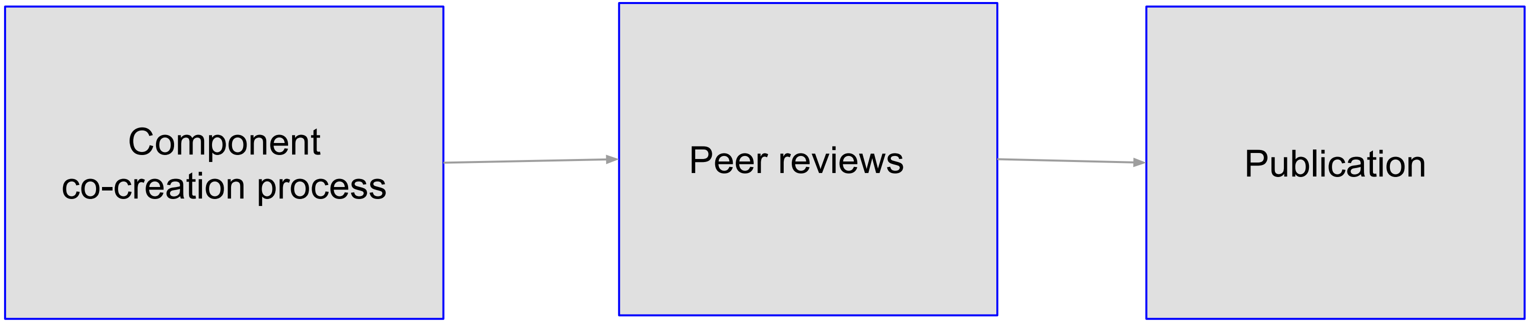
Improved Component Documentation Creation and Review process

•••

Charlene Jaszewski

The current component documentation process, simplified

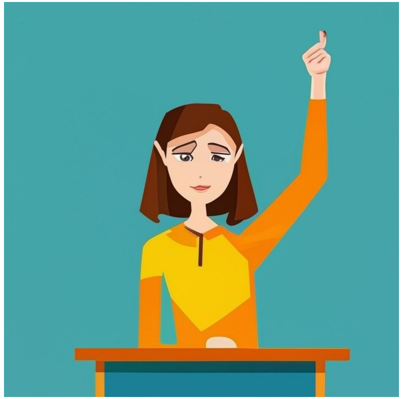


Component co-creation process

Peer reviews

Publication

Here’s the current peer review process:



Someone writes in the Teams *Content* channel:

“Hey folks, can someone review this for me?”

This isn’t a great process.

Most documentation/editing problems arise from the same issues:

- Lack of structure and expectations around documentation and reviews (what/who/why/when/how)
- Review criteria hasn’t been defined
- Poorly defined roles and responsibilities

The U.S. Bank peer review process has all these issues.

My background as a content consultant

- I’ve created documentation and editing processes for companies that didn’t have them
- I’ve captured and solidified standards and processes at places where they were lived in people’s heads and/or were mushy

These issues happen everywhere

- These issues happen in many companies, whether they’re startups or established companies.
- Either the process doesn’t exist/isn’t fleshed out (in the case of startups), or, in the interest of just “getting things done,” thought isn’t given to *process*.

What’s the solution?

- Clarify roles and responsibilities
- Define review layers
- Spell out review criteria for each layer/role

Roles and Responsibilities

Roles and responsibilities issues

- A11y's role not clearly defined
- Quad Review's purpose (and criteria) not clear
- Categorization of non-component things (who should do it)

Issue: A11Y’s role in reviews is not clearly defined

It’s not clear where A11Y reviews should occur in the process, and the scope of their reviews.

Sometimes A11Y contributes content during co-creation, and other times they have been heavy reviewers of non-A11Y content in peer reviews.

Solution: Work with A11Y team members to set expectations

We need to define:

- The scope of A11y's involvement
- The best time in the documentation creation/review process for A11Y to contribute

Issue: Quad Review purpose and timing aren’t clear

- Quad Review doesn’t have a set place in peer review process
- Unclear whether Quad Review purpose is to approve content before completion, or to simply sign off on completed content

Solution: Define Quad Review purpose, timing & criteria

- Quad Review is for final signoff on completed content
- Quad reviews should occur after peer review process has been completed, and after content has been placed into ZeroHeight
- CS will take screenshots of component documentation and place it in Figma, where Quad can leave comments (thanks to Courtney for this great solution!)

Issue: Categorization of non-component things

This is something that came up recently (in regards to the datagrid “component” and the table component). Documentation for these [things] is coming to peer review, and we’re discovering that the [thing] is not a component, and then we have some of the following issues:

- How do we document it? Do we shove it into the current component template, when it seems to require a slightly different template?
- Who should decide what it is and how it should be documented? (Product owner? Content strategists? Platform teams?)
- We don’t have a full scope of non-component things in the Shield universe.

Next Steps: Categorization of non-component things

- Determine WHO decides this (Product Owner vs. whole CS team vs. platform teams)
- Ensure categorization occurs before peer review, during co-creation
- Try to speak to all platforms to suss out how many non-component things exist
- Determine if we need to expand current template to accommodate them, or if new templates should be developed

Structure and Expectations

Structure and expectations issues

- No time limit to complete review
- No one is assigned to review
- Peer review time not accounted for in sprint planning
- We haven’t defined HOW to review
- Zeroheight template adherence not part of review
- Documentation review criteria not defined

Issue: No set time limit to complete review

Requestor simply asks for the review, and doesn’t state time limit.

Solution: Communicate a time limit

This issue has a simple fix: when requesting a peer review, specify a time limit for comments.

We could define a set time limit (e.g., a week), but it might be better to leave this flexible. The important thing it to communicate one.

"I need comments back by Wednesday 5p.m. CT. Thanks!"

Issue: No one is specifically assigned to review

When a request for review is tossed into Teams, no one is assigned to complete it, so it's basically assigned to everyone and no one at the same time. This causes the following problems:

- It might not be reviewed in a timely manner
- There might be a "too many cooks in the kitchen" situation, generating a flood of comments, sometimes about the same issues.

Solution: Assign review responsibility

Proposed solutions:

- Rotating responsibility for content reviews (TBD rotation, but by sprint might make sense).
- Multiple people can review (TBD how many), but they'll review in sequence, so there (hopefully) won't be multiple comments on the same issues.
- We may want to assign reviewers per speciality (line editing for content clarity, copy editing, etc)
- JIRA stories to account for regular content peer reviews.

Next steps:

- Decide on rotation period and assign
- Decide how many people will review
- Ensure JIRA stories account for peer review time, for all platforms (they have been assigned for React)

Issue: Peer review time not accounted for in sprint planning

Currently peer content review time is not accounted for in sprint planning, so reviews end up being “extra” work and may affect other work happening.

Solution: Add JIRA story each sprint for peer review time

Each CS will have a JIRA story for "peer review" for each sprint.

We'll start with two hours per sprint and adjust upwards if necessary.

Issue: Zeroheight template compliance isn’t part of the current review process

Zeroheight is the new home for component documentation, and content needs to be reviewed for adherence to the (new) Zeroheight template.

Solution: Zeroheight pre-flight check will be another “layer” of review

As part of defining “layers” of content review, doing a Zeroheight “pre-flight” check will be a separate process from the content peer review process.

- There will be a ZeroHeight template “checklist” CS can review to ensure component documentation adheres to template.
- A Foundations CS will review component documentation after it’s been added to ZeroHeight to ensure compliance.

Issue: No defined documentation quality/review dimensions

When talking about documentation (or other content), there are a few dimensions of quality and or/reviews:

- Completeness
- Accuracy
- Consistency
- Content clarity
- Copy edit

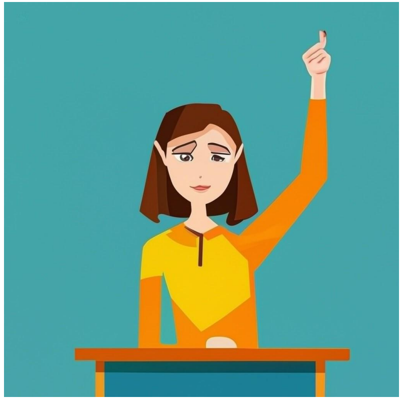
There are a few more specific to Shield:

- Blueprint check (that something is a Blueprint element vs. a customization option)
- Zeroheight template adherence
- Final proofread in ZH to check for copy/paste errors

Note: These don’t need to be separate reviews. I’m just breaking out the dimensions of a review

Review Structure & Criteria

Here's the current peer review process:



Someone writes in the Teams *Content* channel:

"Hey folks, can someone review this for me?"

We need to define review structure & criteria.

Documentation review “layers”

Here are the following review “layers” that documentation should go through.

- Completeness (Have all properties been documented?)
- Accuracy (Is the documentation correct?)
- Consistency (of information both inside document and between documents)
- Content clarity (Is it written so the intended audience can easily understand it?)
- Copy edit

The Shield documentation review needs several additional layers:

- Adheres to *U.S. Bank Brand Guide* and new *REACT Writing Guidelines*
- Zeroheight template adherence
- Final proofread after placement in CMS

Depending on the company/process, one person might do these, or they might be split between multiple roles.

Issue: With no specific review structure or criteria, content strategists are doing more work than they should be

Here’s what’s happening now:

Co-creation
Write documentation

Who: CS, Dev, Design

Peer Review
Completeness
Accuracy
Consistency
Content clarity
Copy edit
Adheres to US Bank Brand Guide

Who: CS

Solution: Accuracy/completeness checks should be done during co-creation process

By the time documentation gets down to peer review, we should be confident that information is accurate and complete.

For this to happen, we need:

- Leadership support for sprint time to ensure this happens.
- Clearly defined review criteria that details everything that should be present and accurate in documentation, and questions that should be answered. This should help both the writer, and the accuracy/completeness reviewer.

Solution: Split review tasks between layers and roles

Content should be accurate and complete before being sent to peer review.

Co-creation

- Write documentation
- A11y review
- Accuracy review
- Completeness review

Who: Design, Dev, A11y CS

Peer Review

- Consistency
- Content clarity
- Copy edit
- Adheres to brand guide
- *Adheres to *REACT Writing Guidelines*

Who: All CS

Zeroheight pre-flight checklist (after content placed in ZH)

- *Adheres to ZH template
- *Final proofread in ZH

Who: Foundations CS

* New review tasks for Shield 2.0

I.LOVE.CHECKLISTS

Review checklists will be a great way to:

- Set expectations around what needs to be done (and the level of quality)
- Get everyone on the same page
- Help new team members get up to speed faster

Completeness checklist examples

- Have all customization options been documented?
    - Is there a default setting or value?
    - Are there min/max settings? (and if there are, is there guidance around their use?)
    - Does this setting conflict with any others?
    - Can this option only be used in certain situations?
    - Is there a difference between what property says and what guidance is? Provide that rationale. (e.g., property says "set minimum of 1" but devs told you a minimum of 3 is better.)
- Have all behaviors been documented?
- Have states (if any) been fully described (and images provided if UI changes)?
- Has A11Y content been captured?
- Has usage guidance been captured?
- Has implementation guidance been captured?

Completeness, cont.

- Is this similar to any other components? Describe any differentiators. Why would you use one over the other?
- Should this be used instead of any other components? Describe the situations, types of content, etc. that will make it clear which one you'd use. What's the key differentiator?
- Can this be used with any other components? In which situations, or in which types of content?
- Are any other components part of this (e.g., text input as part of editable cell in table)
- Is this component best used in a certain type of content? In a certain situation? In forms?

Accuracy checklist examples

- Are the explanations for customization options and behaviors correct? Is functionality described accurately?
- Is guidance described accurately?
- Have any Blueprint terms been characterized as customization options?
- etc.

Note: There's some overlap between this and content clarity

Consistency checklist examples

- Are the customization options named the same as the component properties?
- Does the option documentation say the same thing as the properties in the readme and/or functional specs?
- Is terminology and/or language describing functionality used consistently throughout the document?
- Does this document use the same terminology and/or language as other component documentation? (The new *REACT Writing Guidelines* I'm authoring will help with this)

Content clarity/efficiency basics

- Content follows COP guidelines (use plain language, active voice, etc.)
- Terminology/concepts are clearly explained
- Sentences are concise and clear (short phrases used to capture meaning vs. a verbose few sentences)
- Repetitive terms or phrases eliminated
- Information duplication/redundancies eliminated (e.g., two sentences in same paragraph say pretty much the same thing, or there's one description in customization options but a slightly different one present in screen reader information)
- Writing has good rhythm (i.e., not too many lists in one sentence, not too many run-on sentences, etc.)

Content clarity/efficiency basics (cont.)

- Content answers user questions before they’re asked. For example, if content says, “You must set at least one column as *essential*,” you should define what “essential” means.
- Efficient words are used. Don’t say “tabular arrangement of data” when you could say “table.”
- Don’t use a sentence if you can say it in a word. If you’re listing variations and one is the default, you don’t need to spend a sentence after listing the options saying, “Option A is the default.” You can say, “The options are A (default), B, C and D.”
- More TBD

Anticipate questions from reviewers

Leave a comment for reviewers if:

- You’re missing some information
- Content is a little vague (due to lack of info)
- You wrestled with a sentence/paragraph and know it’s not great (say so!)
- You suspect people might not understand why it was documented a certain way.

Copy edit/proofread checklist examples

- Fix typos/misspellings
- FIx incorrect grammar (e.g. “These things is incorrect”)
- Fix errant capitalization (e.g., all headings must be sentence case)
- Ensure proper capitalization in bulleted lists
- Ensure parallel structure in bulleted lists
- Fix incorrect punctuation (e.g., a colon where a semicolon should be)
- Ensure the document has proper formatting
    - Remove extra spaces between paragraphs
    - Remove extra spaces after periods (should only be one)
    - Ensure Em and En dashes are correctly used (AP wants spaces around them)
- Ensure the document conforms to Content COP guidelines and AP style

Final proofread after content placed into Zeroheight

At this stage errors will likely be copy/paste errors or lack of formatting:

- Missing spaces
- Missing punctuation
- Extra lines between sections
- Content not formatted correctly
- Content duplicated

Issue: We haven’t defined *how* to edit/review

So now we’ve defined review criteria for the various layers, but there aren’t clearly defined guidelines on *how* to peer review/edit component documentation. For example:

- Can we directly edit the document to fix a typo?
- How about directly performing a line edit on a sentence to make it clearer?
- Can we move a sentence or paragraph from one location to another?
- If you directly edit, should you leave a comment as to WHY you made the change?

Solution: Define “how to review” guidelines

- Content strategists will meet to agree on the guidelines on how to review, and answer the previous questions.
- Guidelines that I have used with other clients are shown in next two slides.

“How to review” guidelines

You can directly edit the content if it’s to fix any of the following issues (mostly proofreading issues):

- Typos
- Incorrect punctuation
- Extra spaces (or missing spaces)
- Grammatical errors
- Errant capitalization

Leave comments/suggestions for everything else.

“How to review” guidelines (cont.)

Leave suggestions for the following issues:

- If you feel a sentence is confusing/not clear, highlight the sentence, and leave a comment with:
  - 1) the reason you don’t think it’s clear, and
  - 2) a suggestion for a clearer sentence (if possible).
  - 3) Questions to get more information if you can’t suggest something.
- If you think a sentence or paragraph should be moved elsewhere:
  - Highlight sentence and leave comment like, “this sentence belongs on page 10, in [blah blah blah] section. Then in that section, add placeholder like [add sentence from page 32].

Upstream issue: Functional specs aren’t comprehensive enough

Once we started talking about how accuracy checks need to happen before peer review, I found out that functional specs for components exist, but apparently aren’t comprehensive enough to provide a solid base for documentation.

Solution: Improve the functional specs

- Review current functional specs and provide recommendations for improvement.
- Get leadership to allow dev sprint time to write better functional specs.
- Ply the involved people with baked goods

Next Steps:
Effort & Impact

Next steps: Effort and Impact

Task	Effort	Impact	Who
Define accuracy checks to be done before peer review	High	High	Content strategists
Get Dev and Design to agree to accuracy checks before peer reviews	High	High	CS, Dev, Design
Review functional specs, define desired improvements	High	High	CS, Dev
Get Dev to agree to create better FS	High	High	CS, Dev
Define peer review criteria, get CS agreements	Medium	Medium	Content strategists
Agree to request time limits for reviews	Low	Medium	Content strategists
Determine number of reviewers and reviewer rotation	Low	Medium	Content strategists
Define HOW we will edit	Low	Medium	Content strategists
Determine who owns categorization of non-component things	Low	Medium	CS, PO
Define ZH adherence checklist	Low	Low	Content strategists
Move quad reviews until after content is in ZH	Low	Low	Content strategists
Move ZH adherence check to after content has moved to ZH	Low	Low	Content strategists

Thank You!