Overview

The Hyrax Development Support & Engagement Working Group is creating this document to describe known challenges that may limit contributions to Samvera community code.

Challenges to participating in community work

One-off contributions

Definition: Creating a PR that addresses a single issue outside the scope of a larger effort.

Challenges:

- TIME
    - contributing to Samvera code base can take twice as long as putting it in local institution code base
        - review can be significantly less locally
        - there may be a higher tolerance for less testing locally
        - matching patterns and style requirements may be more restrictive for Samvera code base
        - limited set of PR reviewers in community so there can be wait time
        - process of responding to review requests, making changes, committing, and getting another review takes significant time
    - CLA for individual or organization. Can take significant time get this in place especially if the organization CLA isn't in place.
- ACCEPTANCE
    - determining if the functionality you want to contribute fits with the vision for the code base by the Tech Lead and Product Owner can be hard to figure out
    - lack of plug-in architecture prevents some contributions that may be helpful for others but that do not rise to the level of direct inclusion in the community code base. (e.g. subscription email notifications when new works are added to Hyrax)
    - no designated reviewer to take the first look at the issue. Can get lost with no one following up on the review.
- SKILL
    - new contributors may not be aware of community norms (e.g. requirement of full testing for all PRs, a preference for smaller incremental PRs over larger PRs making multiple changes, etc.)
    - new contributors may not understand the patterns being used in the code base
    - new contributors may not have the skills and/or confidence to create a PR for the community code base
- FRUSTRATION

    - extra time to successfully get a PR merged
    - unable to get sufficient feedback and support during the development process
    - effort was exerted, but the PR never merged (e.g. PR was never reviewed, contributor didn't have time to address reviewer requests, solution was not accepted, functionality was not compatible with the current roadmap, etc.)

Working group contributions - targeted functionality

Definition: Participating in a working group that is focused on bringing specific functionality into the code base (e.g. Collection Extensions, Analytics, Permissions)

Challenges:

- TIME
    - time commitment may be significant (e.g. 6 months) depending on the functionality being implemented
    - time commitment may be close together (e.g. 6 consecutive weeks) leaving less time for local concerns during the development period
- ACCEPTANCE
    - not enough institutions want the functionality limiting the pool of potential developers
    - many institutions want the functionality but are unwilling (e.g. pressing local concerns) or unable (e.g. small development staff) to provide developers
- SKILL
    - too many of the developers are unfamiliar with the code base and, as such, require support from other developers in the group *(NOTE: This is a challenge of balance. Bringing new developers on board is a core value of the community.)*
    - new contributors may not ask for help when they get stuck
    - lose knowledge about the targeted functionality as developers rotate off the project
    - new developers rotating on in the middle of the working group require extra time to familiarize themselves to the targeted functionality and its current state of development
- FRUSTRATION
    - unclear problem descriptions
    - not enough support can lead to feeling lost
    - unable to contribute an acceptable solution can lead to feeling like time was wasted

Working group contributions - maintenance

Definition: Participating in a working group that is focused on addressing ongoing maintenance of the code base (e.g. Core Component Working Group, Hyrax-Maintenance Working Group)

Challenges:

- TIME
    - time commitment may be significant (e.g. 2 weeks on/off for 6 months) *(NOTE: Ideally, a developer commits to the entire time period.)*
- ACCEPTANCE
    - *NOTE: Acceptance is not a challenge as the functionality being worked on is predetermined maintenance work (e.g. technical debt, bug fixes, infrastructure, etc.)*
- SKILL
    - If too many of the developers are unfamiliar with the code base, extra support is required from other developers in the group *(NOTE: This is a challenge of balance. Bringing new developers on board is a core value of the community.)*
    - new contributors may not ask for help when they get stuck
- FRUSTRATION
    - unclear problem descriptions
    - not enough support can lead to feeling lost
    - unable to contribute an acceptable solution can lead to feeling like time was wasted

Other challenges for contribution

- ORGANIZATION SIZE
    - smaller institutions have few developers (sometimes only 1 or 2) making contributions a much higher percentage of total staff time
- DEMANDS OF LOCAL PROJECTS
    - developers continue to work on local institution priorities between sprints and can get called to solve a local problem during sprint times
    - institutional projects and priorities do not often align with community projects
- DEMANDS OF GRANT PROJECTS
    - Can pull priorities in unexpected directions in community
- LANGUAGE
    - code base has some obstacles for supporting languages other than English
    - asking questions on Slack or emails can be challenging if English is minimal or non-existent
    - lack of documentation in languages other than English
- BURNOUT
    - the same set of developers tend to contribute and this can lead to burnout over time
    - lack of engagement coupled with high expectations on the part of stakeholders is a major factor

- INSTITUTIONAL SUPPORT
    - Developers are directly or indirectly told that this kind of work is not as important as other work.
- TRAINING
    - Lack of consistent opportunities to formally onboard new contributors to the community

Challenges to leading community work

- TIME during sprints beyond writing code
    - answer questions
    - help developers find issues to work on (e.g. based on expertise, skill, priorities of what work makes sense next)
    - organize stand-ups and other meetings
- TIME outside sprints
    - create the vision and architecture documents
    - convey the vision to the community
    - create work statements in GitHub issues
    - call for participation (e.g. through emails, Slack)
    - report on work (e.g. emails, Slack, Connect)
- SKILLS
    - varied levels of skills for
        - communication
        - project management
        - organization
        - general technical expertise
        - Samvera code base expertise
- FRUSTRATION
    - having to solicit development resources
        - general emails calls for participation have variable levels of success in getting commitments
        - individual invites are generally more successful, but may lead to the same developers being invited
        - lack of information on potential developers to contact
        - having to solicit resources when you've already taken on all the technical coordination work
    - a ton of work for one person to address all the skills and time requirements
    - Difficulty consistently engaging community group volunteers to attend meetings, communicate, engage in the work