THE CODE

REGISTRY

Project: **Retail Software**

Report for code vault EPOS system v1.0.0

In this report we detail all of the insights we have generated for this version of your codebase.

Snapshot summary

15 Languages	**43** File types	**40,284** Total files
1.72 Code complexity	**1%** AI Quotient™	**265** Security issues
11 Components	**2,204** Code contributors	**26** Total GIT branches

$4,180,000 - $4,620,000
"Cost to Replicate"

Scroll down for more details ↓

01. Languages

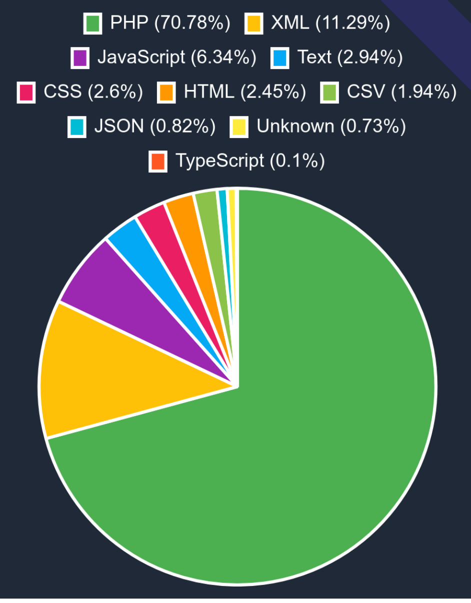
What programming languages did we find?

In total we detected **15** languages in your codebase. Below we lay out the most common programming languages based on the total number of lines of code in each.

Most Common Languages (#)

Language	# files	# lines of code
PHP	24,634	2,273,759
XML	9,521	362,601
JavaScript	1,463	203,648
Text	1,169	94,394
CSS	599	83,634
HTML	1,401	78,798
CSV	326	62,281
JSON	390	26,434
Unknown	474	23,520
TypeScript	1	3,238

Most Common Languages (%)



AI generated insight

Ada's thoughts...

Programming languages

Here are the 5 most common programming languages used in your project, listed from most to least used:

- **PHP:** PHP is a server-side scripting language mainly used for web development. It is versatile and can handle forms, generate dynamic page content, create cookies, and more.
- **XML:** XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
- **JavaScript:** JavaScript is a programming language that enables interactive web pages and is essential for web development. It can dynamically update content, control multimedia, animate images, and more.
- **Text:** Text files contain plain text without any special formatting. They are used for storing and exchanging data in a readable format.
- **CSS:** CSS is a style sheet language used for describing the presentation of a document written in HTML. It controls the layout, colors, fonts, and overall design of a webpage.

02. File types

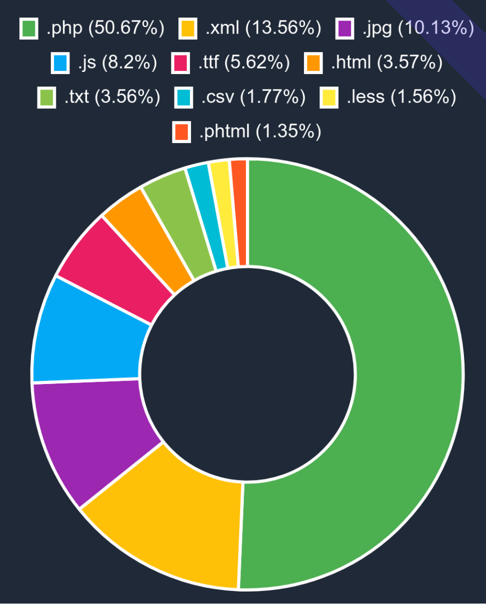
What file types are being used?

We found a total of **43** file types across **40,284** files with an overall size of **185.38 MB**. We're showing the most used file types below based on the total file size for each.

Most Common File Types (#)

File type	File size	File #
.php	88.42 MB	24,640
.xml	23.67 MB	9,500
.jpg	17.68 MB	65
.js	14.32 MB	1,465
.ttf	9.8 MB	11
.html	6.24 MB	426
.txt	6.22 MB	729
.csv	3.09 MB	326
.less	2.72 MB	513
.phtml	2.35 MB	977

Most Common File Types (%)



AI generated insight

Ada's thoughts...

File types

Here are the 5 most common file types in your project, listed from most to least used:

- **.php:** PHP files contain server-side code that helps your website or application function.
- **.xml:** XML files store and transport data in a structured format, commonly used for configuration.
- **.js:** JS files contain JavaScript code that adds interactivity and dynamic elements to your web pages.
- **.phtml:** PHTML files are a combination of PHP and HTML, often used in template files.
- **.txt:** TXT files are plain text files used for various purposes like storing notes or documentation.

03. Code complexity

How complex is your code?

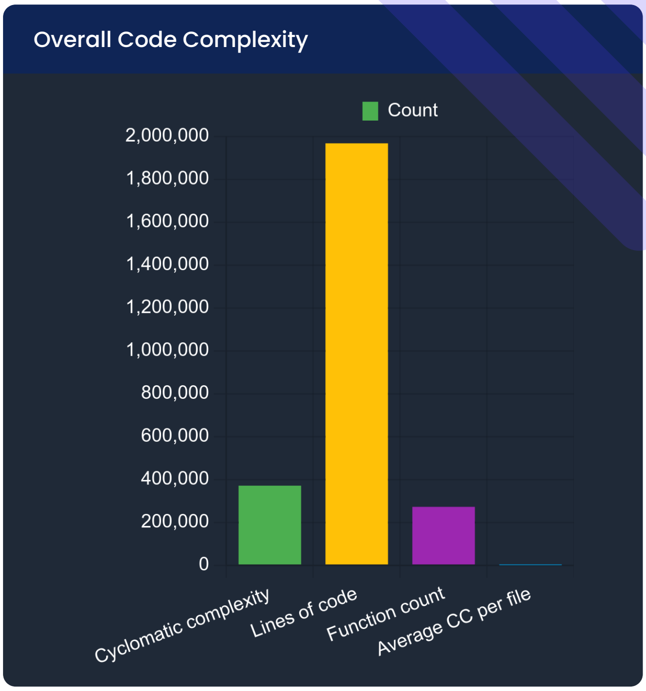
We've calculated an overall complexity score of **1.72**, taking into account many factors such as the number of languages and how complex the code is written in each.

We use a number of metrics to calculate how "complex" your code is. The main one is "Cyclomatic Complexity" or "CC".

> **"Cyclomatic complexity is a software metric used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code."** *Wikipedia*

CC - along with other metrics - are measured for every single file in your codebase to give us an average, which we use in our algorithms and calculations.

On this page you can see the overall average CC, and then the statistics for each language found in your code.


Overall Code Complexity

AI generated insight

Ada's thoughts...

Code complexity

Cyclomatic Complexity is a software metric used to measure the complexity of a program. A low score indicates simpler code with fewer decision points, while a high score suggests more intricate and complex code with many conditional statements and loops.

The overall Cyclomatic Complexity score for your project is 1.72, which falls in the low to moderate complexity range. This means that the codebase is relatively straightforward with a manageable level of complexity.

When we break it down by language:

- JavaScript (JS) has a Cyclomatic Complexity score of 0.94, indicating low complexity across 1452 files.
- TypeScript (TS) scores 1.84, showing slightly higher complexity with only 1 file.
- PHP has the highest complexity score of 2.04, with a large codebase spread across 24508 files.

It's important to note that Cyclomatic Complexity is just one factor we consider when evaluating the complexity of your code. It provides insights into the structure and potential maintenance challenges, helping us ensure code quality and readability.

04. AI Quotient™

How much of your code could be improved by AI?

We've scanned your code for common programming bad practises and coding issues and calculated that **1%** of your codebase could be improved by AI.

In the age of AI this is a burning question in the minds of many. But how do you quantify this data to come up with an answer?

At The Code Registry, we are in a prime position - with our proprietary code intelligence and analytics engine - to be give you this information. And we're the first in the World to do it!

Our platform analyses your entire codebase for common bad practices, coding quality and structure issues that we know most generative AI coding models are perfectly suited to improve. We can then compare this against other data we've analysed about your code and come up with the AI Quotient™ you see on this page.

AI Quotient™ summary

1%
Can be improved by AI

Total issues found	3,839
Affected lines of code	18,426
Total lines of code	3,212,493

AI could improve 1% of your code

What types of coding issues did we find?

Login to our web app to see more details (including code snippets and file paths)

Issue type	Number of issues	Total lines of code affected
Insecure URL	1,482	1,482
Weak RNG Usage	69	69
Suspicious Comment	475	475
Outdated OSS Component	8	13,853
Debug Code in Production	570	570
Hardcoded Elliptic Curve	2	2
Weak Cipher Mode Detected	10	10
Replace mcrypt with OpenSSL	32	32
Unsafe Data Deserialization	383	383
Known Security Vulnerability	265	728
Sensitive Data in Source Code	130	130
Use of Eval on Untrusted Data	13	71
Weak or Broken Hash Algorithm	205	205
Avoid Low-Entropy Content Hashing	36	36
Review setTimeout for Data Security	159	380

AI generated insight

Ada's thoughts...

AI Quotient

The AI Quotient of your codebase is 0.57. This percentage indicates the level of common bad practices, coding quality, and structure issues that AI has detected in your code.

Our system found a total of 3839 issues within your codebase, affecting 18426 lines of code out of a total of 3212493 lines.

If the AI Quotient is over 50%, it suggests there are numerous issues that should be addressed to prevent technical debt. We recommend reviewing these issues with your development partner or considering engaging an AI Code Generation company to help improve the quality of your code.

05. Security issues

Did we detect any security issues?

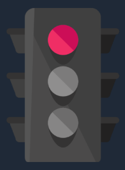
The Code Registry system scans your codebase and any detected third party dependencies to find any potential security vulnerabilities. If you don't see any records below - congratulations! There are possibly no known vulnerabiltiies in your code. If you see records below, you should review with your development team or partner.

In the latest scan, we scanned 40,284 files with 4,051 rules. There are 265 findings which you can review below.

Understanding security vulnerabilities in your codebase is essential to prevent malicious attacks, data breaches, and unauthorized access. These vulnerabilities can lead to serious legal and financial consequences, including loss of customer trust and non-compliance penalties. They may also disrupt services, causing downtime and productivity loss.

Remediation typically involves patching, code modification, configuration changes, or new security controls. If immediate fixes aren't possible, implement temporary mitigations. Regular software updates, secure coding practices, and thorough testing are key to preventing vulnerabilities, while a robust incident response plan is crucial for effectively managing security breaches.

Security Status



We've given your code a red traffic light in it's current version. This is because there is at least one urgent issue found in our scans. You can see more details by clicking "More info" and review these with your development team.

Once an issue is resolved, you can manually update the code from the original code source using the "Update Code" button in our web app. The Code Registry will also automatically update your code vault based on your schedule and send you an updated report.

AI generated insight

Ada's thoughts...

Security vulnerabilities

After analyzing the security of your codebase, we found the following issues:

- **INFO severity:** 1 issue
- **ERROR severity:** 91 issues
- **WARNING severity:** 173 issues

Urgent issues were detected in the file app/code/Magento/CardinalCommerce/Test/Unit/Model/JwtManagementTest.php:

- Line 116: JWT token detected (Severity: ERROR)
- Line 130: JWT token detected (Severity: ERROR)
- Line 145: JWT token detected (Severity: ERROR)

We recommend that you visit the dedicated security page for more detailed information on these vulnerabilities.

Dependency vulnerabilities

Your project has several security vulnerabilities in its open source dependencies. Here is a summary of the issues:

- INFO severity: 1 issue
- ERROR severity: 91 issues
- WARNING severity: 173 issues

Urgent issues have been found in the following file:

- File: app/code/Magento/CardinalCommerce/Test/Unit/Model/JwtManagementTest.php
- Lines: 116, 130, 145
- Issue: JWT token detected (Severity: ERROR)

Please visit the dedicated security page for more detailed information on these vulnerabilities.

10 most urgent code issues

Login to our web app to see more details

lib/internal/Magento/Framework/Filesystem/Io/Ftp.php

ERROR

Line 293

FTP allows for unencrypted file transfers. Consider using an encrypted alternative.

```
                                                 return @ftp_rename($this->_conn, $src, $dest);
```

lib/internal/Magento/Framework/Encryption/Adapter/Mcrypt.php

ERROR

Line 65

Mcrypt functionality has been deprecated and/or removed in recent PHP versions. Consider using Sodium or OpenSSL.

```
                                                    $initVectorSize =
@mcrypt_enc_get_iv_size($this->handle);
```

lib/web/scriptaculous/effects.js

ERROR

Line 1053

User controlled data in methods like `innerHTML`, `outerHTML` or `document.write` is an anti-pattern that can lead to XSS vulnerabilities

```
                                            String.__parseStyleElement.innerHTML = '<div
style="' + this + '"></div>';
```

lib/internal/Magento/Framework/System/Ftp.php

ERROR

Line 304

FTP allows for unencrypted file transfers. Consider using an encrypted alternative.

```
                                                                     @ftp_close($this->_conn);
```

lib/internal/Magento/Framework/Encryption/Crypt.php

ERROR

Line 65

Mcrypt functionality has been deprecated and/or removed in recent PHP versions. Consider using Sodium or OpenSSL.

```
                                                                     $handle = @mcrypt_module_open($cipher, '',
$mode, '');
```

lib/internal/Magento/Framework/System/Ftp.php

ERROR

Line 385

FTP allows for unencrypted file transfers. Consider using an encrypted alternative.

```
                                                                return @ftp_rawlist($this->_conn, $dir,
$recursive);
```

lib/internal/Magento/Framework/System/Ftp.php

ERROR

Line 136

FTP allows for unencrypted file transfers. Consider using an encrypted alternative.

```
                                                                $this->_conn = ftp_connect($params['host'],
$port, $timeout);
```

lib/web/knockoutjs/knockout-fast-foreach.js

ERROR

Line 51

User controlled data in methods like `innerHTML`, `outerHTML` or `document.write` is an anti-pattern that can lead to XSS vulnerabilities

```
                                                            parentNode.innerHTML = sourceNode.text;
```

app/code/Magento/Integration/Test/Unit/Oauth/OauthTest.php

ERROR

Line 826

Generic Secret detected

```
'oauth_token_secret="a6agsfrsfgsrjjjjyy487939244ssggg",' .
```

lib/internal/Magento/Framework/System/Ftp.php

ERROR

Line 293

FTP allows for unencrypted file transfers. Consider using an encrypted alternative.

```
                                    return @ftp_pasv($this->_conn, (bool)$pasv);
```

10 most urgent issues with your dependencies

Login to our web app to see more details

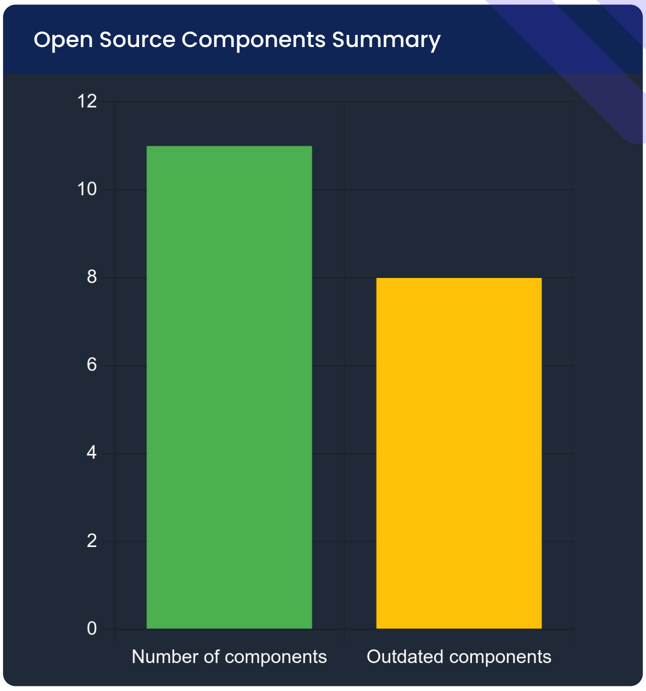
06. Open Source Software Components

What third party packages and dependencies are being used?

Our system scans your code for any references to third party dependencies and packages. We've found a total of 11 open source components in the current version of your code. We list some of these below, but a lot more information can be found in our web app.

Open source software components are like pre-made building blocks for software, widely used for their efficiency. They're publicly shared code pieces that anyone can use and modify, often integrating them into larger, custom projects.

While beneficial, they can pose challenges if not carefully managed. Keeping them updated is crucial for security and performance. Using many different components can lead to maintenance complexities and potential security vulnerabilities.


Open Source Components Summary

20 largest open source components by lines of code

Login to our web app to see more details (including urls and license information)

Component	Vendor	Size	Version
kora	robertu	Total files 82 Total lines of code 6,860	Current version: 0.0.1 Latest version: 0.0.3 Outdated
Service-Hub	JovianX	Total files 46 Total lines of code 4,778	Current version: 0.6.0 Latest version: 0.9.1 Outdated
neighborhood-association	badawi1713	Total files 7 Total lines of code 1,172	Current version: 1.0.0 Latest version: 1.2.0 Outdated
apiraiser	horizech	Total files 8 Total lines of code 601	Current version: 0.5.0 Latest version: 1.0.0 Outdated
qubryx-react-component	Bibin	Total files 12 Total lines of code 453	Current version: 0.1.0 Latest version: 0.1.0 Up to date

Component	Vendor	Size	Version
material-ui	mui	Total files 4 Total lines of code 290	Current version: 5.10.10 Latest version: 5.13.5 Outdated
tic-tac-tic-tac-toe	jwngr	Total files 1 Total lines of code 131	Current version: 8957029 Latest version: 8957029 Up to date
rage-ts	rage	Total files 1 Total lines of code 109	Current version: 0.2.18 Latest version: 0.4.18 Outdated
atom	file-icons	Total files 1 Total lines of code 30	Current version: 1.6.13 Latest version: 1.7.15 Outdated
uzimakeke-test	uzimainc	Total files 1 Total lines of code 18	Current version: 1.0.0 Latest version: 1.0.0 Up to date
cra-template-pwa	Ian Sutherland	Total files 1 Total lines of code 13	Current version: 1.0.1 Latest version: 2.0.0 Outdated

What licenses did we find in your third party packages and dependencies?

Any third party package can potentially have it's own license which dictates how you can use it. It's important to know about these because you could be breaking compliance with a third party package without realising it. We've found a total of 8 licenses in your packages and dependencies. We give a bit more info below but you can find a lot more in our web app.

Commercial licenses

Type	Count	More info
No commercial licenses found		

Open source licenses

Type	Count	More info
MIT	5	Link
AGPL-3.0-only	1	Link
GPL-1.0-or-later	1	Link
AGPL-3.0-or-later	1	Link

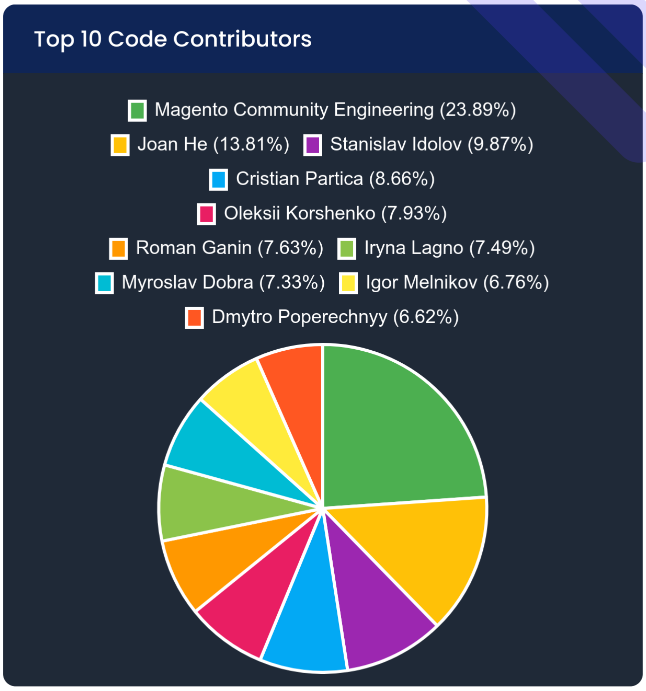
07. GIT history analysis

Who's been working on your code?

We interrogate the data we find in your original GIT code source. We can normally find the full change history from the very first change, including the names of code contributors, what exact files were changed and what messages (if any) were left by the developers. In the current version of your code we've found 2,204 code contributors.

Understanding who has contributed to your codebase helps in maintaining the security and integrity of your code. It's crucial to ensure that those who have access to and modify your code are trusted and authorized individuals. This knowledge aids in preventing unauthorized access and potential malicious changes.

In the event of a problem or a bug, knowing who worked on specific parts of the code makes it easier to address issues quickly and efficiently. This information is not just about security; it's also about enhancing collaboration, ensuring quality, and managing your team's dynamics effectively.


Top 10 Code Contributors

20 most recent changes by any contributor

Login to our web app to see more details (including exact files which have been worked on)

Change Message	Date / Time	Contributor	Changed Files
Merge pull request #9613 from magento-gl/marcomprs	2025-03-05 08:45:29	internal-magento-queue-manager[bot]	Files changed: 1 Insertions: 0 Deletions: 0
Merge remote-tracking branch "39601/bug/issue-39568-reorder-bug" into marcomprs	2025-03-05 02:50:39	engcom-Charlie	Files changed: 1 Insertions: 0 Deletions: 0
Merge branch "2.4-develop" into marcomprs	2025-02-28 15:52:39	magento-devops-queue-mgr-svc	Files changed: 1 Insertions: 0 Deletions: 0
Merge pull request #9513 from magento-gl/aprilrelease	2025-02-28 15:51:33	internal-magento-queue-manager[bot]	Files changed: 1 Insertions: 0 Deletions: 0
AC-13735 Change dependencies in 2.4-develop branch to github repos instead of packages on the	2025-02-28 05:38:43	Rajesh Kumar	Files changed: 2 Insertions: 37

Change Message	Date / Time	Contributor	Changed Files
repo.magento.com			Deletions: 19
AC-13735 Change dependencies in 2.4-develop branch to github repos instead of packages on the repo.magento.com	2025-02-27 12:21:50	Rajesh Kumar	Files changed: 1 Insertions: 2 Deletions: 2
Merge remote-tracking branch "origin/2.4-develop" into aprilrelease	2025-02-27 12:19:51	Rajesh Kumar	Files changed: 1 Insertions: 0 Deletions: 0
Merge branch "2.4-develop" into marcomprs	2025-02-27 11:52:27	magento-devops-queue-mgr-svc	Files changed: 1 Insertions: 0 Deletions: 0
Merge pull request #9602 from magento-gl/functional-mainline-deployment-branch	2025-02-27 11:51:37	internal-magento-queue-manager[bot]	Files changed: 1 Insertions: 0 Deletions: 0
Merge remote-tracking branch "origin/AC-14039" into aprilrelease	2025-02-27 11:16:12	Rajesh Kumar	Files changed: 1 Insertions: 0 Deletions: 0
Merge remote-tracking branch "origin/AC-13735" into aprilrelease	2025-02-27 11:10:19	Rajesh Kumar	Files changed: 1 Insertions: 0 Deletions: 0
Revert "AC-13793::Removed lusitanian/oauth package as laminas-oauth is migrated"	2025-02-27 11:07:54	Rajesh Kumar	Files changed: 2 Insertions: 88 Deletions: 22
AC-13735::Change dependencies in 2.4-develop branch to github repos instead of packages on the repo.magento.com	2025-02-27 09:26:40	Rajesh Kumar	Files changed: 1 Insertions: 12 Deletions: 0
AC-13735::Change dependencies in 2.4-develop branch to github repos instead of packages on the repo.magento.com	2025-02-27 08:08:10	Rajesh Kumar	Files changed: 1 Insertions: 0 Deletions: 12
AC-14039::[Integration Test] GraphQlCheckoutMutationsStateTest	2025-02-27 07:39:54	Dnyaneshwar Jambhulkar	Files changed: 2 Insertions: 1 Deletions: 2
ACQE-7276 | Mainline PR deployment	2025-02-27 07:29:01	manjusha729	Files changed: 1 Insertions: 1 Deletions: 0
Merge remote-tracking branch "39371/issue38891" into marcomprs	2025-02-27 06:20:10	engcom-Charlie	Files changed: 1 Insertions: 0 Deletions: 0
Merge remote-tracking branch "31231/MTWO-5" into marcomprs	2025-02-27 06:09:28	engcom-Charlie	Files changed: 1 Insertions: 0 Deletions: 0
AC-13735::Change dependencies in 2.4-develop branch to github repos instead of packages on the repo.magento.com	2025-02-27 06:05:22	Rajesh Kumar	Files changed: 4 Insertions: 39 Deletions: 252
Merge remote-tracking branch "origin/2.4-develop" into AC-13735	2025-02-27 06:02:49	Rajesh Kumar	Files changed: 1 Insertions: 0 Deletions: 0

08. GIT branch analysis

How are your code contributors managing GIT branches?

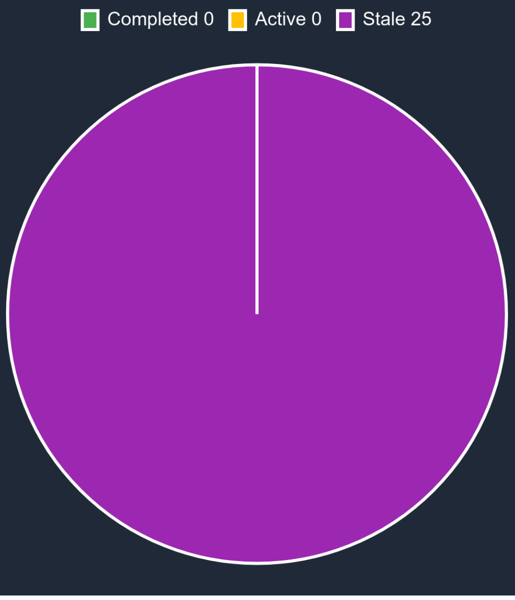
GIT branches help teams work on features and fixes in parallel. Understanding your branch metrics reveals your team's development patterns and identifies potential bottlenecks.

We analyze every branch in your repositories to calculate key metrics like age, activity status, and authorship. This data helps identify stale branches that need cleanup and shows how effectively your team uses branching strategies.

Branch analysis

Total Branches	26
Most Active Branch	2.4-develop (150779 commits)
Most Stale Branch	2.0 (2569 days)
Average Branch TTM (Time To Merge)	0 days
Merged Branches	9
Most Common Merge Target	2.4-develop
Average Changes per Branch	3033.3 files, +46359 / -30665.7 lines
Branch Types	Feature: 0, Bugfix: 0, Hotfix: 0, Other: 26

Branches by status



AI generated insight

Ada's thoughts...

Branch analysis

To understand the branch analysis data, let's start with some basics:

- **GIT:** GIT is a version control system used by developers to track changes in their code.
- **Branch:** A branch in GIT is a separate line of development that allows developers to work on features, fixes, or experiments without affecting the main codebase.

Now, looking at the branch analysis data:

- **Total Branches:** There are 26 branches in the project, each serving a different purpose.
- **Most Stale Branch:** The branch named '2.0' has not been updated for 2569 days, indicating it may need attention.
- **Most Active Branch:** The branch '2.4-develop' has seen 150779 commits, showing it's a key area of ongoing development.
- **Average Branch Size:** On average, branches have added 46359 lines, changed 3033 files, and deleted 30665 lines, giving an idea of the scope of work in each branch.

- **Total Merged Branches:** 9 branches have been successfully integrated back into the main codebase, ensuring completed work is included.
- **Branches Merged Multiple Times:** 3 branches have been merged more than once, possibly indicating complex development tasks.

Overall, this data provides insights into the project's development activity, highlighting areas that may need attention, active branches driving progress, and successful integration of completed work back into the main codebase.

09. Cost to Replicate

What might it cost to replicate your IP?

The Code Registry is the only platform that estimates the replication value of your code.

Across many factors and data points, we estimate your "Cost to Replicate" valuation at $4,180,000 - $4,620,000

The most important data points in our calculation are about how complex your code is, which includes how many languages that it uses and how "complex" the code is for each language. We also maintain our own database of all programming languages and what their average going market rate is for a developer, plus other data points that help us in our estimation.

These are just some of the many factors that we use to calculate a very "practical" valuation of your codebase. I.E. based on what we know about your code and how long we estimate it would take for a developer to recreate it entirely from scratch, this is a valuation we give it.

"Cost to Replicate" Summary

Total lines of code	3,174,518
Lines of code discounted	14,455
Languages multiple	1.2
Complexity multiple	1
The Code Registry "Cost to Replicate" Value	**$4,180,000 - $4,620,000**

AI generated insight

Ada's thoughts...

Replication valuation

The Replication Valuation is a practical estimation of the cost to recreate your entire project from scratch. It takes into account factors like the total lines of code in each language, complexity of those languages, and current market rates for developers.

Your project's total replication valuation falls between $4,180,000 and $4,620,000. This means that if you were to start from scratch today, this would be the approximate cost to rebuild your project.

Among the languages used in your codebase, 'PHP' contributes the most, with a valuation range of $2,945,000 to $3,255,000. 'JavaScript' follows with a contribution of $304,000 to $336,000, and 'CSV' contributes $66,500 to $73,500.

To explore more details and implications, we recommend visiting the dedicated replication valuation page for further information.