Content workflow workshop

HealthCare.gov

Context

While working on the team that supports HealthCare.gov's online healthcare application (internally referred to as App3), I found that our process for creating, reviewing, approving, and updating content was causing pain points for various members of our team at different stages of our product development workflow.

We lacked certainty of what was the "right" content that should actually be on the website.

Problems to solve

- Updating content in multiple places - Confluence, Phrase (localization platform), Invision. Inefficient and a big burden, and confusing to figure out what the source of truth is.

- Unclear how to use the Confluence content deck as a source of truth when iterating on content.

- Our current system broke down with a recent content audit – hundreds of changes made in PDFs then directly to Phrase, missed step in Confluence to record the changes.

- Don't have a clear, mutually understood process for how QA should test content changes - what should they reference?

Goals

- Understand each team's needs and current process

- Streamline this process, eliminate redundancies and gaps

- Ensure everyone has a shared understanding of the process and responsibilities

- Ensure we have adequate documentation – both of the content, and the process around it

Discovery process

I facilitated a series of workshops with members of our team (designers, product managers, engineers), as well as our government partners (English & Spanish language content teams, product owner, testing partners, policy SMEs) in order to:
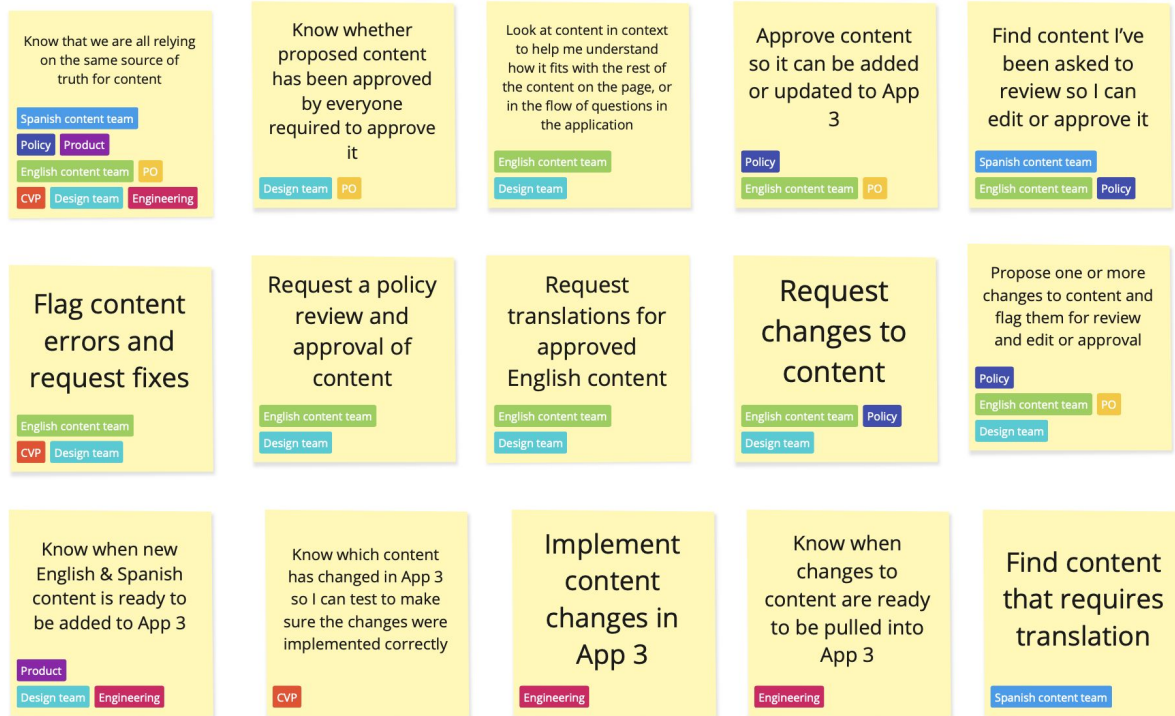
1. **User needs -** Collect different parties' needs around content in App3

2. **Journey mapping -** Identify pain points in our workflow that prevent meeting those needs

3. **Solutions -** Develop recommendations for process changes and tooling to resolve pain points

User needs

"As [role] I need [x] so that [outcome]"

I asked our team members to write user stories for their needs around content in App3.

I then synthesized common themes to show where different groups' needs overlapped.

Know that we are all relying on the same source of truth for content

Spanish content team
Policy   Product
English content team   PO
CVP   Design team   Engineering

Know whether proposed content has been approved by everyone required to approve it

Design team   PO

Look at content in context to help me understand how it fits with the rest of the content on the page, or in the flow of questions in the application

English content team
Design team

Approve content so it can be added or updated to App 3

Policy
English content team   PO

Find content I've been asked to review so I can edit or approve it

Spanish content team
English content team   Policy

Flag content errors and request fixes

English content team
CVP   Design team

Request a policy review and approval of content

English content team
Design team

Request translations for approved English content

English content team
Design team

Request changes to content

English content team   Policy
Design team

Propose one or more changes to content and flag them for review and edit or approval

Policy
English content team   PO
Design team

Know when new English & Spanish content is ready to be added to App 3

Product
Design team   Engineering

Know which content has changed in App 3 so I can test to make sure the changes were implemented correctly

CVP

Implement content changes in App 3

Engineering

Know when changes to content are ready to be pulled into App 3

Engineering

Find content that requires translation
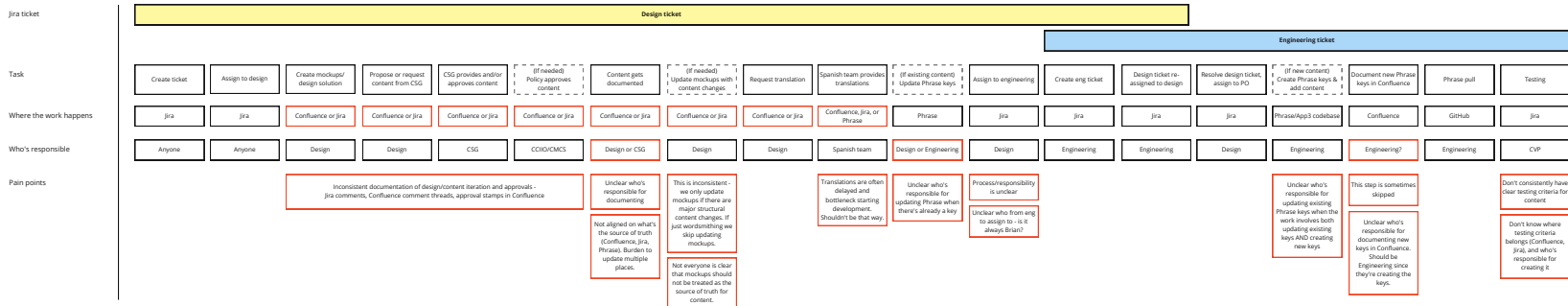
Spanish content team

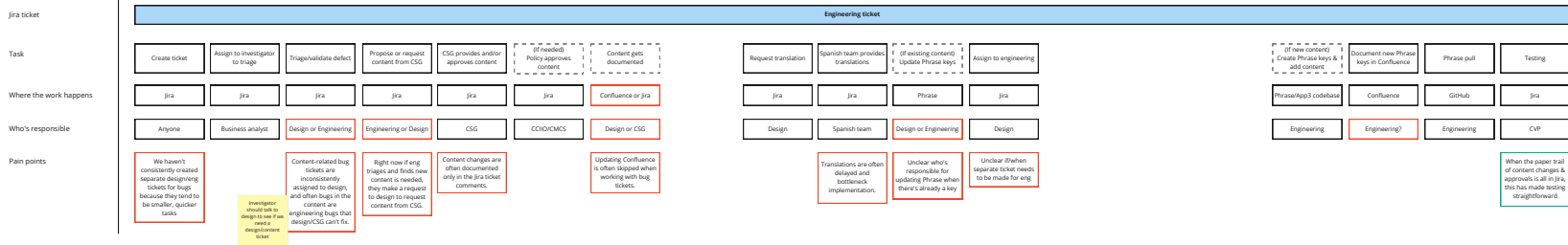# Journey mapping content workflows

I interviewed multiple team members in order to develop a journey map of sorts of our content workflow – both for content that's produced as a result of new feature development, as well as content that's produced as a result of a reported bug.

Workflow phases

| Identifying & assigning the work | Content drafts & edits / Review / Approval | Documenting changes | Translations | Documenting changes | Implementing changes in App 3 | Documenting changes | Testing updates | Bugs |

Current content workflow - Stories

Jira ticket: Design ticket / Engineering ticket

Task	Where the work happens	Who's responsible	Pain points
Create ticket	Jira	Anyone	
Assign to design	Jira	Anyone	
Create mockups/ design solution	Confluence or Jira	Design	Inconsistent documentation of design/content iteration and approvals - Jira comments, Confluence comment threads, approval stamps in Confluence
Propose or request content from CSG	Confluence or Jira	Design	
CSG provides and/or approves content	Confluence or Jira	CSG	
(If needed) Policy approves content	Confluence or Jira	CCIIO/CMCS	
Content gets documented	Confluence or Jira	Design or CSG	Unclear who's responsible for documenting / Not aligned on what's the source of truth (Confluence, Jira, Phrase). Burden to update multiple places.
(If needed) Update mockups with content changes	Confluence or Jira	Design	This is inconsistent - we only update mockups if there are major structural content changes. If just wordsmithing we skip updating mockups. / Not everyone is clear that mockups should not be treated as the source of truth for content.
Request translation	Confluence or Jira	Design	
Spanish team provides translations	Confluence, Jira, or Phrase	Spanish team	Translations are often delayed and bottleneck starting development.
(If existing content) Update Phrase keys	Phrase	Design or Engineering	Unclear who's responsible for updating Phrase when there's already a key
Assign to engineering	Jira	Design	Process/responsibility is unclear / Unclear who from eng to assign to - is it always Brian?
Create eng ticket	Jira	Engineering	
Design ticket re-assigned to design	Jira	Engineering	
Resolve design ticket, assign to PO	Jira	Design	
(If new content) Create Phrase keys & add content	Phrase/App3 codebase	Engineering	Unclear who's responsible for updating existing Phrase keys when the work involves both updating existing keys AND creating new keys
Document new Phrase keys in Confluence	Confluence	Engineering?	This step is sometimes skipped / Unclear who's responsible for documenting new keys in Confluence. Should be Engineering since they're creating the keys.
Phrase pull	GitHub	Engineering	
Testing	Jira	CVP	Don't consistently have clear testing criteria for content / Don't know where testing criteria belongs (Confluence, Jira), and who's responsible creating it

Current content workflow - Bugs

Jira ticket: Engineering ticket

Task	Where the work happens	Who's responsible	Pain points
Create ticket	Jira	Anyone	We haven't consistently created separate design/eng tickets for bugs because they tend to be smaller, quicker tasks
Assign to investigator to triage	Jira	Business analyst	Investigator should talk to design to see if we need a design/content ticket
Triage/validate defect	Jira	Design or Engineering	Content-related bug tickets are inconsistently assigned to design, and often bugs in the content are engineering bugs that design/CSG can't fix.
Propose or request content from CSG	Jira	Engineering or Design	Right now if eng triages and finds new content is needed, they make a request to design to request content from CSG.
CSG provides and/or approves content	Jira	CSG	Content changes are often documented only in the Jira ticket comments.
(If needed) Policy approves content	Jira	CCIIO/CMCS	
Content gets documented	Confluence or Jira	Design or CSG	Updating Confluence is often skipped when working with bug tickets.
Request translation	Jira	Design	Translations are often delayed and bottleneck implementation.
Spanish team provides translations	Jira	Spanish team	
(If existing content) Update Phrase keys	Phrase	Design or Engineering	Unclear who's responsible for updating Phrase when there's already a key
Assign to engineering	Jira	Design	Unclear if/when separate ticket needs to be made for eng
(If new content) Create Phrase keys & add content	Phrase/App3 codebase	Engineering	
Document new Phrase keys in Confluence	Confluence	Engineering?	
Phrase pull	GitHub	Engineering	
Testing	Jira	CVP	When the paper trail of content changes & approvals is all in Jira, this has made testing straightforward

Recommended solutions

After reviewing the user needs and journey maps with our team, I developed a set of recommendations focused on using existing tooling in our localization software Phrase, rather than building custom processes or tools ourselves.

	Need	Process ideas	Phrase features we could use
Approvals	PO needs to know that the content that's going to prod is approved by WNMG, CSG, and policy	Keep a rolling changelog of updates w/ WNMG and policy approval checkpoints as needed in Confluence. Use Phrase history and/or proofreading/jobs workflows to show that each content change has been been done by or verified by CSG.	**Verification & proofreading** Each time a key is added or updated, it flips to "unverified" & can be manually verified (in this case, by CSG)
Documenting requirements	PO/stakeholders need to know what the content in prod SHOULD be	If we have a solid approval and testing process in place, what **should** be in prod doesn't need to be documented separately from what **is** in prod. Our process should make it a given that what's in prod is what should be there.	
Meeting requirements	PO/stakeholders need to know what the content in prod ACTUALLY is	Need to investigate if this is actually a significant need; if it is, the In-Context Editor would be the best way to display the source of truth for content in a friendly way.	**In-Context Editor** Shows a preview of all App3 fields on one long page, with the ability to directly edit content on the page. We have this, but it needs developer time to get it back to parity with all of this past year's cutover features.
	CVP needs to know what the content for a feature SHOULD be to validate it's been implemented correctly	If we do the approval process right, CVP should only need to validate that the content changes made in Phrase are showing correctly in the test environment.	**To investigate:** Can we get a printout of the changes tied to a job (see below)? Does that provide enough context to understand what parts of the application changed?
Reducing overhead	Ad Hoc needs content team to own content changes	Get CSG comfortable using Phrase as much as possible. They should also manage the translation workflow internally instead of Ad Hoc requesting translations.	**In-Context Editor** Gives CSG a friendlier environment for editing content. **Jobs** Assign specific people a specific set of keys to work on, tied to a Jira ticket and with a due date.
	Ad Hoc needs less hands involved in pushing a content change through	Investigate ways to streamline Phrase <> Github processes. Can we automate the push from Phrase to Github? For new content, can we create new keys directly in Phrase?	**Github sync** Gives Phrase controls to pull updates from Github or create a pull request with recent updates. **Webhooks** Set up to notify of different updates in Phrase, e.g. locale updates, job completion, etc.
	Ad Hoc needs better control over how Phrase updates are deployed	Investigate ways to streamline Phrase <> Github processes. Can we automate the push from Phrase to Github? For new content, can we create new keys directly in Phrase?	**Branches** If a certain body of work is in progress and should be deployed all together, work on it in a separate branch and merge into master when ready to deploy it to a test environment. Can also configure the In-Context Editor to display content from a specific branch.

Thank you!