - FreeBSD

Luis Cañas-Díaz <lcanas@bitergia.com>

Miguel Ángel Fernández Sánchez <mafesan@bitergia.com>

Introduction

🤝 These slides are part of the collaboration between the FreeBSD community and Bitergia, as an effort to make a characterization of the backlog from FreeBSD's Bugzilla instance.

📈 Although the whole history has been analyzed, we agreed to focus on the **open bugs** from the last two years and the last year, starting with the bugs from the **Base System** product.

⚙️ We produced a series of Dashboards to help with this characterization. This analysis can be extended to other periods and products by selecting the appropriate options.

Bird's eye view

October 2022 to October 2024

Bugzilla **bugs** over the last 2 years.
Let’s have a look at some insights 👉

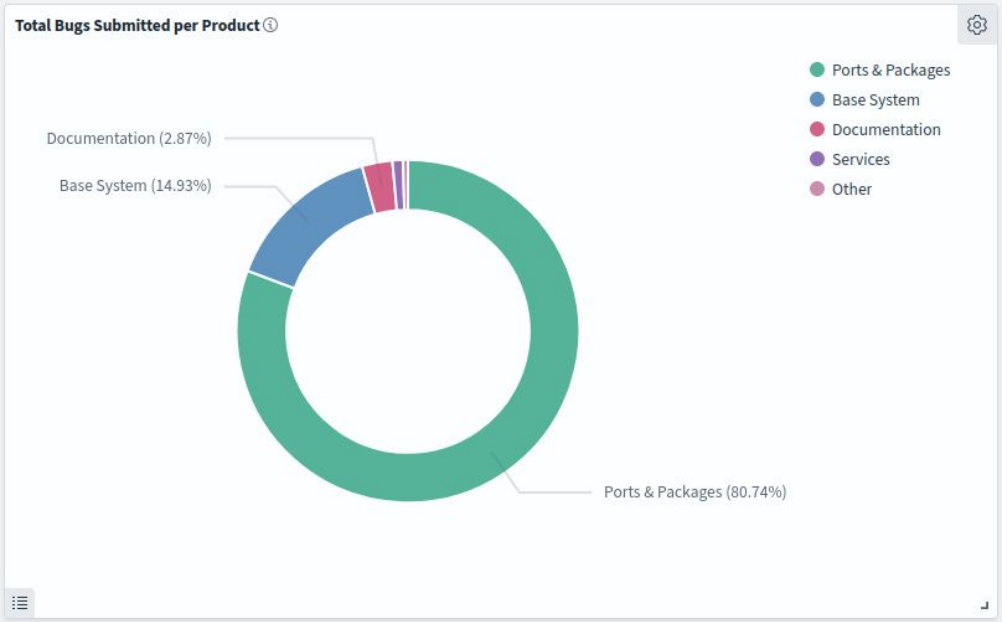
14,925
Bugs

2,326
Submitters

365
Assignees

5
Products

34
Components



Ports & Packages is the product with
more bugs submitted during the last 2 years (~80%),
followed by **Base System** (~15%)

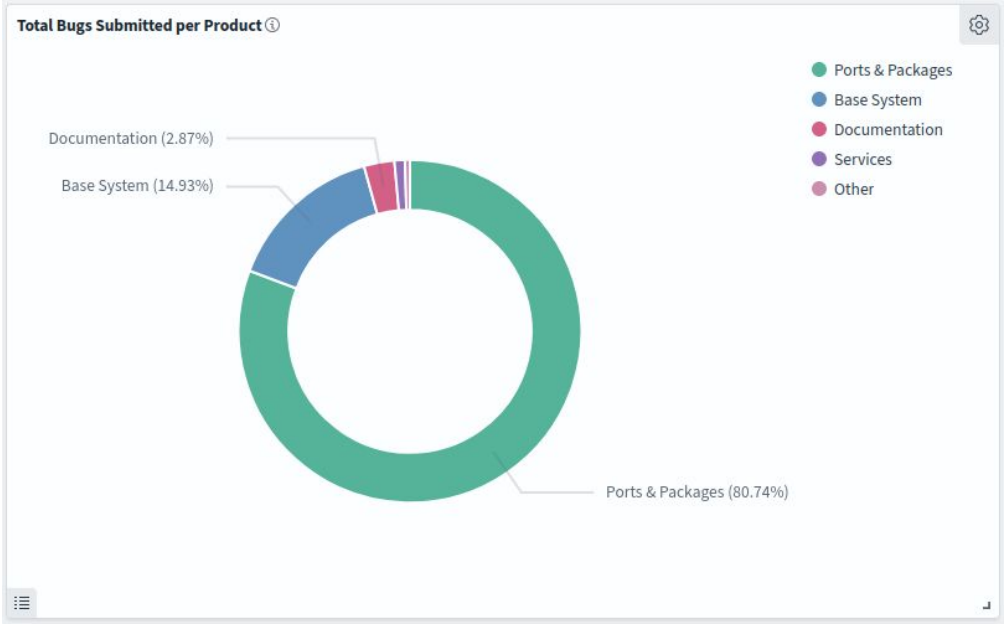
14,925
Bugs

2,326
Submitters

365
Assignees

5
Products

34
Components



Let’s focus on the bugs from “**Base System**”. We observe that there were between 68 and 131 bugs submitted per month.

1,058
Closed - Bugs

1,172
Any Open Status - Bugs

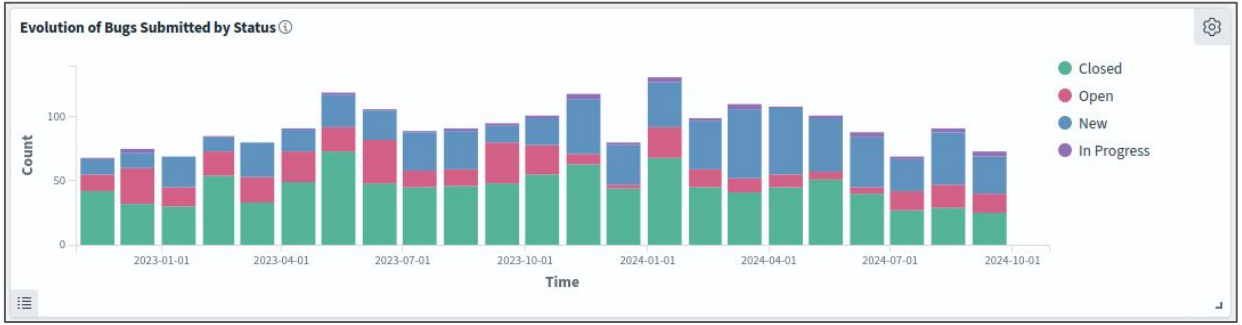
719
New - Bugs

396
Open - Bugs

57
In Progress - Bugs

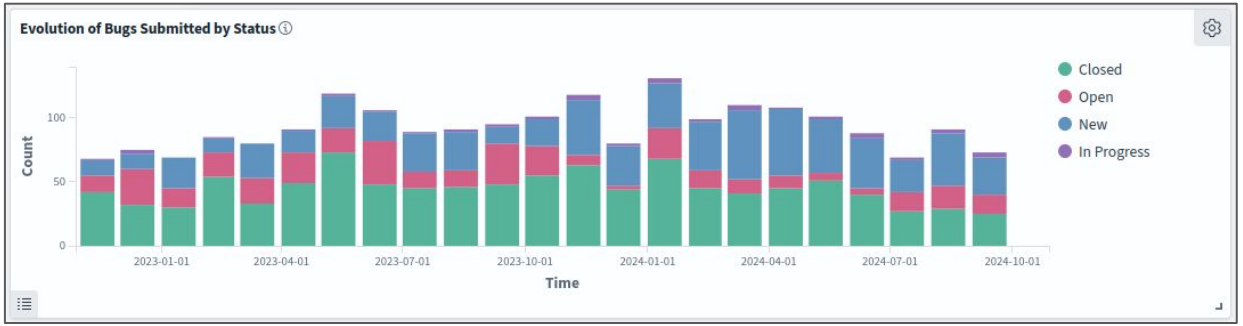
Looking closer at the bugs from **Base System** by status:

The level of activity is stable over the last 2 years

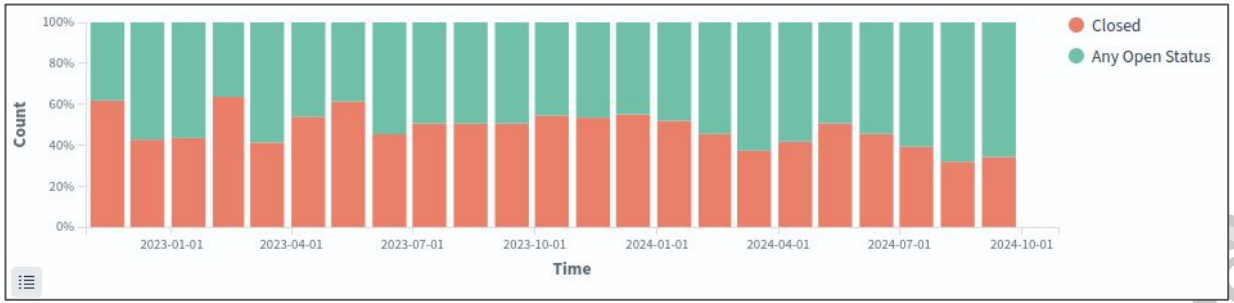


Looking closer at the bugs from **Base System** by status:

The level of activity is stable over the last 2 years



In percentage, around 50% of the bugs remain open, even those created more than a year ago.



Link to custom view

Characterization of the Backlog:

Backlog Evolution in Base System

October 2023 - October 2024

How does the backlog growth of **Base System** evolve for the last year?

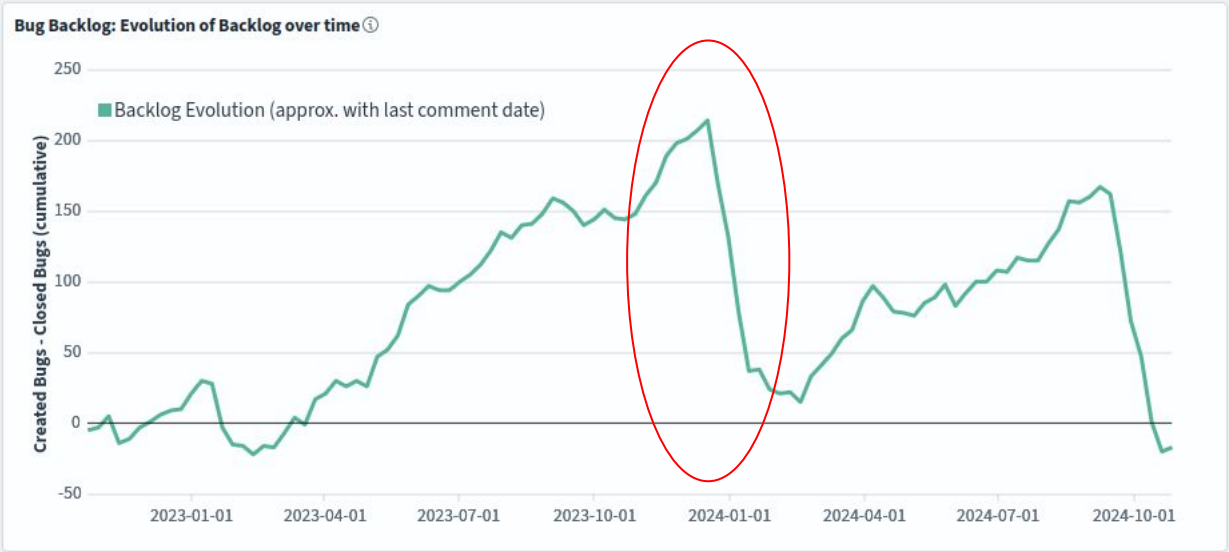
Bugzilla does not provide the closing date of the Bugs, so the values are approximated using the date of the last comment in "Closed" bugs.

Line above zero: Backlog is increasing

Line below zero: Backlog is decreasing

How does the backlog growth of **Base System** evolve for the last year?

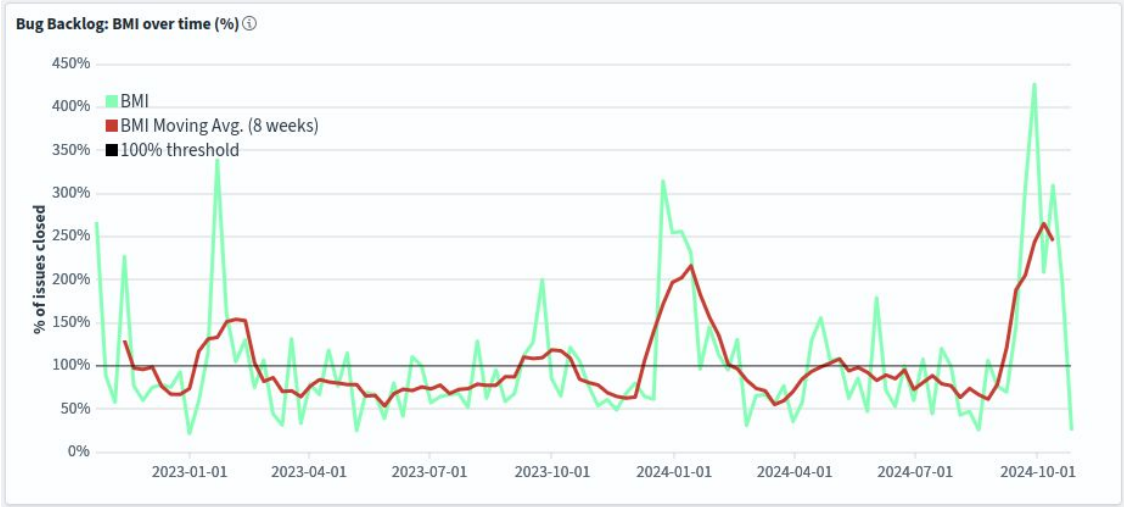
The Backlog started growing since March 2023 until there was a huge reduction of the backlog around January 2024.



Is the community able to digest the bugs of **Base System** and keep the pace?

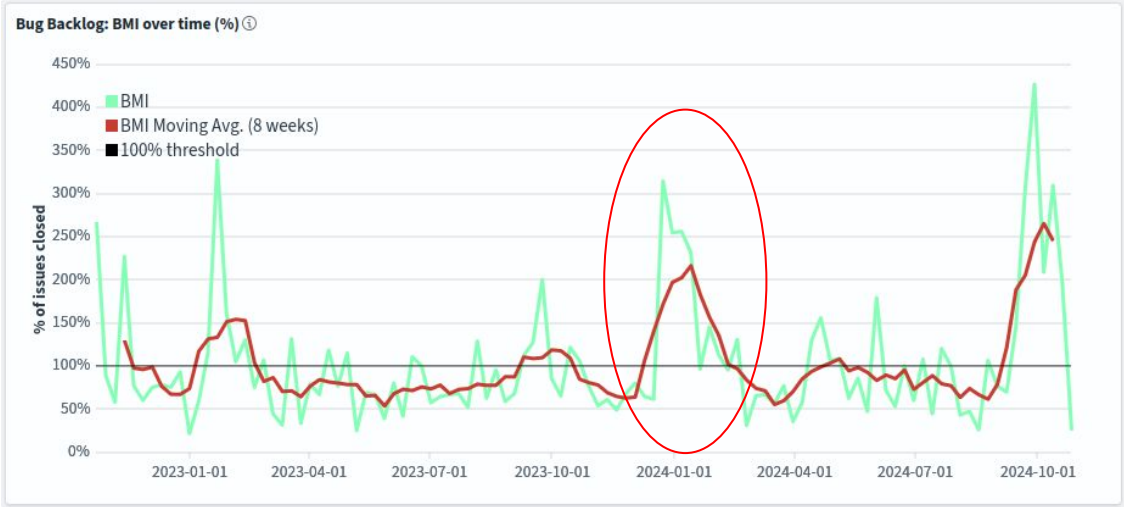
Another way to check the Backlog growth is the BMI (Backlog Management Index), which measures efficiency in closing issues.

If you are above the 100% threshold, the team is closing more tickets than the ones received, so the backlog is decreasing. If you are below 100%, the backlog is growing.



Is the community able to digest the bugs of **Base System** and keep the pace?

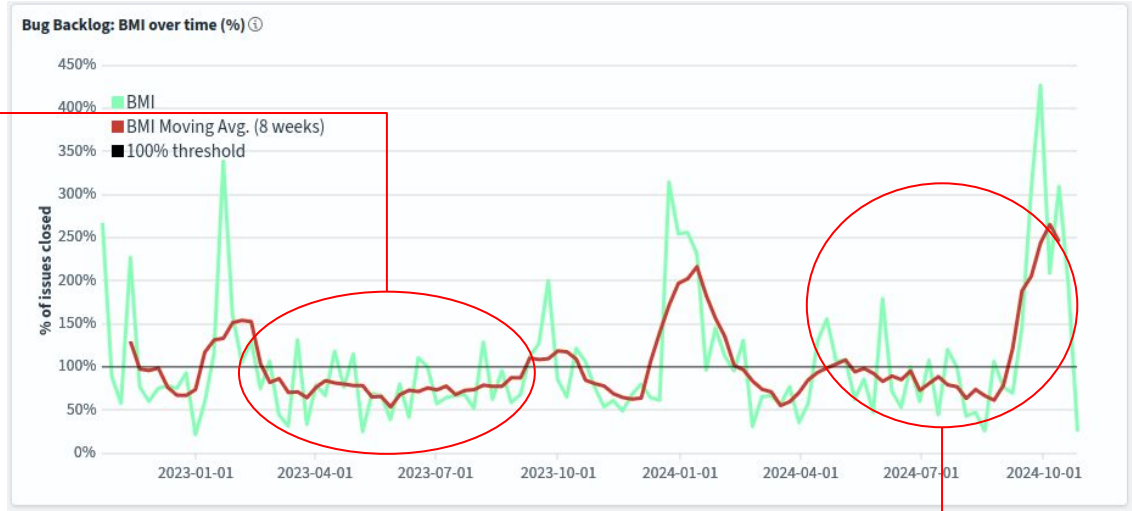
The peak from the last chart can be observed with this metric too. The community closed three times the number of bugs that were created during that period.



Is the community able to digest the bugs of **Base System** and keep the pace?

The moving average (the smoother line) indicates that the community had a hard time digesting the tickets from March to September 2023 (approximate dates).

More recently, the community seem to be able to keep the pace for Base System bugs.



Characterization of the Backlog:

Attention Times in Base System

October 2023 - October 2024

Let’s focus on the open bugs in the last year for **Base System** and classify them according to their attention times.

Let’s focus on the open bugs in the last year for **Base System** and classify them according to their attention times.

Bugs with any open status (Open, New, In Progress).

Let’s focus on the open bugs in the last year for **Base System** and classify them according to their attention times.

Bugs with any open status (Open, New, In Progress).

Open bugs, attended and recently commented.

Let’s focus on the open bugs in the last year for **Base System** and classify them according to their attention times.

Bugs with any open status (Open, New, In Progress).

Open bugs, attended and recently commented.

Open bugs, unattended (no comments from any user different from the author).

Let’s focus on the open bugs in the last year for **Base System** and classify them according to their attention times.

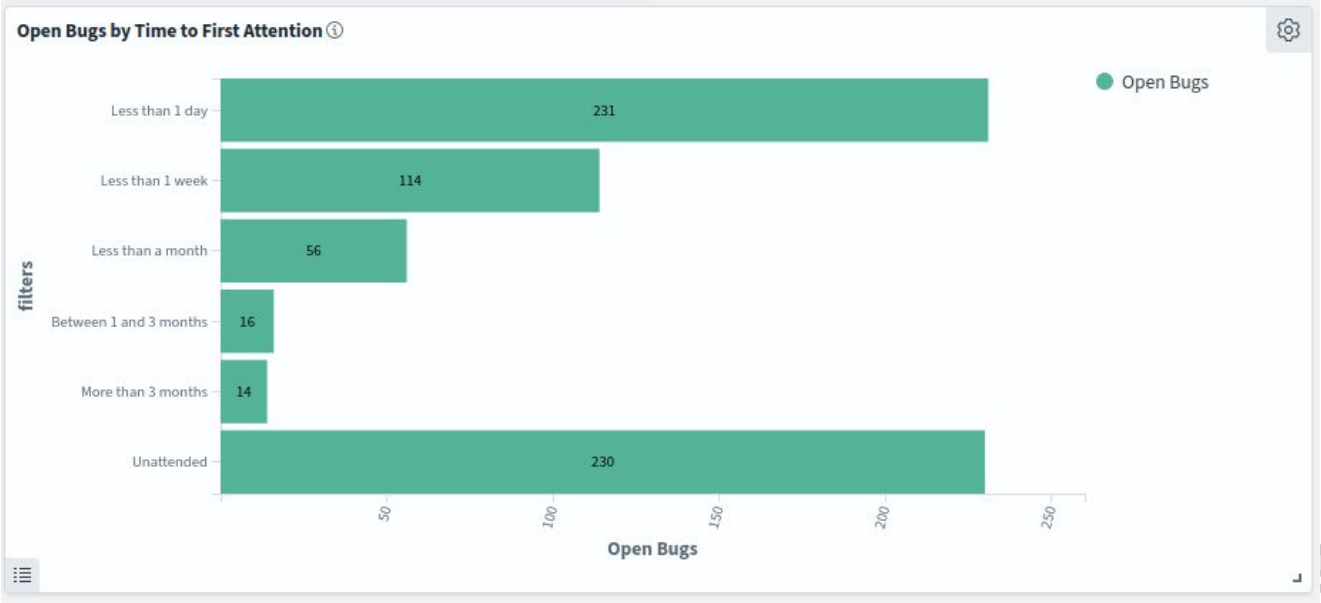
Bugs with any open status (Open, New, In Progress).

Open bugs, attended and recently commented.

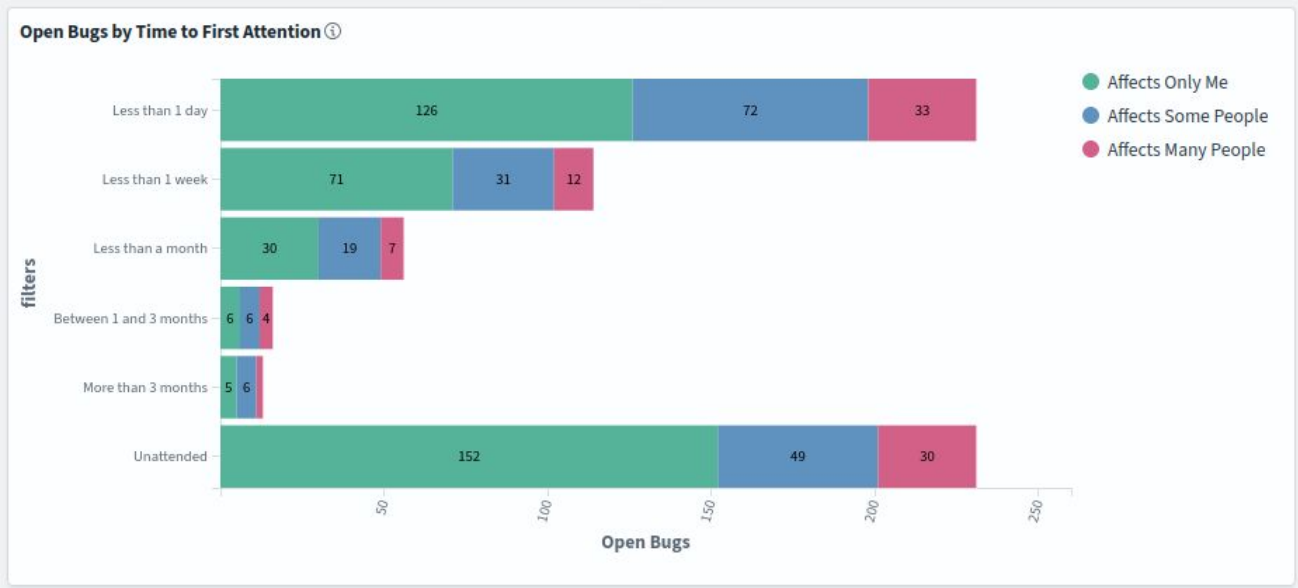
Open bugs, unattended (no comments from any user different from the author).

Open bugs, abandoned (last comment submitted more than 3 months ago).

Classifying the Bugs by the Time to First Attention

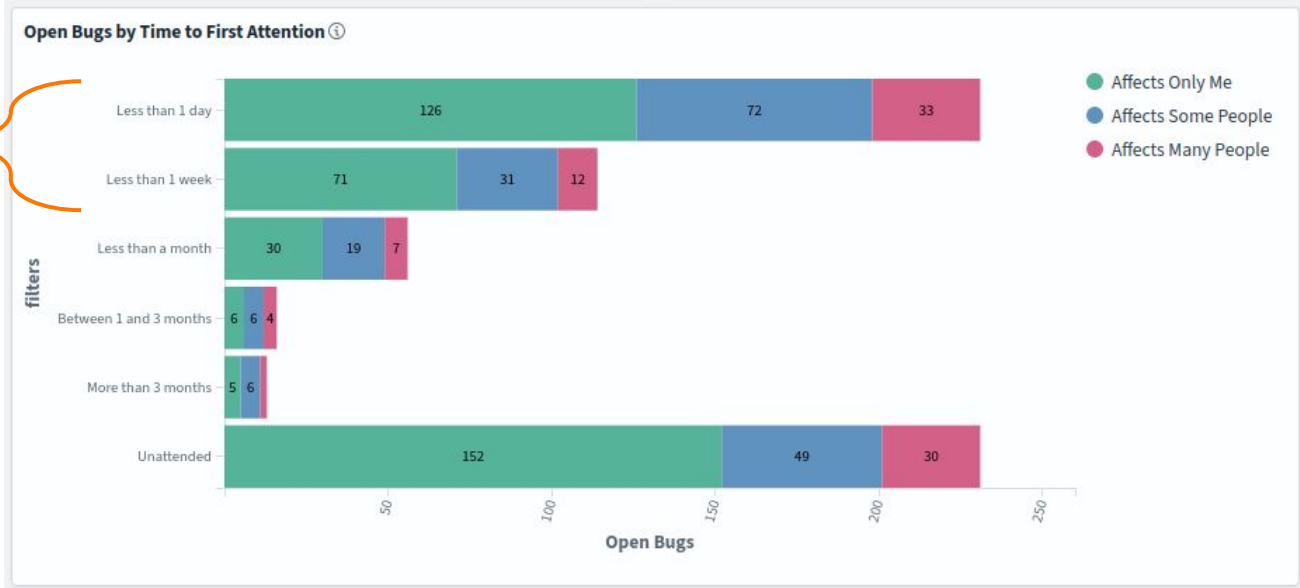


Classifying the Bugs by the Time to First Attention and Severity



Classifying the Bugs by the Time to First Attention and Severity

Almost 50% of the open bugs are attended in less than a week, and most of them within the same day.

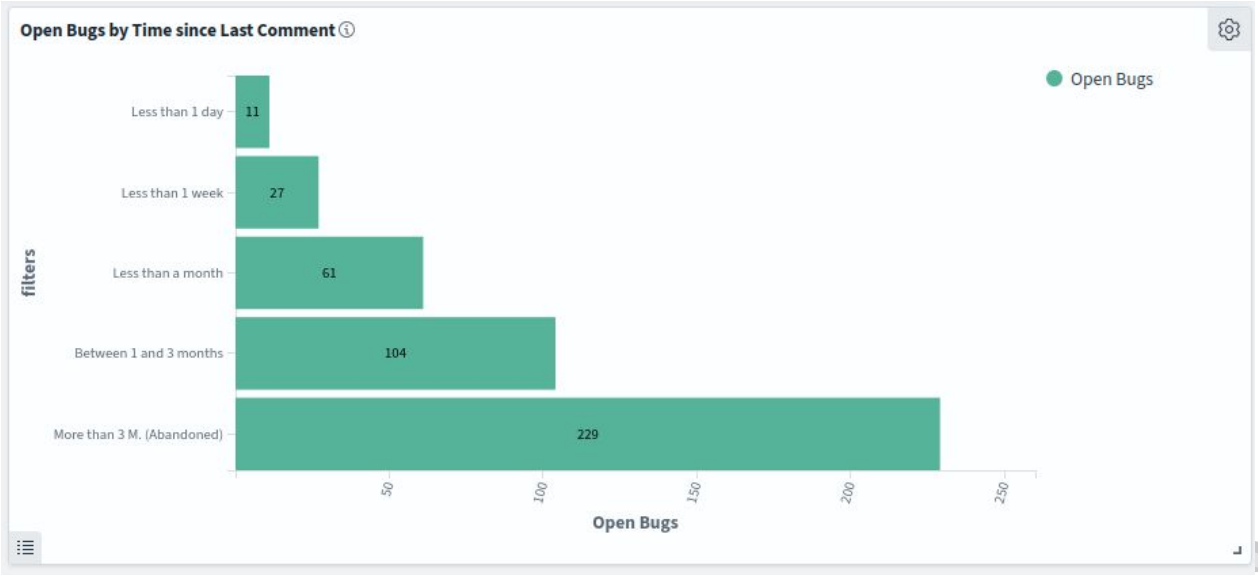


Classifying the Bugs by the Time to First Attention and Severity

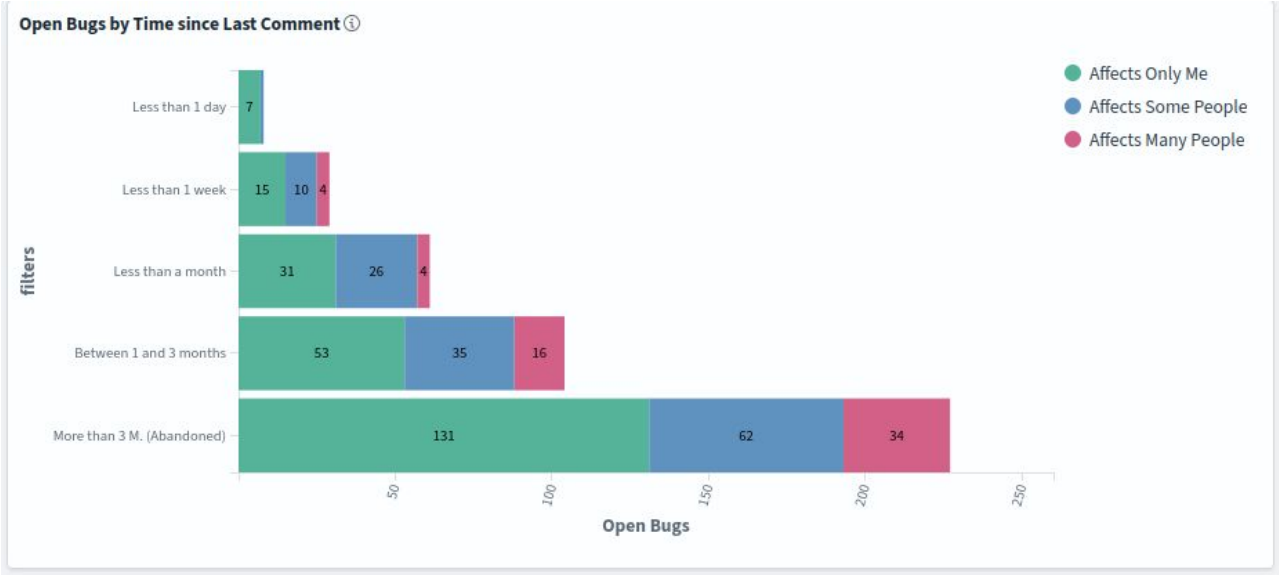
Almost 50% of the open bugs are attended in less than a week, and most of them within the same day.

There is a relevant number of unattended bugs (~80) affecting more than one person in the last year.

Classifying the Bugs by the Time since Last Comment



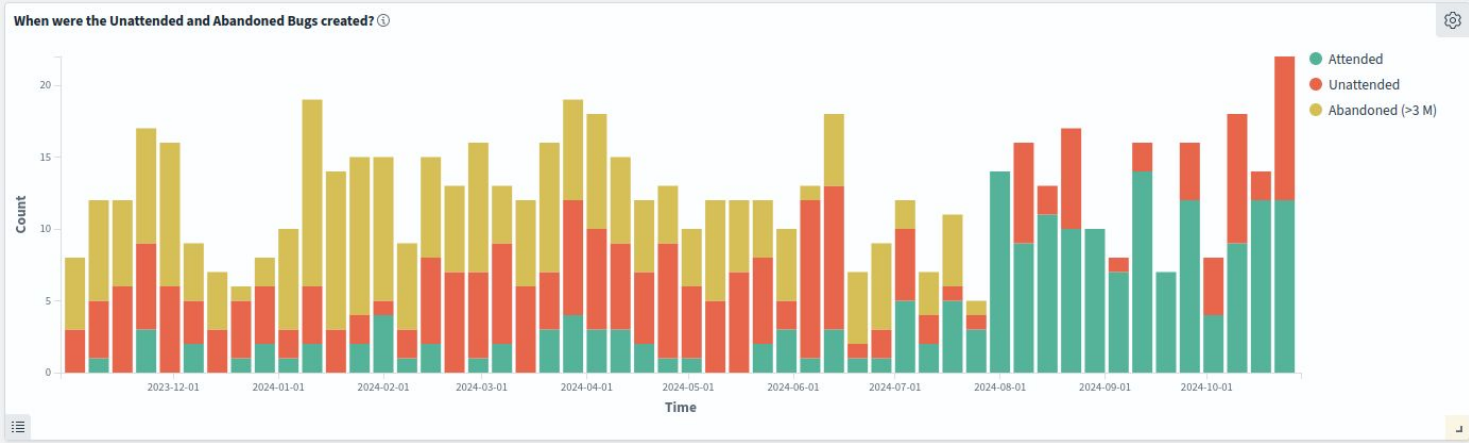
Classifying the Bugs by the Time since Last Comment + Severity



Classifying the Bugs by the Time since Last Comment + Severity

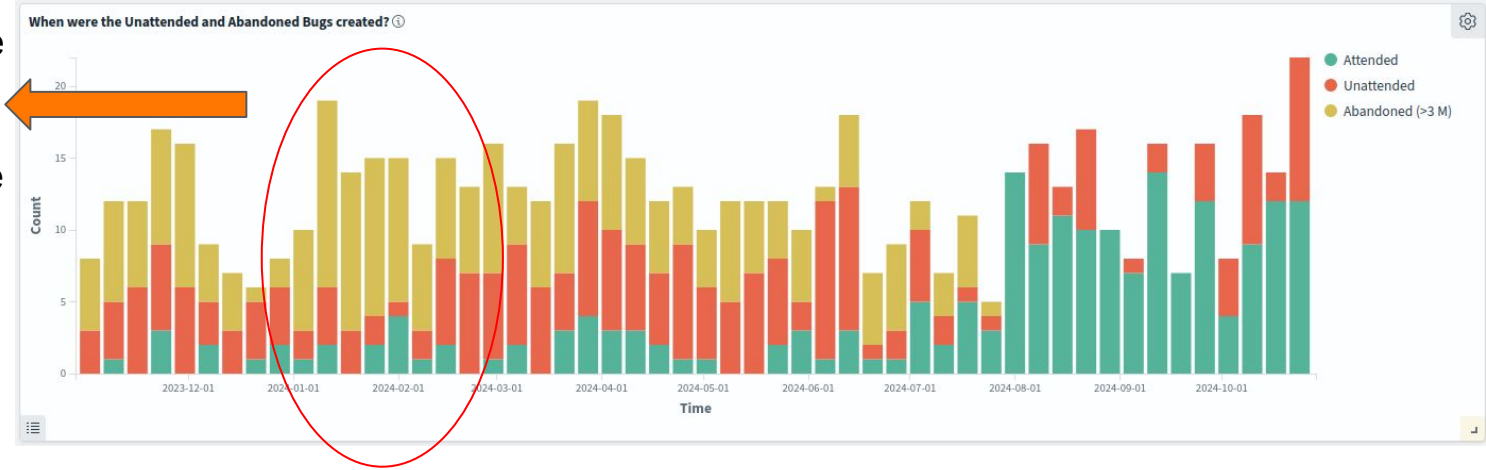
Besides the unattended bugs, there are almost 100 tickets affecting more than one person that are still open and with no activity for three months or more.

When were the Unattended and Abandoned Bugs created?



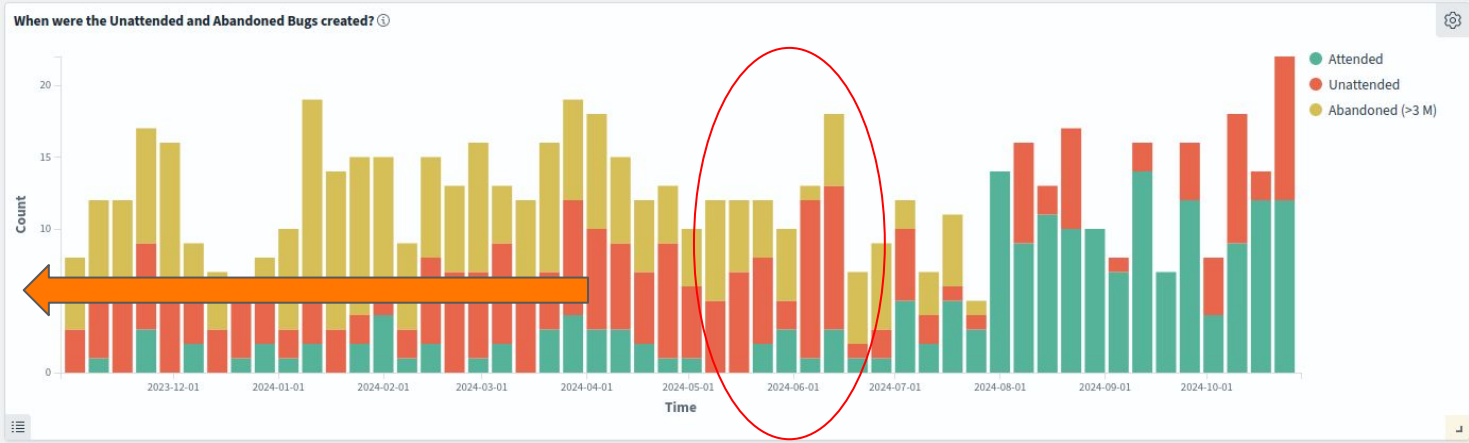
When were the Unattended and Abandoned Bugs created?

Tickets created in the first two months of 2024 are attended, but many of them are abandoned.



When were the Unattended and Abandoned Bugs created?

Most of the tickets created around June 2024 are still unattended.



Characterization of the Backlog:

Ideas for further bottleneck detection

Which other parameters could help the community identifying where the bottlenecks are happening?

- ✅ Backlog Growth
- ✅ Time to First Attention
- ✅ Time since Last Comment
- ✅ Severity
- ❌ Priority (Not used by the community)
- ❓ Component
- ❓ Keywords

Which other parameters could help the community identifying where the bottlenecks are happening?

- ✅ Backlog Growth
- ✅ Time to First Attention
- ✅ Time since Last Comment
- ✅ Severity
- ❌ Priority (Not used by the community)
- ❓ Component
- ❓ Keywords

Classification as “Attended”, “Unattended” and “Abandoned”

Which other parameters could help the community identifying where the bottlenecks are happening?

- ✅ Backlog Growth
- ✅ Time to First Attention
- ✅ Time since Last Comment
- ✅ Severity
- ❌ Priority (Not used by the community)
- ❓ Component
- ❓ Keywords

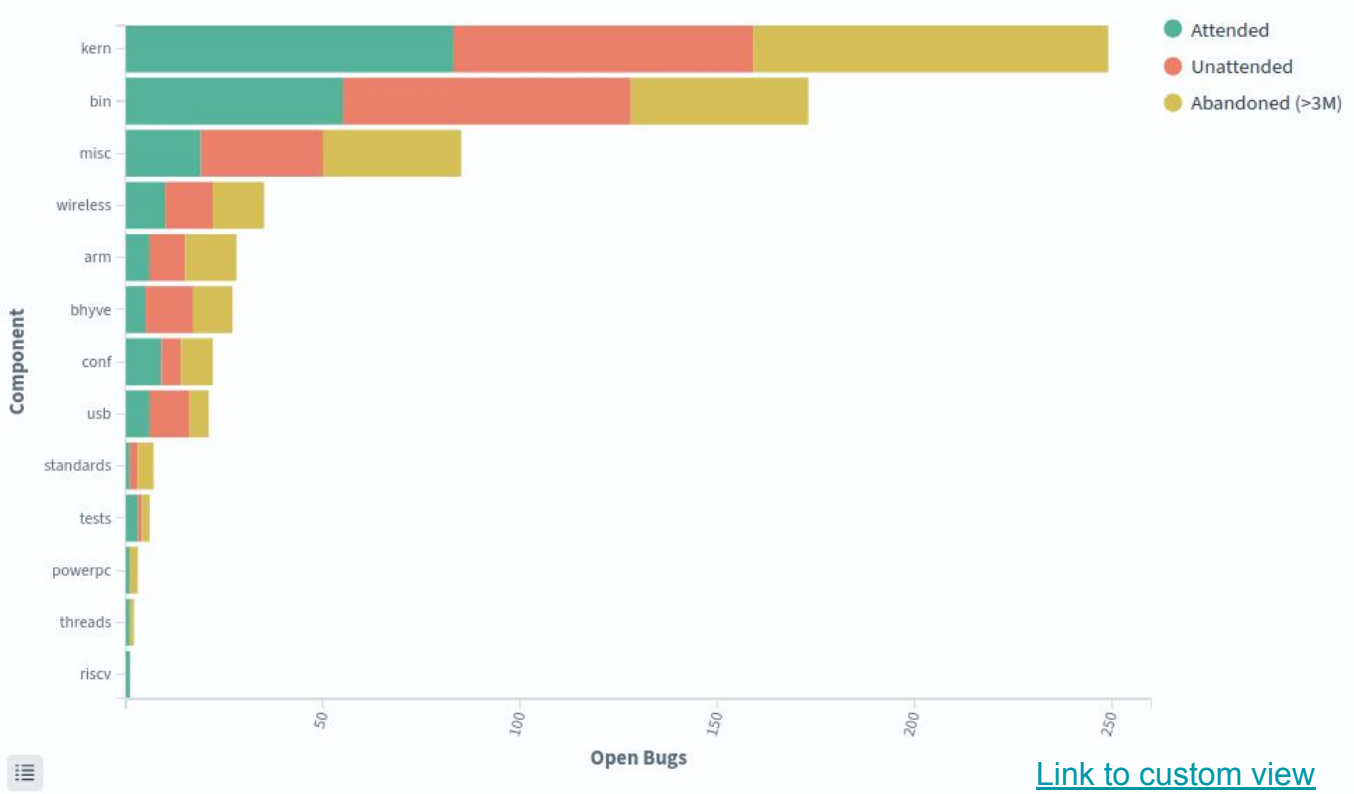
Classification as “Attended”, “Unattended” and “Abandoned”

Bugs with “Affects Only Me” priority can be filtered out to reduce noise and identify more relevant bugs needing attention.

Classifying Open Bugs by Component in the **Base System** product

The “Component” specified in the bug could help to identify where the bottlenecks may be happening.

However, the bugs from “kern” and “bin” components are accumulating many results. Additional sub-labels would be needed to improve the classification.

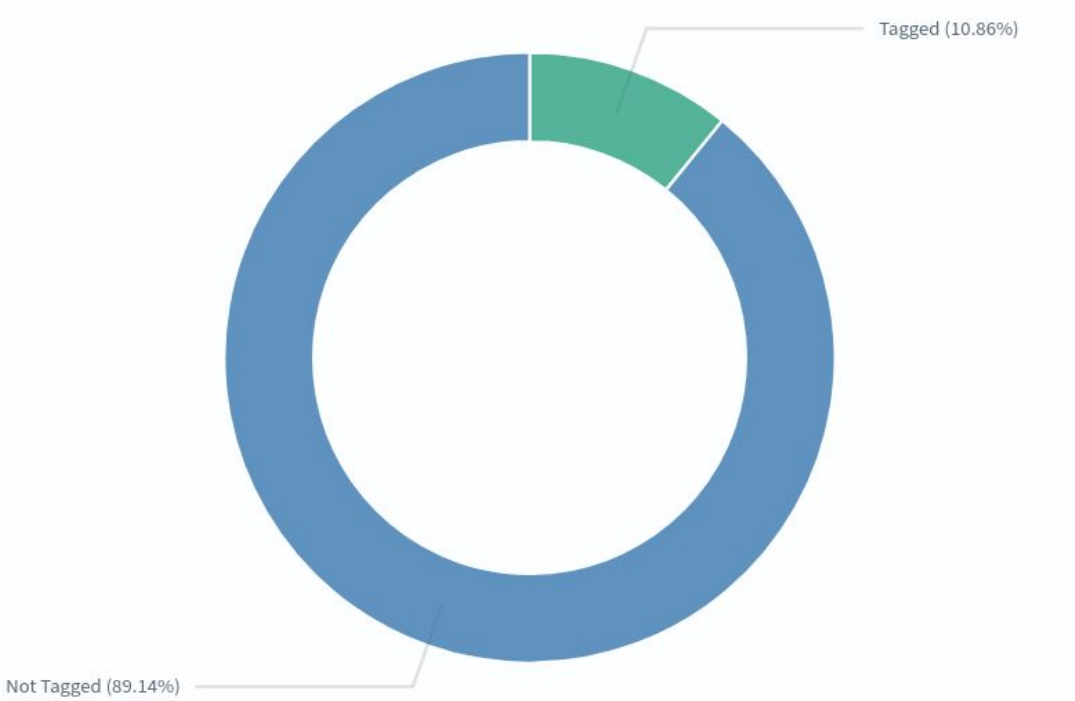


Link to custom view

Using Keywords to classify bugs

In the whole history of the project since November 1993, the Keywords have been barely used in the Bugzilla tickets.

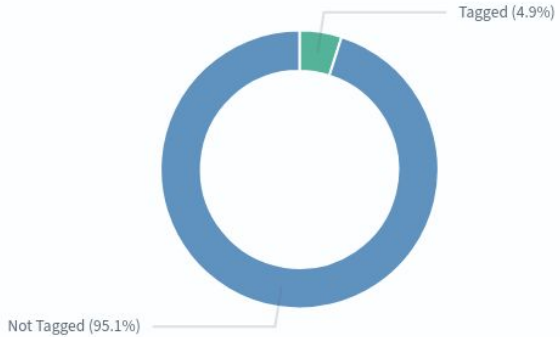
Only 10% of the bugs are tagged with at least one keyword. This is still happening for the bugs created in the last year.



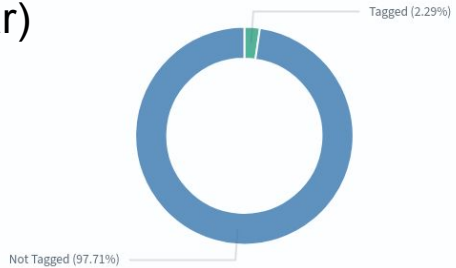
Link to custom view

Usage of Keywords to classify bugs in each Product (Last year)

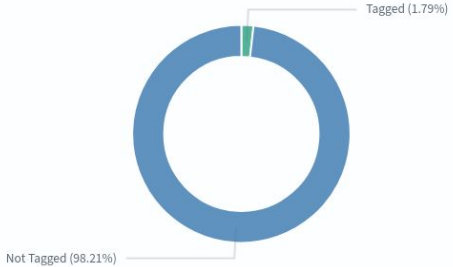
Products & Services



Documentation

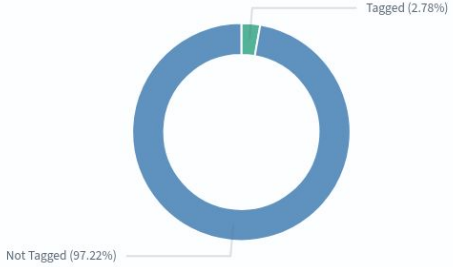


Services



Base System

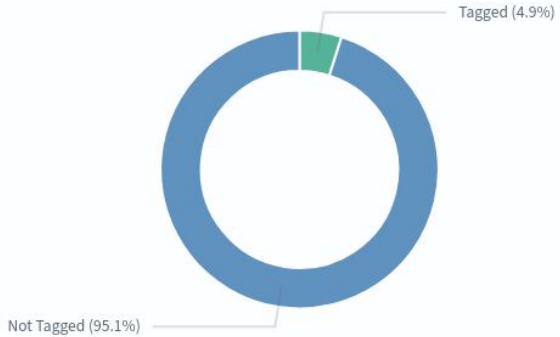
Other



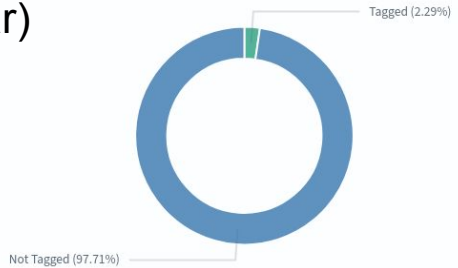
Link to custom view

Usage of Keywords to classify bugs in each Product (Last year)

Products & Services

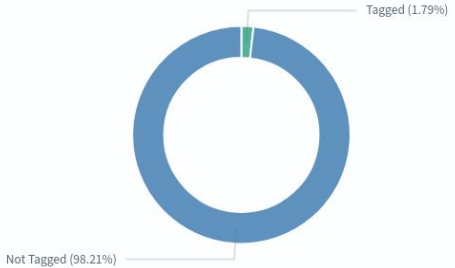


Documentation

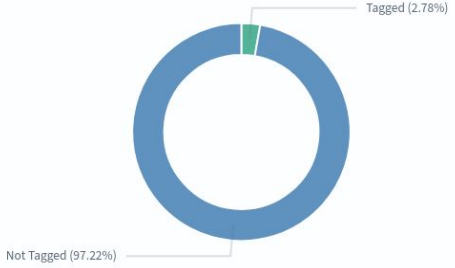


Base System

Services

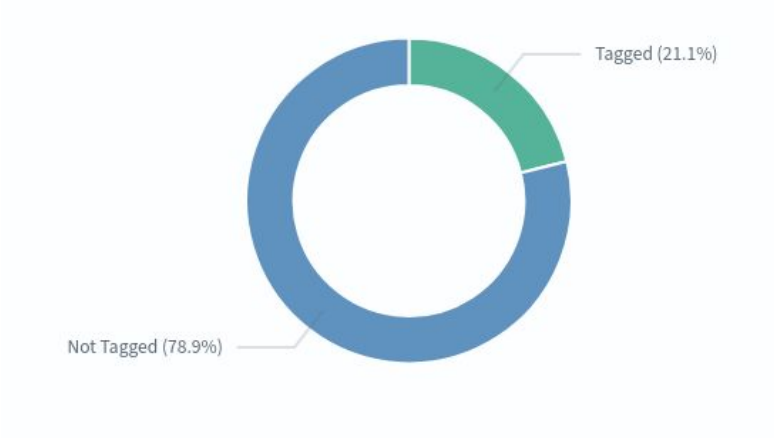


Other



Most relevant Keywords by number of bugs in Base System

Base System is the product with a greater proportion of tagged bugs: around 20% of the ones created during the last year contain one or more Keywords.



Most relevant Keywords by number of bugs in Base System

Base System is the product with a greater proportion of tagged bugs: around 20% of the ones created during the last year contain one or more Keywords.

These are the most common Keywords by number of bugs where they are used. Having such keywords could help understand where the bottlenecks may be happening.

Link to custom view

Ideas for next steps and future analysis

💡 Improve Backlog Characterization:
- Enforce best practises on tagging bugs with Keywords.
- Compute the real closing date by analyzing the history.

💡 Analyze the Bug descriptions to further classification (hot topics, sub-categories, etc.).

💡 Review Fairness: Are bugs being attended equally regardless of the submitting organization?

Version 1.0, October 2024

Characterization of the Backlog Bugzilla - FreeBSD

Luis Cañas-Díaz <lcanas@bitergia.com>

Miguel Ángel Fernández Sánchez <mafesan@bitergia.com>