

Git Hub - Collab Process

[Step-by-step guide](#)

Some details regarding github:

* Can use either a desktop client or the web client to access the repository

* Can have either a public repository (anyone can see it, but the owner controls what commits are accepted; free), or a private repository (only the owner and invited collaborators can see and commit to it; paid subscription required) —

<https://guides.github.com/introduction/getting-your-project-on-github/>

For actual document editing:

Step one: A repository is created.

Step two: The owner of that repository adds the original, master copy of the document (i.e., "mvcr-spec).

Step three: Individuals who want to contribute (change, add, remove)

text from the master copy creates a "fork" of the repository and after editing, creates a pull request representing the proposed changes. This gives them their own space within which to make changes. They might call their copy "mvcr-spec-06"

Step four: The changes made to mvcr-spec-06 are "committed" back to github, where others may review the new file, add comments or issues, or even create a branch from that to offer their own text.

Step five: After the WG comes to consensus on a set of changes, the forked copy will be "merged" back to the master file, and all changes will be incorporated. We will have a history of what changed, and can run a diff to highlight those changes at any point in time in the future.

Step six: After a new master file is created, based on merging the pull requests back into the master, the WG chair or the WG editor should create a copy on the Kantara wiki.

For issue tracking:

* This might be used to help keep track of suggestions that come out of the call, and could serve as a way to make sure we have properly and publicly responded to the issues and comments received, and indicate in which version of the spec those comments were addressed

* We can decide upon a set of labels (e.g., diagram, use-case,

technical, editorial) to help structure the list of issues.

* Note that anyone can create and view issues on free public repositories; only teammates and collaborators can create and view issues on private repositories



Related articles

Error rendering macro 'contentbylabel'
parameters should not be empty