2024 OpenAFS Workshop

OpenAFS Project Report

June 10, 2024

About me

Michael Meffie

Principal Software Engineer

Sine Nomine Associates

Commercial support for OpenAFS

mmeffie@sinenomine.net

Table of Contents

1. Year in Review

Updates and some statistics of work in the OpenAFS project.

Table of Contents

1. Year in Review

Updates and some statistics of work in the OpenAFS project.

2. Maintainers

Announcement of new maintainers added to the OpenAFS project.

Table of Contents

1. Year in Review

Updates and some statistics of work in the OpenAFS project.

2. Maintainers

Announcement of new maintainers added to the OpenAFS project.

3. Project Infrastructure

Review of our current project infrastructure and who is keeping it going.

Table of Contents

1. Year in Review

Updates and some statistics of work in the OpenAFS project.

2. Maintainers

Announcement of new maintainers added to the OpenAFS project.

3. Project Infrastructure

Review of our current project infrastructure and who is keeping it going.

4. Going forward

Setting general project goals.

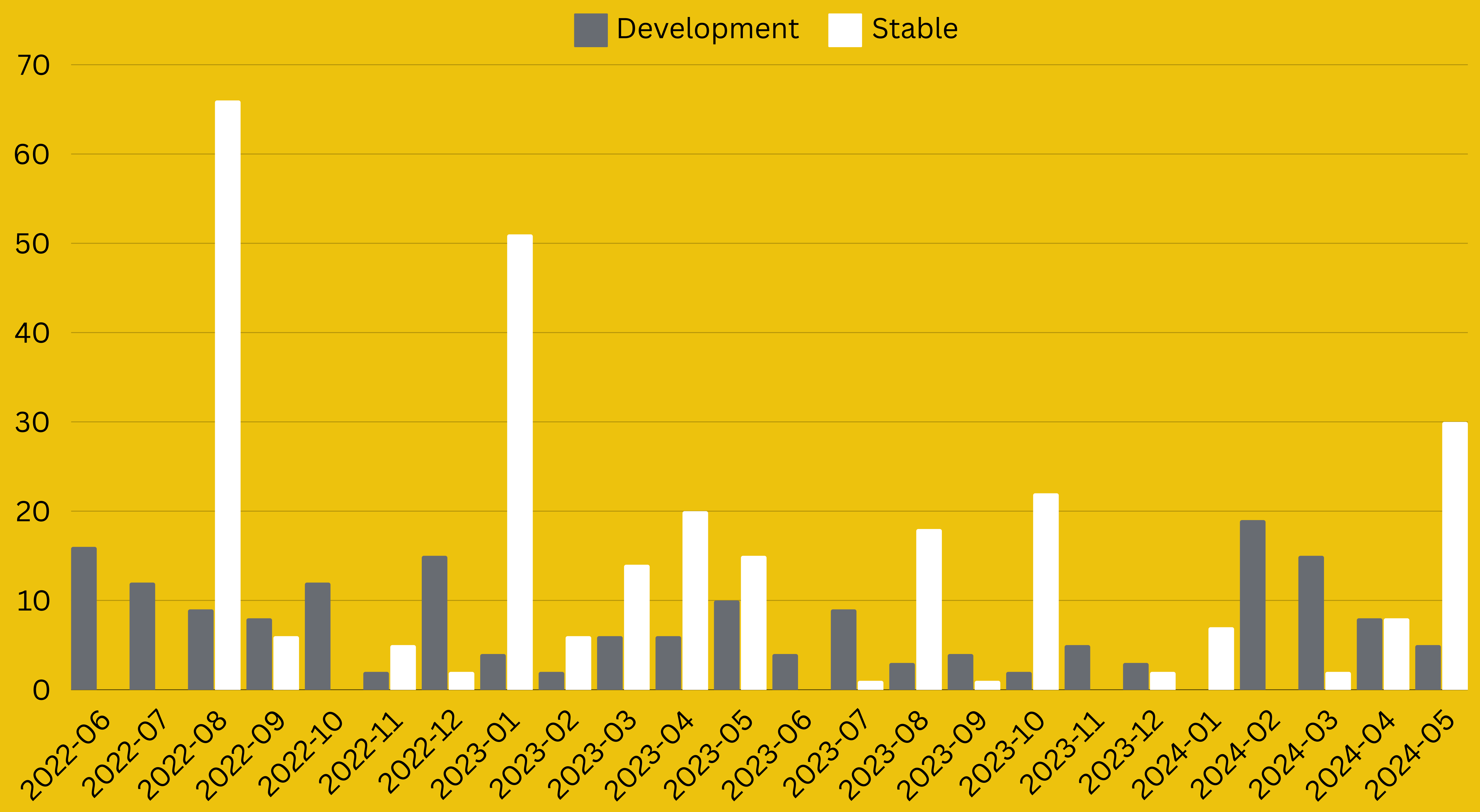
01

Year in Review

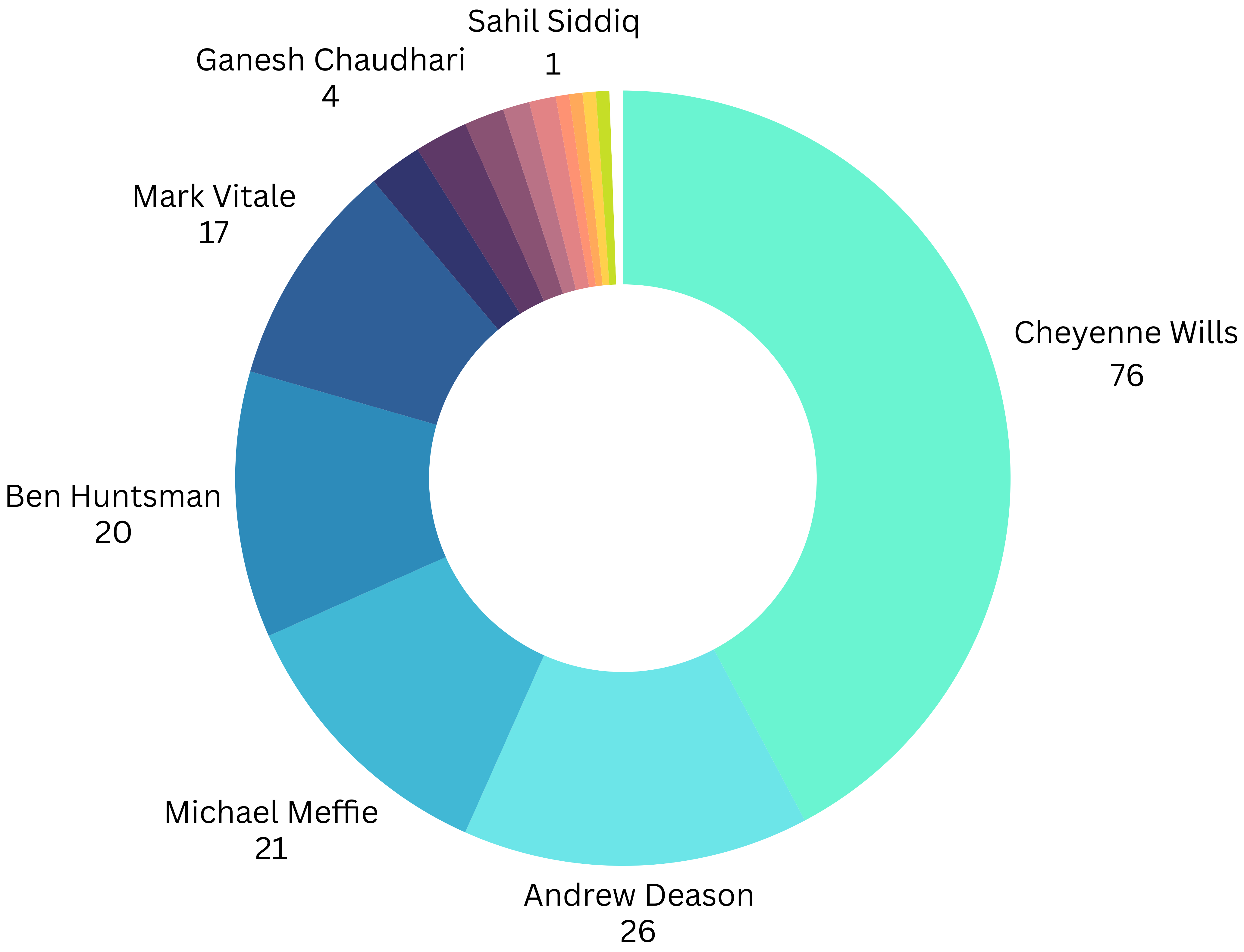
01 Year in Review

A look at our recent project activity.

Code merges per month



Developer activity in the last 24 months, showing number of commits merged on the master branch.



Tracking the backlog

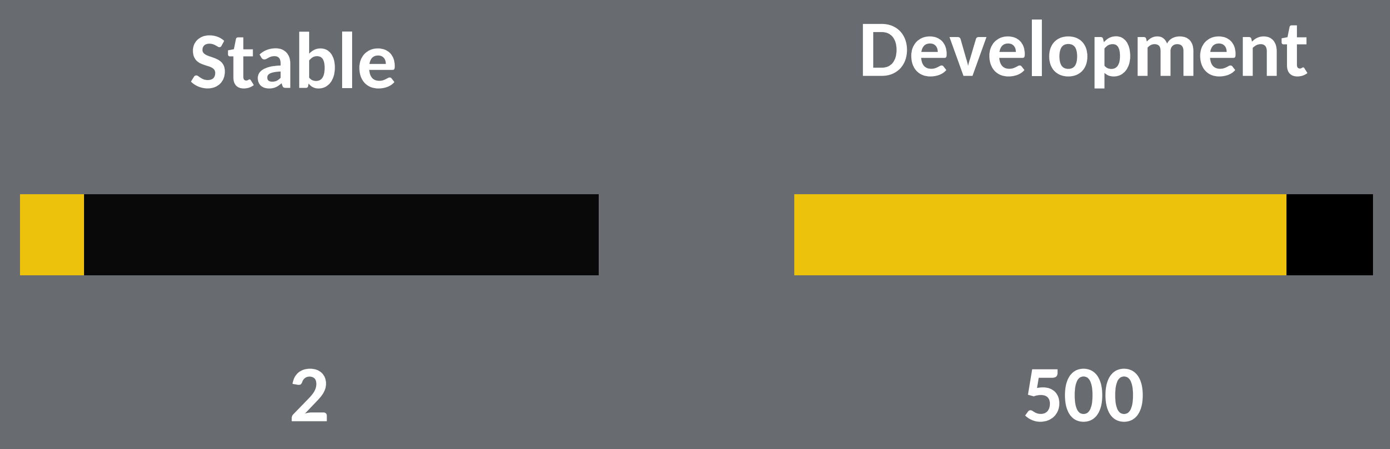
We currently have a backlog of changes to be integrated.

We already have a collection of code changes to be integrated. Most are code cleanup and minor bug fixes.

A cleanup and review of the ticket queue is needed. No showstoppers recently reported.

Better tracking these metrics would be useful.

Change Backlog



Stable

Development

2

500

Issues Tracking

02

Maintainers

Project Roles

Nomenclature we are using as we update the project process documentation on wiki.openafs.org

Of course, some people have multiple roles.

Release Manager

Security Officer

Maintainer

Contributor

System Administrator

New Maintainers

Andrew Deason

Michael Meffie

Andrew and Michael have been active OpenAFS developers for many years. They join Benjamin Kaduk as maintainers with commit access to the code base.

What is changing

Maintainer documentation

New maintainer guidelines for managing the change approval process.

Access rights granted

New maintainers have access to approve code changes for merge and to merge approved changes.

What is staying the same

Workflow

gerrit.openafs.org continues to be used to submit and review code. All changes are merged to the development branch, before being considered for the stable branch.

Release Team continues

The Release Team will continue to meet weekly to discuss and review changes.

What to expect

Contributor Guide

A new document for improved guidance on how to contribute code to OpenAFS.

Backlog Clearance

A renewed effort to review and accept existing change requests.

Faster Release Cadence

Our goal is to release fixes and improvements more quickly without reducing quality.

03

Project Infrastructure

03 Project Infrastructure

Systems Review

A review of the systems currently in use by the OpenAFS project and who are the owners and maintainers of these systems and services.

Source Code Repository

https://git.openafs.org

Git repositories and GitWeb code browser.

Provided by MIT

Benjamin Kaduk is currently the sole application administrator.

MIT operations manage virtual machine, network, storage.

Source Code Review

https://gerrrit.openafs.org

Git based, online source review system.

Provided by MIT

Benjamin Kaduk is currently the sole application administrator.

MIT operations manage virtual machine, network, storage.

Buildbot

https://buildbot.openafs.org

Buildbot continuous integration

Provided by MIT

Michael Meffie is currently the application administrator.

Application configuration is managed with Ansible and git.

MIT operations manage virtual machine, network, storage.

Project Web Site

https://www.openafs.org

Static HTML website with frames for navigation bars.

Provided by Grand.Central.org

Content is managed with Git using the Gerrit Code Review.

Homebrew Perl scripts used to generate release pages and deploy content.

Project Wiki

https://wiki.openafs.org

IkiWiki instance.

Provided by MIT

Content is managed with Git or web interface.

No reviews required.

Bug Tracker

Provided by
Grand.Central.org

https://rt.central.org

Request Tracker bug tracking system for regular bugs and embargoed security sensitive reports.

Mail Lists

Provided by

Grand.Central.org

https://lists.openafs.org

GNU Mailman mail lists manager.

GitHub mirror

https://github.com/openafs/openfs

GitHub public mirror of the OpenAFS code base.

Free tier public github project

Github project owner

Automatically mirrors git.openafs.org

Issues and pull requests are not used.

Domain Registration

Provided by
Grand.Central.org

openafs.org Domain Registration

04

Going Forward

04 Going Forward

New Contributor Guides

Updated documentation for developers.

Clearing backlogs

The new maintainer have already started accepting changes.

General objectives

Setting goals for Security, Reliability, Performance, and Usability improvements.

OpenAFS Contributor Guide

Up to date documentation for developers and maintainers.

Guidelines and procedures on how to contribute code and documentation changes.

Includes updated style and process guideline.

The contributor guide will be maintained in the code base and changes will be managed like code changes.

Security

Addressing vulerabilities is our highest priority.

The OpenAFS code base has a long history and many legacy features.

Improvements have been made over the years, but much more will be done modernize and harden the codebase.

Objectives

Timely security releases.

Unsafe interfaces will be hardened or disabled by default.

Correct unsafe threading and memory management.

Obsolete and unused code will be removed at a faster pace.

Applying principles of least privilege, (such as non-root file servers) and strong encryption over the wire.

Reliability

Correcting and reducing bugs is a high priority.

The OpenAFS code base has a long history and many legacy features.

Improvements have been made over the years, but much more will be done modernize and harden the codebase.

Objectives

Obsolete and unused code will be removed at a faster pace.

Clear the outstanding change and ticket backlog.

Performance

Performance and scalability changes need to be a focus.

The OpenAFS code base has a long history. Many of the limits are due to outdated assumptions.

Objectives

Modernize internal hashing in the file server and the cache manager.

Modernize database storage to vastly improve throughput.

Remove networking bottlenecks without breaking compatibility.

Usability

Improving the experience without compromising compatibility.

The OpenAFS code base has a long history.

Objectives

Modernize documentation for setup and usage.

Improve deployment experience.

Improve logging and error messages.

New server administration commands.

We are excited to build up the successful heritage of OpenAFS.

Thank you for participating.

Thank you

Michael Meffie
mmeffie@sinenomine.net