

Reading Project Analytics for Documentation Decisions

August 25, 2026

Opening the project analytics view and identifying the date range you are reviewing

To make good documentation decisions in Atloria, start from the project's analytics area rather than relying on a single document or page view. This keeps your reading focused on the full documentation set for that project, including guides, references, and release-related content. If you need a refresher on the general analytics workflow inside a project, see [Analyzing Project Performance and Activity](#).

1. Open the project you want to review from your project list or project workspace.
2. Check the project header or project selector before reading any numbers. Make sure you are looking at the correct project, especially if you switch between multiple documentation workspaces during the day.
3. Open the **Analytics** view for that project. Stay in the project-level analytics screen so the totals reflect the project as a whole, not just one page.
4. Find the date-range control and select the reporting period you want to review. Choose the time window that matches the decision you are making, such as a recent release period, a launch week, or a longer trend window.
5. Look for any comparison setting that shows the current period against an earlier period. If a comparison is turned on, note that the screen may show both current values and change indicators.
6. Check whether any filters are active for version, environment, or similar scope controls. Before you compare results, confirm you are not mixing draft content with released documentation or reviewing a staging set when you meant to review published content.
7. Pause before interpreting the charts and cards. Read the filter bar and date controls one more time so you know exactly what content and time period the screen is showing.

[SCREENSHOT: Project analytics view with the project header, date-range selector, and active filters highlighted]

Reading the headline metrics that signal project performance

The summary cards at the top of the analytics view give you the fastest read on how a project is performing. In Atloria, treat these cards as project-wide signals. They help you decide whether your documentation is attracting readers, keeping their attention, and supporting release activity over time.

When you review headline numbers such as **Total Views**, **Unique Visitors**, **Average Time on Page**, and bounce or exit-related indicators, read them together instead of one at a time. A rise in **Total Views** can look positive on its own, but if **Average Time on Page** drops sharply during the same period, the project may be reaching more people while helping fewer of them find what they need. In the same way, a healthy number of **Unique Visitors** can mean new discovery, but it does not automatically mean the content is useful.

1. Start with the top metric cards and read the current values first.
2. Check the change indicator on each card to see whether the number is up or down compared with the previous period.
3. Compare traffic signals and engagement signals side by side. For example, pair **Total Views** with **Average Time on Page** instead of reading each card in isolation.
4. Watch for release-week distortion. A launch can create a short-term traffic spike that makes the project look stronger than it really is over a longer period.
5. Look for repeated movement across several cards. If views, visitors, and engagement all improve together, that usually points to a meaningful change in project performance.
6. Be cautious when only one card changes dramatically. A single spike may reflect a release announcement, a migration deadline, or a temporary burst of attention rather than a lasting improvement.

A good reading of the headline area should answer one question: is this project gaining useful attention, or just temporary traffic?

Using traffic and engagement views to find what content is helping or hurting users

After the headline cards, move into the traffic and engagement tables. This is where Atloria becomes most useful for documentation decisions because you can see which pages are drawing people in, which ones hold attention, and where readers leave.

1. Open the table or list that shows top pages, top content, or similar page-level performance results inside the project analytics view.
2. Sort by **Views** first to identify the pages that attract the largest share of project traffic. These are often your most visible guides, release notes, onboarding pages, or reference content.
3. Sort by **Unique Visitors** next. This helps you separate pages with broad reach from pages that are revisited by the same group of readers.
4. Then sort by **Average Time on Page** to find pages where readers spend meaningful time. These often include setup guides, migration instructions, and detailed technical explanations.
5. Review entry-page data to see where readers begin. If many visitors start on a troubleshooting page, that may mean users are finding help through search before they ever reach your main navigation.
6. Review exit-page data to see where readers stop. A high exit rate on a setup or onboarding guide can signal confusion, missing steps, or weak follow-up links.
7. Compare low-engagement pages against what those pages are supposed to do. A short visit on a quick answer page may be fine. A short visit on a complex migration guide usually deserves attention.

Look for patterns, not isolated pages. If several setup guides show high exits, or several troubleshooting articles show very short reading time, that points to a broader documentation problem rather than a single weak page.

[SCREENSHOT: Top pages table with sorting controls for Views, Unique Visitors, and Average Time on Page]

Comparing project trends across releases, launches, and documentation updates

Project analytics become much more valuable when you compare one period against another. In Atloria, comparison views help you see whether a release, content update, or navigation change created a temporary spike or a lasting improvement.

1. Open the comparison option in the project analytics view.
2. Set the current reporting window to the release, launch, or update period you want to evaluate.
3. Choose a comparison period that makes sense for the question you are asking. A previous release window works well for release comparisons, while an earlier launch period can help you compare adoption patterns.

4. Review the trend lines and change indicators across the same set of metrics, rather than switching between unrelated views.
5. Match spikes and dips to known events in your team's timeline, such as a product launch, a migration deadline, a documentation restructure, or a major update to navigation.
6. Check whether release-note traffic leads readers into other content afterward. A strong release usually creates an initial spike in release notes, then continued visits to task guides, upgrade instructions, or reference pages.
7. Compare before-and-after performance when you change page titles, reorganize content groups, or adjust navigation labels. If discovery improves after those changes, the trend should show up across the project rather than on only one page.

This kind of comparison helps you avoid overreacting to launch-week excitement. A page that surges for two days and disappears from the top results tells a different story from a guide that keeps drawing readers for weeks after release. Use the analytics screen to separate short-term attention from durable documentation value.

Turning analytics signals into documentation and release decisions

Once you have read the project analytics clearly, use those signals to decide what to update, protect, expand, or retire. Atloria is most helpful when you connect the numbers to concrete documentation work instead of treating analytics as a reporting exercise.

1. Start with consistently high-traffic pages. Put these pages near the top of your maintenance list before each release. Review their screenshots, task steps, navigation labels, and version accuracy because they are likely to affect the largest number of readers.
2. Look for troubleshooting, migration, or onboarding pages that receive repeated search-driven traffic. These pages often reveal friction in the product experience or gaps in your core guidance. Improve cross-links, add clearer task steps, or strengthen the related onboarding content.
3. Protect low-traffic pages that show strong engagement. A specialist guide may not appear in the top traffic group, but if readers spend meaningful time there, it may be critical for advanced users or release teams.
4. Use repeated patterns to decide when a page should be split. If a very long page gets heavy traffic but weak engagement, readers may be struggling to find the

right section quickly.

5. Retire or merge outdated content only after checking project-level trends across more than one period. A page with low traffic during one window may still matter during release cycles or migration events.
6. Feed your findings into release planning. Pages with sustained traffic should be validated before launch, while pages with high exits may need revision before the next version goes live.

If you are deciding what to improve first, combine traffic, engagement, and release timing. The strongest candidates are usually pages that are both widely used and closely tied to upcoming product changes.

Avoiding common mistakes when interpreting project analytics

Analytics are useful only when you read them in context. In Atloria, a few common mistakes can lead to the wrong documentation priorities, especially when teams move quickly between projects, releases, and audience groups.

- Do not compare two projects by raw **Views** alone if they serve different audiences. A broad public project and a specialist internal project will naturally show different traffic levels.
- Do not assume a short **Average Time on Page** always means failure. Quick-reference pages, short release notes, and direct-answer content often do their job in a short visit.
- Do not make structural changes until you confirm the date range, active filters, and comparison settings at the top of the analytics view.
- Do not mix draft, staging, and released documentation when reviewing project performance. If version or environment filters are active, verify them before drawing conclusions.
- Do not treat launch-week spikes as proof of long-term content success. Check whether readers continue into setup guides, upgrade instructions, or reference pages after the initial burst.
- Do not remove specialist content just because it has lower traffic. Some pages support a smaller audience but still deliver strong value, especially when engagement is high.
- Do not change staffing or roadmap priorities based on one short reporting window. Confirm the same pattern across multiple periods before making larger planning

decisions.

A careful reading usually means slowing down for a minute at the top of the screen. If the project name, date range, and filters are wrong, every conclusion below them will be wrong too.

Overview

Use this guide when you need to turn project analytics into documentation decisions inside Atloria. The focus here is not on learning what analytics are in general, but on reading the project-level analytics view in a practical way so you can decide what content to maintain, improve, expand, or review before a release.

This document assumes you are already familiar with the basic project analytics area and want to go further by interpreting trends and acting on them. If you need the broader walkthrough of project analytics screens and common views, return to [Analyzing Project Performance and Activity](#).

What this guide helps you do:

- Open the correct project analytics view and confirm the reporting scope
- Read headline metrics such as **Total Views**, **Unique Visitors**, and **Average Time on Page**
- Use top-page and engagement views to spot strong and weak content
- Compare current performance with earlier release or launch periods
- Turn analytics patterns into documentation and release decisions
- Avoid common interpretation mistakes that lead to poor content changes

This guide stays at the project level. It is meant for reviewing the overall documentation set for one project rather than judging a single page in isolation. That makes it especially useful when you are preparing for a release, reviewing the impact of a documentation restructure, or deciding where your team should spend update time.

[SCREENSHOT: Project analytics landing view showing summary cards and page-performance sections]

Prerequisites

Before you use this workflow in Atloria, make sure the following basics are in place:

- You can sign in to Atloria and open the project workspace you want to review

- You know which project you are evaluating and can confirm it from the project header or project selector
- You have access to the project's **Analytics** view
- The project has enough documentation activity to show meaningful trends in the analytics screen
- You know the time period you want to review, such as a recent release window, launch period, or ongoing monthly trend
- You understand whether you need to review released documentation only or compare it with another version or environment
- You are familiar with the project's recent changes, such as new releases, migration deadlines, navigation updates, or major content revisions

It also helps to have these reference points ready before you begin:

- A list of recent releases or launches you want to compare
- Awareness of major documentation updates that may affect traffic patterns
- Agreement within your team on what decision you are trying to make, such as improving onboarding, validating release readiness, or identifying pages that need maintenance first

For the next step in this learning path, continue to [Using Project Activity Signals to Improve Release Planning](#).