



## Documentation Release Process

Version	Date	Author	Summary of Changes
1.0	18-3-24	Sachin S	Initial release of the documentation process

I. Handling Documentation Update Requests.....	1
Channels for Documentation Update Requests.....	1
Request Form Submission Guidelines.....	1
II. Task Management with Kanban/Trello.....	2
Workflow Overview.....	2
Transition between Release Candidates.....	3
Review and Approval.....	3
Criteria for Transition.....	5
Creating Release Branch.....	6
Handling Hotfixes.....	6
Finalizing Release Candidate.....	6
Document Diff List.....	6
IV. User Sharing Strategy.....	7
Information to Share with Customers.....	7
Documentation Sharing Formats.....	7
V. Finding Latest Version Documentation.....	7
Accessing the Latest Version.....	8

If Direct Project Link is Shared:.....	8
If Accessing ReadtheDocs Dashboard:.....	8

# I. Handling Documentation Update Requests

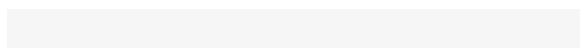
## Channels for Documentation Update Requests

All documentation requests should be through the following “Request Forms”:

- **New Documentation Creation Request:** for new features
- **Documentation Bug Fix Request:** for issues in the doc, such as broken links, incorrect information, or typos
- **Documentation Investigation Request:** for areas needing improvement, such as unclear explanations, incomplete sections, or user-reported confusion
- **Documentation Review Request:** for stakeholders to request a formal review of specific documentation sections
- **Documentation Localization Request:** for translation requests

## Request Form Submission Guidelines

Each Request Forms must follow the below template:



<Template to include the necessary information (e.g., description of the update, urgency level, affected sections)>

# II. Task Management with Kanban/Trello

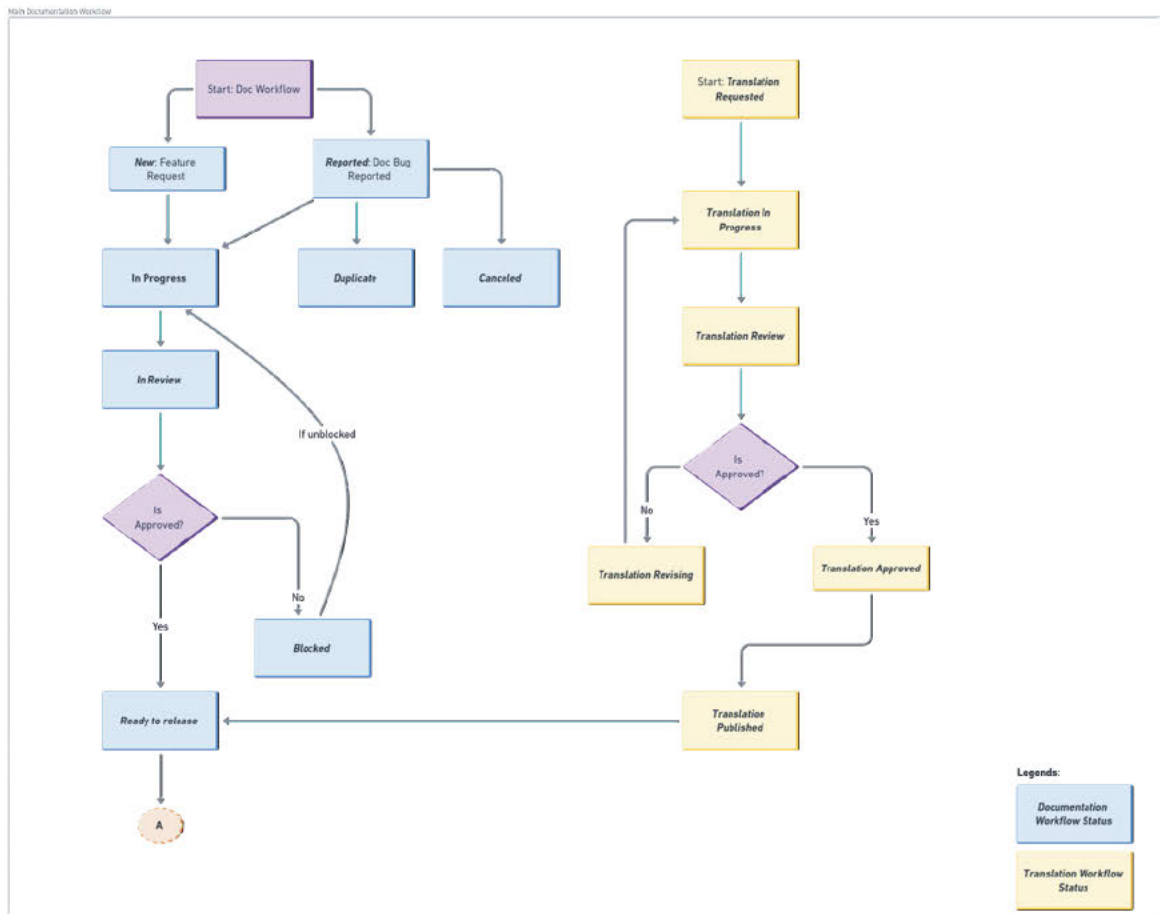
## Workflow Overview

The documentation team utilizes Kanban/Trello boards to efficiently manage documentation tasks throughout their lifecycle. The workflow is structured into several stages, including:

- **Reported** - Documentation bug has been identified and reported
- **Completed** - Documentation has been updated, and the bug is resolved/  
Finalized and published to ReadtheDocs
- **In Progress** - Initial drafting of technical specification in progress or updating the existing document based on technical and peer feedback
- **In Review** - Review by stakeholders
- **Blocked** - Awaiting information or decisions from engineering to proceed
- **Ready to release** - Ready to release a new RC. See [Release Process](#) for next steps.
- **Duplicate** - Duplicate documentation task
- **Canceled** – Canceled documentation task

The workflow for translation is as follows:

- **Translation Requested** - Translation required for the specification
- **Translation In Progress** - Translation underway
- **Translation Review** - Translated documents are under review for accuracy and localization
- **Translation Revising** - Incorporating feedback into the translated documents
- **Translation Approved** - Translated documentation has been approved
- **Translation Published** - Translated documents are published to ReadtheDocs
- **Translation Archived** - Older versions of translated documents are archived for reference



### III. ReadtheDocs Release Process

This section outlines the process for managing and transitioning documentation updates through various release candidates on ReadtheDocs.

#### Transition between Release Candidates

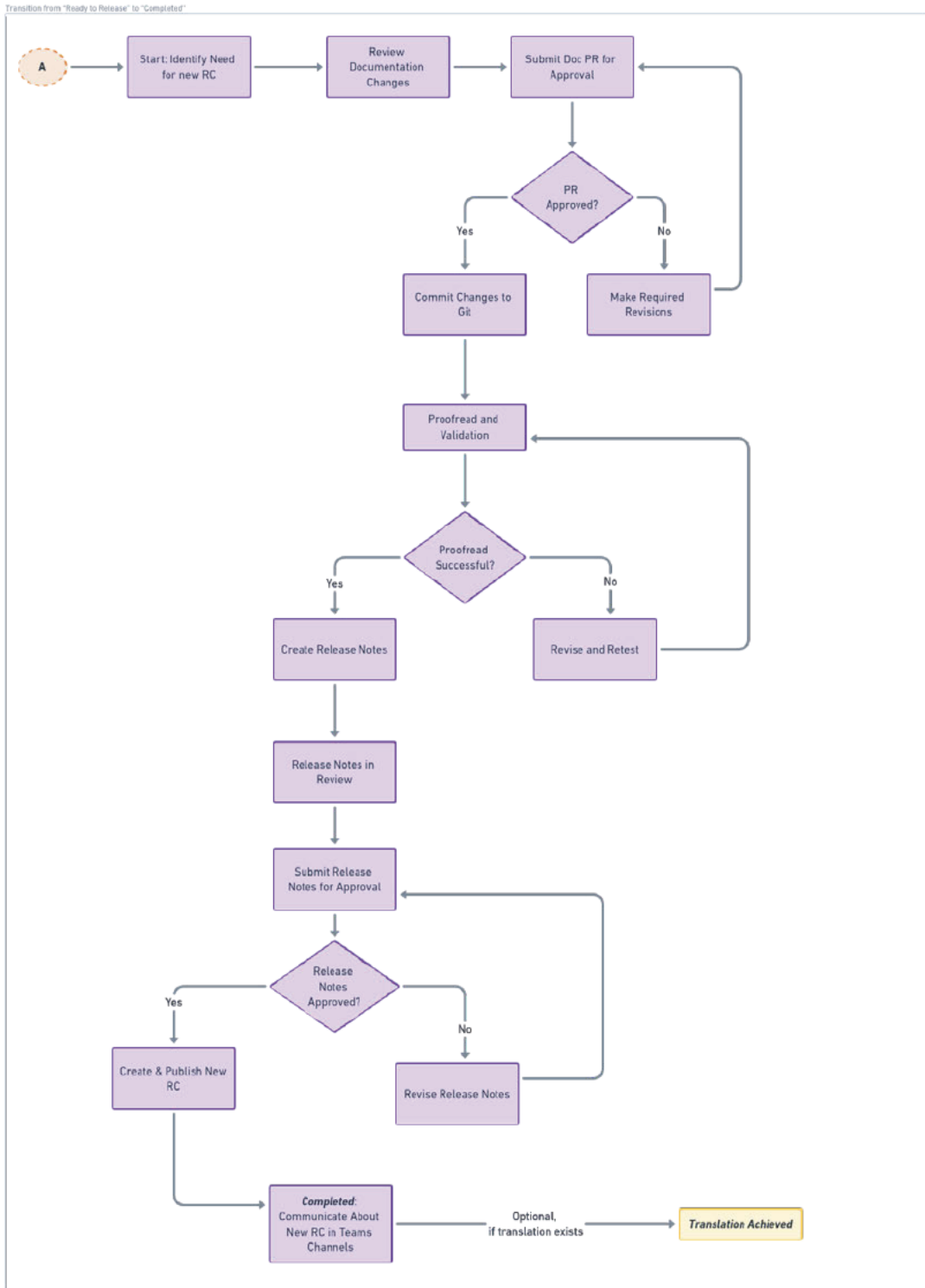
This subsection describes the overview of transitioning between release candidates, from initial review and approval to the finalization of a new RC.

#### Review and Approval

Before transitioning to a new release candidate (RC) on ReadtheDocs, the following conditions must be met without which a new RC would not be created.

- o All pending documentation PRs must be reviewed and approved by engineers and necessary stakeholders
- o Additionally, a well-defined Release Notes outlining the changes must be created, reviewed and approved. This collaborative effort ensures that the documentation accurately reflects the software's current state and user requirements.





## Criteria for Transition

The transition criteria for moving from one release candidate to the next should be clearly defined. This may include completing all planned documentation updates, resolving any outstanding issues or bugs, and obtaining necessary approvals from stakeholders.

- **Git Integration:** Ensure that all documentation changes are committed to the version control system (e.g., Git) before transitioning to a new release candidate. This helps track changes, maintain version history, and facilitate collaboration among team members.
- **Testing and Validation:** Before finalizing the transition, it's essential to conduct testing and validation to ensure that the documentation accurately reflects the latest software changes. This may involve verifying links, updating screenshots, and confirming accuracy of instructions.

## Updating Documentation

### Creating Release Branch

1. Start by creating a new release branch from the **main** branch using the following commands:  
Example: `git checkout -b r[ ] 0`
2. Tag the release branch to mark the initial release candidate:  
Example: `git tag 3.[ ]`
3. Push the tag to the remote repository

### Handling Hotfixes

- If bugs are identified during testing of the release candidate, create hotfix branches from the release branch:  
Example: `git checkout -b [ ] re-fix`
- After completing the hotfix, merge the changes to the release branch and cherry-pick the commits to the **main** branch and other active release branches.

### Finalizing Release Candidate

- After testing the hotfixes and ensuring no further bugs, create a new release candidate by re-tagging the release branch:

Example: `git tag 3. [redacted] 1`

- Push the new tag to the remote repository:

Example: `git push origin <new-tag-name>`

## Document Diff List

When reviewing changes between different versions or releases of documentation, there are two methods to access the document diff list:

### 1. Using ReadtheDocs:

Access the document diff list by using the version drop-down option on ReadtheDocs to compare differences between versions.

### 2. Using Release Notes:

Release notes provide a summarized overview of the changes made between different versions or releases of the documentation.

## IV. User Sharing Strategy

### Information to Share with Customers

Currently, the User Guide (incl functional specification) /API Guide is shared with customers.

### Documentation Sharing Formats

Both the User Guide and API Guide are shared with customers:

- In HTML format via ReadtheDocs links
- In PDF format for customers who prefer offline access or printing

### Communication Channels

Internal communication channels such as [redacted] (e.g., [redacted]) are utilized. Managers within these channels share updates and announcements with customers.

## V. Finding Latest Version Documentation

### Accessing the Latest Version

Users can easily [redacted] for a specific release through various methods provided by the ReadtheDocs platform. These methods include direct links, navigation menus, and search functionality.

### Navigation Instructions

To ensure users can navigate ReadtheDocs and locate the desired version of documentation with ease, follow these step-by-step instructions based on how the documentation link is accessed:

#### If Direct Project Link is Shared:

- **Access Direct Link:** Click on the direct link provided to you for the project documentation.
- **Select Suitable Version:** Once on the documentation page, locate the [redacted] on the bottom-right which list of available versions of the documentation.

#### If Accessing ReadtheDocs Dashboard:

1. **Access ReadtheDocs:** Open your web browser and navigate to the [redacted].
2. **Select Project:** Locate and select the project or documentation repository for the product or service you are interested in.
3. **Navigate to Versions:** Once on the project page, navigate to the [redacted] on the bottom-right which list of available versions of the documentation.