ATLORIA

Understanding Version Review Feedback and Follow Up

August 25, 2026

Opening a version review and reading the outcome

In Atloria, start from the **version details** screen for the version you submitted for review. This is the best place to understand the current decision because it brings together the **review status**, the **discussion area**, and any **comparison view** used during review. If you already worked through review decisions in Managing Version Review Requests and Decisions, use that same version record and focus on the feedback attached to it.

Look first for the **review status indicator** on the version. This status tells you the overall outcome at a glance, such as **Approved**, **Changes Requested**, or **Rejected**. Treat that badge as a starting point, not the full answer. A version can show a clear status while the real detail lives in the reviewer conversation.

Open the **review panel** or **comment thread** and read each reviewer note carefully. Reviewers may explain why they approved the version, what still needs work, or what prevented approval. If several people reviewed the same version, read all comments before deciding what to change.

Pay attention to where each comment appears:

- Some comments apply to the **entire version**
- Some are attached to a **specific section**
- Some are tied to **highlighted content** in the **diff** or comparison view
- Some point to **changed text** only, which helps you focus on what was updated since the last version

When feedback is attached to a highlighted area, open the related comparison or changed-content view so you can see the exact wording under review. This is especially useful when a reviewer comments on a small edit that is easy to miss in the full document.

[SCREENSHOT: Version details page showing review status badge, comment thread, and comparison area]

Understanding what reviewers are asking you to change

Once you have read the outcome, sort the feedback by importance. In Atloria, the most urgent comments are the ones that clearly block release. These usually appear when the version status is **Changes Requested** or **Rejected**, but you should still read the wording in the comment thread to understand the exact expectation.

Look for signs that a comment is **blocking**:

- The reviewer says the version is **not ready**
- The comment points out an **accuracy problem**
- The reviewer identifies **missing information**
- The note asks for a correction before approval can continue
- The review outcome itself is **Changes Requested** or **Rejected**

Other comments may be helpful but not release-blocking. These often read more like suggestions, wording improvements, or style preferences that can be handled later if your team agrees.

Use the context around each comment to understand what needs to change. Atloria may show feedback beside highlighted text, within a specific content area, or against a changed section in the version comparison. That context matters. A short comment like “clarify this” is much easier to interpret when you can see the exact paragraph or changed line it refers to.

As you review the thread, group similar comments together. Repeated themes often point to a larger issue, such as:

- **Accuracy** problems across several sections
- **Style** or wording corrections throughout the version
- **Missing details** in setup, release notes, or feature explanations
- Gaps between the updated content and the reviewer’s expectations

Also check whether the reviewer expects:

- a **direct edit** to the content
- a **reply in the thread** with clarification
- a **revised version** for another review pass

If the request is not obvious, do not guess. Use the existing comment thread to confirm what the reviewer wants before changing the version.

Responding to requested changes and rejections

When a version comes back with **Changes Requested** or **Rejected**, respond in the review conversation before you start editing. A short reply in the existing **comment thread** helps reviewers see that you received the feedback and are acting on it. This is especially helpful when several reviewers are involved or when the version has multiple open comment threads.

Your reply should make the next action clear. For example, you might note that you are updating the affected section, checking related pages, or preparing a revised version for another review. Keeping that note in the same thread creates a visible record on the version details screen.

After that, make the required updates in the draft or revised version. The right path depends on how your team handles version changes in Atloria, but the key point is the same: update the actual content that the reviewer flagged, not just the discussion. If the reviewer rejected the version because of larger issues, review the full set of comments before editing so you can address everything in one pass.

As you complete each fix, return to the related comment thread and verify that the content now matches the request. Only then should you mark that conversation as resolved, if that option is available on the review screen. Avoid resolving comments too early, because unresolved threads are often the easiest way for reviewers to track what still needs attention.

Before asking for another review, add a clear follow-up note that summarizes what changed. Keep it specific:

- which sections were updated
- which comments were addressed
- whether any suggestions were deferred
- whether the version is ready for re-check

That summary saves reviewers time and reduces back-and-forth during the next review cycle.

[SCREENSHOT: Review thread with author reply, updated comment status, and response summary]

Coordinating follow-up with reviewers before release

After you finish the edits, bring the right people back into the review. In Atloria, use the **reviewer assignment**, **mentions**, or any available **notification** controls on the version discussion area to direct attention to the updated version. This is more reliable than assuming reviewers will notice the changes on their own.

Focus first on the people whose approval is required for release. If your version has multiple reviewers, check whether all assigned reviewers have responded or whether some are still marked as pending. A version may look nearly complete while still waiting on one required decision.

The version details screen is the best place to track follow-up. Review it for signs that work is still open:

- **unresolved comment threads**
- a **pending review** indicator
- reviewers who have not yet responded
- earlier decisions that may no longer apply after edits

If you updated the version after feedback, mention that directly in the discussion area so reviewers know what changed since their last visit. A short note such as “updated the installation section and corrected the release notes based on review comments” gives reviewers a clear starting point.

It also helps to record release readiness in the same visible area. Use the version discussion or approval area to leave a concise note about the current state, such as whether all requested changes are complete, whether any non-blocking suggestions remain, and whether the version is being held for one final check. This creates a shared decision trail for everyone involved in the release.

If you need a refresher on broader approval handling, return to Managing Version Review Decisions and Approvals. For follow-up work, stay focused on the open comments and current approval state shown on the version itself.

Deciding when a version is ready to move forward

A version is ready to move forward only when the current version record shows that review work is complete. In Atloria, do not rely on memory or earlier comments alone. Open the latest version and confirm that the visible review state matches the content that is actually on screen.

Start by checking that all accepted feedback has been applied to the latest version. Review the updated sections, then compare them against the open and resolved

comment threads. Make sure there are no unresolved comments that still refer to active problems. If a comment is outdated because the content changed, verify that the latest wording truly fixes the issue before treating that thread as complete.

Next, check the **final approval state** again after resubmission. This matters because earlier approvals may no longer represent the current version if substantial edits were made afterward. A version that was once approved may need another review pass if important sections changed.

Use the current review outcome to guide your decision:

- Move toward release when the version shows the needed approval state and no blocking feedback remains
- Request another review cycle when major edits were made after the last decision
- Hold the version when comments are still unresolved, reviewers are still pending, or the outcome does not support release

Be especially careful after large revisions. If the version content changed significantly between review rounds, make sure the release decision reflects the newest draft, not an older approval. The safest habit is to treat the version details page as the source of truth for readiness.

The next step in this workflow is Preparing Versions for Final Approval, where you move from review follow-up into final release signoff.

Fixing common problems during review follow-up

Most review follow-up problems come from looking at the wrong revision, missing an open thread, or assuming an earlier decision still applies. In Atloria, the fastest way to troubleshoot is to return to the **version details** screen and check the current review state, discussion area, and comparison view together.

If comments seem outdated after edits, first confirm that you are viewing the **latest version**. Then open the **comparison view** and make sure you are comparing against the correct earlier revision. A reviewer comment may look incorrect simply because you are reading it against an older draft or the wrong set of changes.

If the version still shows **Pending Review**, check for these common causes:

- one or more **unresolved comment threads**
- a required reviewer who has not responded
- a follow-up round that was started but not completed

- a version update that needs another approval pass

If approval appears to have disappeared after you made changes, verify the current approval state on the version after the new submission or update. In some review workflows, a significant revision means the earlier approval no longer applies to the latest content. Instead of relying on the old outcome, ask reviewers to confirm the updated version directly.

When feedback is unclear, stay in the same comment thread and ask for a concrete change request. Good follow-up questions are specific to the content under review, such as whether the reviewer wants a factual correction, extra detail, or a wording change. This keeps the discussion tied to the exact section and helps avoid unnecessary edits.

[SCREENSHOT: Version review screen showing pending review status, unresolved comments, and latest revision comparison]

Overview

Version review follow-up in Atloria is about turning reviewer decisions into clear action on the **version details** screen. The main items to watch are the **review status badge**, the **comment thread**, any **highlighted feedback** attached to content, and the current **approval state** after edits. Those elements together tell you whether the version can move forward or needs more work.

Keep these core ideas in mind:

- **Status badges** such as **Approved**, **Changes Requested**, and **Rejected** show the overall outcome
- **Comment threads** explain the reason behind that outcome
- **Highlighted comments** and the **comparison view** help you locate the exact content under review
- **Unresolved comments** and **pending reviewers** usually mean follow-up is still open
- **Updated versions** may need a fresh approval check before release

This guide focuses on what to do after a review decision has already been made. If you need help with the earlier request-and-decision process, use Managing Version Review Requests and Decisions. If you need help interpreting statuses and comment behavior more generally, see Understanding Review Statuses Comments and Next Steps.

A practical review habit in Atloria is to read the full discussion before editing, make all related content changes together, and then leave a short response note explaining what changed. That keeps the version history easier to follow and makes re-review faster for approvers.

The goal is not just to clear comments. The goal is to make sure the current version, the visible review outcome, and the release decision all match.

Prerequisites

Before you work through review feedback in Atloria, make sure you already have access to the version that was submitted for review and that the version has an active review outcome or comment history on its **version details** page. You do not need special setup steps in this guide, but you do need to be in the right place in the workflow.

You should already be comfortable with these tasks:

- opening a project and selecting the correct **documentation version**
- viewing the **version details** screen
- reading **review statuses** and **approval outcomes**
- using the **comment thread** and any available **comparison** or **diff** view
- updating the version content or revised draft after feedback

This guide assumes you have already completed the earlier review stages covered in:

- Reviewing and Approving Documentation Versions
- Managing Version Review Decisions and Approvals
- Understanding Review Statuses Comments and Next Steps
- Preparing a Version for Final Release Review

It also helps if the version already includes:

- at least one reviewer decision, such as **Approved**, **Changes Requested**, or **Rejected**
- visible reviewer comments in the version discussion area
- a latest revision you can compare against earlier changes, if follow-up edits were made

If you open the version and do not see any review status, comments, or approval activity, you may still be earlier in the workflow. In that case, return to the review request process before using the follow-up steps in this guide.