
Designing for Emergent Workflow Cultures: eLife, PRC, and Kotahi

*An overview of how Kotahi is
used for PRC.*

Designing for Emergent Workflow Cultures: eLife, PRC, and Kotahi

Scholarly publishing is evolving, and there is a need to understand and design the new (emergent) workflows while also designing technology to capture and support these processes. This article documents an ongoing collaboration to develop technology to meet emergent workflows in scholarly publishing, namely Publish-Review-Curate (PRC). We explore this topic with different eLife (<https://elifesciences.org/>) PRC community stakeholders using Kotahi (<https://kotahi.community>), a flexible open-source scholarly publishing platform that can support variant workflows (built by Coko).

What is PRC?

A preprint is a manuscript that has been shared publicly before submission to a scholarly journal. Preprints have been a popular pre-publication process for many years, however recently there has been an explosion of this kind of early sharing of research due to COVID-19.

There is a need to sort 'the good' preprints from 'the bad' so researchers can take advantage of early research without being mired in a torrent of questionable results. Consequently, volunteer communities of researchers have emerged to review preprints. This activity has been particularly popular during the COVID-19 era.

These communities publish their reviews of preprints to assist other researchers in finding and interpreting the research.

To illustrate this approach simply, we can represent preprint review processes with the initials PR (Publish, Review):

1. **Publish** - researchers publish preprints on 'preprint servers'
2. **Review** - researchers review the preprints (and share the reviews)

Traditional journal publishing, on the other hand, could be described with the initials RP (Review, Publish):

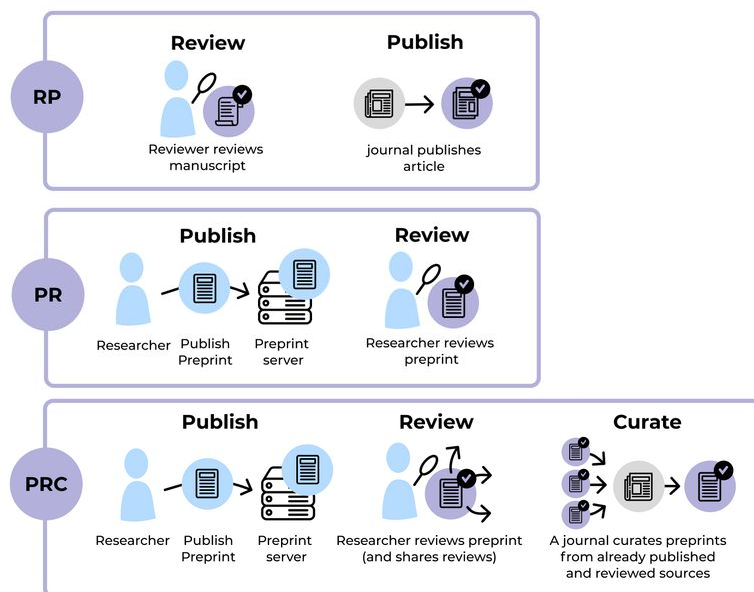
1. **Review** - researchers review a submitted manuscript
2. **Publish** - the manuscript is published

This simple flip of the PR vs RP highlights the fundamental difference in approach. Preprint review processes flip the typical path of a manuscript through

a journal on its head.

PRC adds a new component to this dynamic: Curation.

1. **Publish** - researchers publish preprints on 'preprint servers'
2. **Review** - researchers review the preprints (and share the reviews)
3. **Curate** - a journal curates preprints from already published and reviewed sources. Michael Eisen et al formulate this as 'Publication as curation.' (<https://doi.org/10.7554/eLife.64910>)



The top diagram is a journal, followed by preprint review, and Publish-Review-Curate

PRC is effectively a hybrid approach combining community-evolved preprint review processes and traditional journal processes. The preprint and community review processes assist with the early discovery and interpretation of research, while journal titles are maintained in this ecosystem as they are important "...for many researchers as they pursue their careers." (<https://doi.org/10.7554/eLife.64910>) Journals also act as a secondary mechanism for improving the communication of the results and building trust via careful curation and recommendations.

In this evolving domain, the journals are not discarded but find a new emphasis, as Stern and O'Shea identified :

"As high-volume publishing platforms continue to grow, we'll need curation services that select articles of interest for specific target audiences. Today's selective journals and scientific societies could be well positioned to provide such services." (<https://doi.org/10.1371/journal.pbio.3000116>)

Designing technology for PRC is challenging as the territory is, in effect, a fusion of multiple emergent workflow cultures including preprint review communities, 'journals-that-curate,' and preprint servers which play an important foundational role. Each of these stakeholders are simultaneously working out their own workflow and how to work together.

Journal workflows are far easier to design technical platforms for as the workflows are 'known' and journals 'own' the process from end-to-end.

There is no 'established' PRC workflow as there is for traditional journals and we have multiple interacting stakeholders simultaneously working out 'how this works.' This makes working within the PRC domain very exciting as there is a lot to work through.

eLife has recently started down this path of discovery by transforming its own publishing model to a Publish-Review-Curate (PRC) workflow (<https://elifesciences.org/inside-elifesciences/54d63486/elifesciences-s-new-model-changing-the-way-you-share-your-research>). Workflow transformation of this kind is radical — it requires the transformation of the cultural and technical operating environment of not just its own journal but the entire ecosystem in which it resides. However, eLife is in a good position to do this as recent internal analysis shows that in 2021 over 80% of papers under review at eLife were already available on bioRxiv, medRxiv or arXiv (<https://elifesciences.org/inside-elifesciences/e5f8f1f7/elifesciences-latest-what-we-have-learned-about-preprints>), that is "the eLife community has always strongly supported preprinting." (<https://elifesciences.org/inside-elifesciences/00f2f185/elifesciences-latest-preprints-and-peer-review>)

Society as an enabler for PRC

While operating under a PRC model for their own publishing operations, eLife also is encouraging and facilitating others to do the same. To this end, eLife invested in building Society (<https://society.org/>), a website that helps aggregate all of the preprint review happening across the web with additional tools for curation of reviews.

Society started as a simple aggregator giving PRC organisations a place to showcase their evaluation activities (e.g. those occurring in Kotahi), and for some, providing a home on the web to describe who they are and what they are doing. Society quickly evolved into a place that also enabled the curation of reviews by organisations (<https://society.org/groups/biophysics-colab/lists>) and individual researchers (<https://society.org/users/AvasthiReading/lists/saved-articles>).

These features of Sciety, together with Sciety's commitment to the Principles of Open Scholarly Infrastructure (<https://blog.sciety.org/open-scholarly-infrastructure/>), make it a key partner to Kotahi to assist any organization to review and publicize their preprint evaluation activities.

Design and Build Process

To aid us in the process of designing workflow for multiple preprint review communities, we decided to use the Workflow Sprint (<https://coko.foundation/articles/understanding-and-implementing-workflow-first-design-workflow-sprints.html>) methodology. "A Workflow Sprint is a fast and efficient methodology for helping organizations understand and optimize their publishing workflows before making technology choices." (<https://coko.foundation/articles/understanding-and-implementing-workflow-first-design-workflow-sprints.html>).

Workflow Sprints follow several principles:

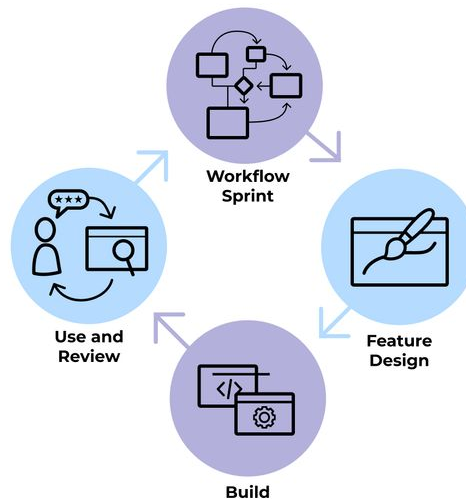
1. Operational stakeholders are best placed to describe and optimize workflow
2. Group discussion simultaneously expedites the process and adds clarity
3. Facilitation is key
4. Keep the documentation light

We leveraged this existing framework and further adapted the methodology to cater for the following:

- an emergent/speculative workflow
- remote collaboration environment
- multiple individual stakeholders, each with multiple team members
- multiple time zones

Our aim has been to work with multiple preprint review communities in parallel to co-design their workflow and build features to encapsulate these ideas within Kotahi.

We held Workflow Sprints with each preprint review community individually so they could focus on their specific workflow. We kept sessions to one hour, forcing us to focus and 'get to the point.' Workflow Sprints were followed by build processes. In its entirety, the process looked something like this:



The 4 Steps from a Workflow Sprint to Build and back again

1. **Workflow Sprint** - A series of regular remote meetings in rapid succession to design and optimize the workflow vision of the group, followed by:
2. **Feature Design** - A series of sessions to design the technology (extensions to Kotahi) that supports the workflow. These sessions focused on extending Kotahi to meet the functional requirements of the workflow.
3. **Build** - Building the features.
4. **Use and Review** - Using the features and evaluating them.

The following sections examine in brief detail each of the above stages.

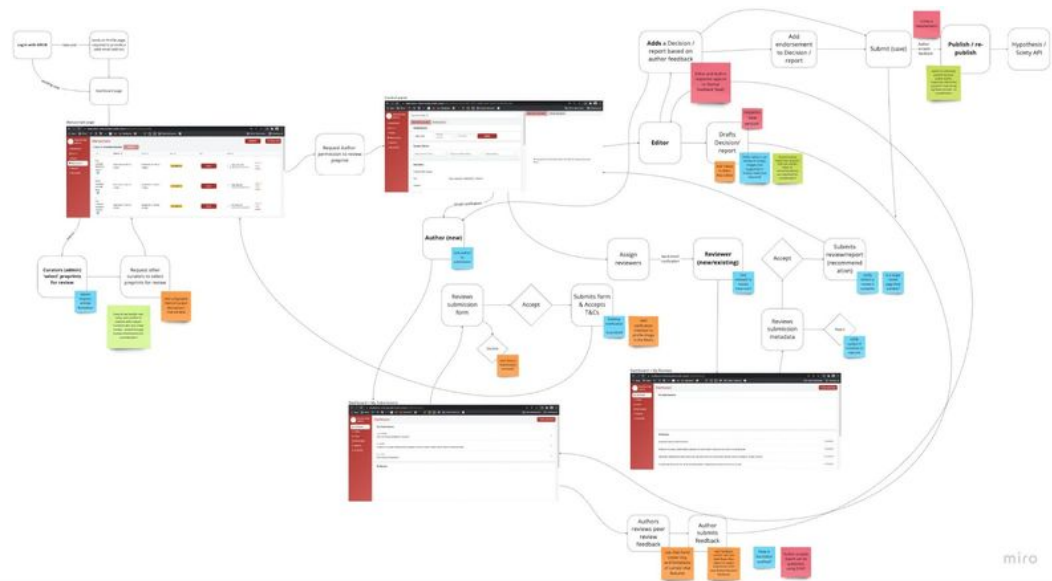
Step 1. Workflow Sprint

The fundamental principle of this part of the process is to focus on the communities workflow (how they do things) and not technology. These discussions investigated what was done now and how it might be done better.

Documentation of the outcomes were via:

1. Workflow notation in the form of WFMU
(<https://coko.foundation/articles/understanding-and-implementing-workflow-first-design-workflow-sprints.html>)
2. Some basic illustrative flow diagrams

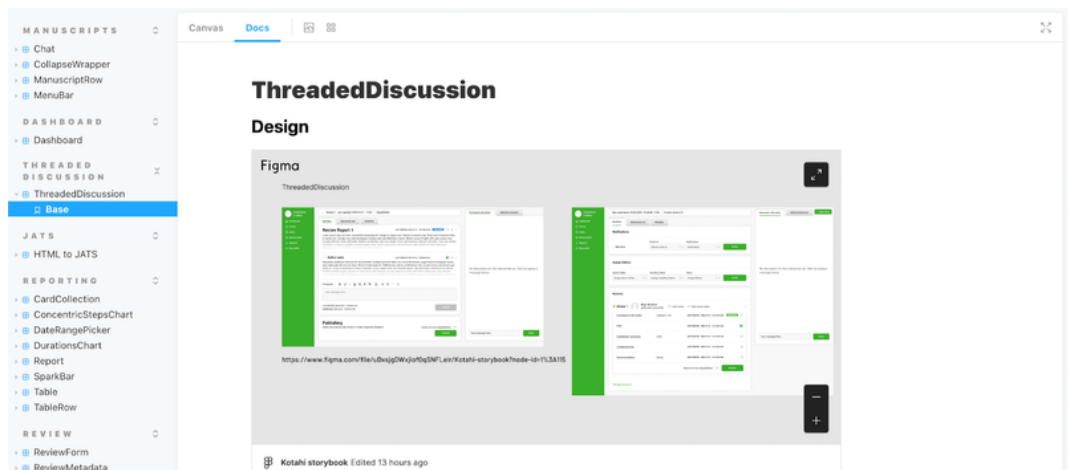
Workflow Sprint meetings were facilitated by Adam Hyde and Ryan Dix-Peek from Coko.



A diagram showing the outputs of a workflow sprint for Kotahi with process flows and screenshots

Step 2. Feature Design

Workflow Sprints inevitably generate many feature ideas to solve some of the workflow problems. In particular, eLife’s Paul Shannon and Coko’s Adam Hyde would often have creative suggestions for solutions, having seen innumerable existing solutions in varying contexts. Sometimes participants with no history of platform design, would offer better solutions. In each case, these ideas were considered by the entire group and discussed. Kotahi Project Manager Ryan Dix-Peek would document the suggested solutions and begin feature designs. These designs were then brought back to the group for discussion, improvement, and validation.



Mocks from Kotahi Feature Design for a Threaded Discussion feature

Step 3. Build

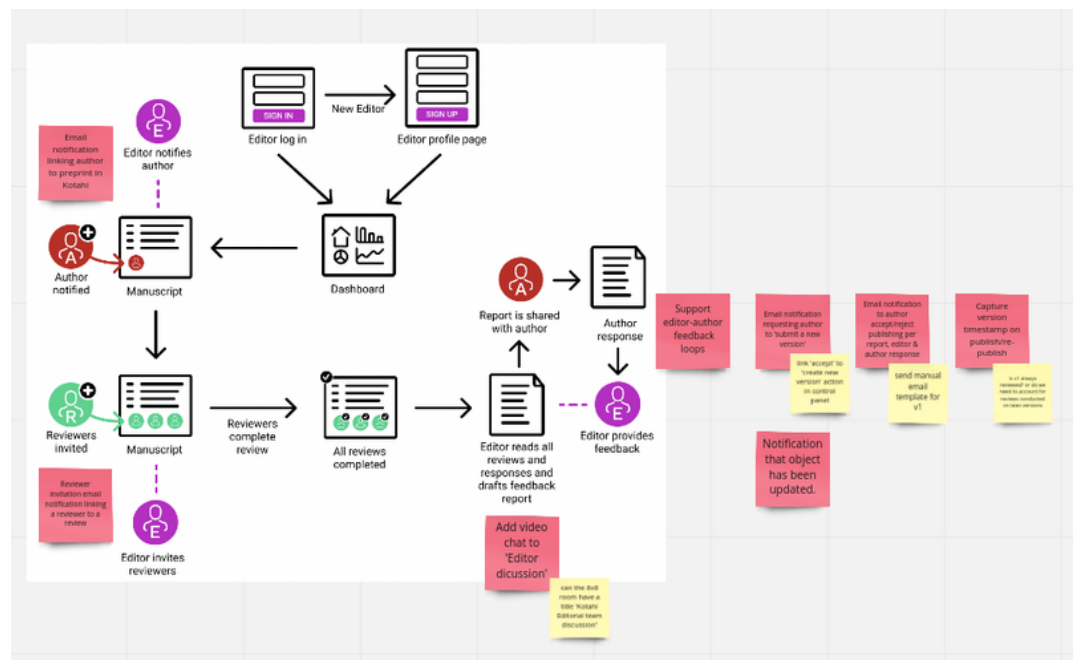
After a solution (generally a Kotahi feature of fixed scope) was agreed Ryan would create documentation for the Coko (<https://coko.foundation>) developers so they could start programming.

Step 4. Use and Review

Once the feature(s) have been delivered, the preprint review stakeholders then use the features in production. Feedback is then presented back to The Kotahi Project Manager, and the process to improve the workflow and technology would begin again.

Meeting technologies used

For these remote meetings, we used minimal technologies (mostly BigBlueButton - <https://bigbluebutton.org/> and Miro -<https://miro.com/>) and focused on discussion with documentation of outcomes often following the event.



Diagrams used to help clarify Workflow when working remotely

Most documentation resided in shared online word processor docs and shared collaborative boards (such as the Miro example above). Between meetings, Kotahi Project Manager Ryan Dix-Peek would spend time creating visuals as documentation to help present the conclusions back in the follow-on meetings

for validation. Workflow Markup (WFMU) and ‘block level’ system diagrams have both been proven to be useful tools.

Participants

The participants of each session included:

- Facilitator (Coko staff: Adam Hyde or Ryan Dix-Peek)
- Stakeholders from eLife/Sciety (usually Paul Shannon)
- Stakeholders from the PRC organization

Lessons learned

We have learned an immense amount about the PRC ecosystem and how to build technologies to support nascent workflows. The following are some top-level takeaways in brief detail.

1 . No two PRC Workflows look the same.

We learned a lot from working through PRC workflows with multiple communities. As expected, with an emergent field, there are varying visions, and there are multiple places in the workflow where there is some notable divergence in approach (with brief notes and lessons learned):

Submission/ingestion process

By submit, we mean the manual ‘submission of an existing preprint for review’. By ingest, we mean automatic processes that import preprint metadata from preprint servers for review.

It is interesting to note that PRC ‘submission’ or ingestion processes seldom require the actual preprint (PDF) itself to reside within the reviewing platform, instead preferring to link to the preprint manuscript on the hosting preprint server via DOI. In this way, preprint review processes differ from journal processes, and to some degree (caveat below), this can simplify submission/ingestion while at the same time also offers many challenges for this emergent sector moving forward (see below).

Approaches to submission/ingestion have been disparate—some review communities automatically ingest periodically (eg daily/on demand) numerous preprints simultaneously from existing preprint servers; some communities prefer authors to volunteer (submit) their preprint for review, others will reach out to

authors and invite them to submit. Still, others submit on behalf of the authors and invite the author to validate the submission before reviewing.

Automatic ingestion processes do sometimes prove problematic since there is no standardization of data available through preprint server APIs.

Consequently, if a preprint review community sources preprints from multiple preprint servers, there is no guarantee the required metadata profile can be obtained.

Since most preprint communities work within a specific, narrow research domain, the ingestion process also needs to target specific categories of research, and this data/search mechanism is not always available to the required fidelity through the existing APIs. We have, in some cases, needed to use AI services (such as Semantic Scholar - <https://www.semanticscholar.org/>) for matching keywords to research domains and then cross-referencing to other services such as EPMC (<https://europepmc.org/>) or preprint servers. This can be technically complex.

Additionally, since preprint reviews are necessarily time-sensitive, ingestion of content needs to be date filtered and we have, in some circumstances, needed to cross reference various services to determine the 'submission age' of the preprint.

Interestingly, this lack of standardization of metadata available from preprint servers has forced some communities to ask authors to 'resubmit' a preprint to a specific, preferred server to ensure consistency of available data.

Author involvement

How the author is engaged (if at all), has been an area of large divergence. As alluded to above, some communities establish a relationship with the author early by inviting the author to submit or 'self-nominate' their preprint for review. Other communities do not engage the author at all, and still others formulate the review process with multiple rounds requiring the author to engage deeply with the process.

This area has possibly been one of the most interesting as it often points towards internal cultural values. In some cases, there is a lot of concern about maintaining a 'respectful' relationship with the author by involving them in the review process. In other cases, the expediency of reviews is considered more important and the author may not be aware that their preprint is under review at all.

Triage

Some selection processes are very hierarchical and others exhibit a flatter dynamic more consistent with 'open' communities. Many preprint review triage processes look very much like a standard journal triage process, with an academic board reviewing content and deciding what should be reviewed. In other communities, preprints are 'self-selected' by reviewers to review with no prior triage process.

Reviewer selection

Some communities self-select preprints to review, others invite reviewers. Still others are a hybrid of both approaches. Some preprint communities suffer the same problems as journals when it comes to identifying and chasing possible reviewers.

Review process

Review processes can vary wildly. Some communities require one round of reviews only, others look all the world like a typical journal review process, including the expectation that authors will consider the reviews and update the preprint (on the host preprint server).

Interestingly, this is where the entire PRC flow (review community --> journal) sometimes becomes out of sync. A preprint might be undergoing multiple review rounds with a preprint community when a journal accepts the manuscript. In this scenario, the authors may consider the journal process has a higher priority and abandon participating in the preprint review process.

The format of the review also varies considerably between communities. Some communities treat reviews separately, others prefer a collaborative review process. Some communities actively engage the author in a threaded discussion and curate that discussion as part of a shared (published) author response.

Publishing

There is a lot of variance here. The format of the published data varies immensely from community to community. The content can have any mix of metadata, commentary, author responses, and 'endorsements'. Some share a synopsis of the preprint with a review or recommendation to a community homepage. Others publish to shared spaces (such as Society).

2. The arc of the PR universe sometimes bends towards journal workflows

It has at times been interesting to hear preprint communities start layering on features incrementally and, as a whole, the workflow starts looking more and more like a typical journal workflow. This has been fascinating to see as sometimes PR communities are antagonistic to the value metrics and inefficiencies they believe the current journal ecosystem supports. It has, at times, been necessary to challenge some preprint stakeholders on their priorities. Are they trying to discard/change/improve the current way of doing things, or are they layering legacy journal processes on top of preprint servers?

3. Preprint Servers are increasingly becoming a center of gravity

It has been clear throughout this process that there is a lot of reverence from the researcher community towards the various preprint servers which service their domain. Preprint servers are the technical, social and content foundations upon which PRC is built, and consequently, preprint review communities and journals-that-curate are both reliant on the ongoing good faith and technical viability of the preprint server teams. The question going forward is whether the preprint servers will be able to meet and maintain the growing requirements of the emergent PRC processes. In particular, issues of metadata standardization, process synchronization, and external service integrations (incoming and outgoing APIs) are critical for the advancement of this sector.

4. Software is a conversation

We have built features into Kotahi for each workflow requirement which has required a lot of conversation and co-design. Building a scholarly publishing platform that can manage this high degree of process variance is not the simplest of tasks.

We have achieved support for all of these disparate workflows, primarily because of good faith, open discussion, and the co-design principles of Workflow Sprints.

We have often found that a feature that initially seemed like a large variance with current features turns out, after many hours of drilling down into the problem together, to require only a small extension to the platform. If we were too quick to design and build, or we abandoned the principles of co-design, we would have many complex features in Kotahi that ‘approximated the solution’ and did not actually solve the problem. Thankfully this hasn’t been the case.

It is also true that sometimes the preprint stakeholders have a high-level idea of what they are trying to achieve, but when we drill down, this reveals contradictions or consequences they find difficult to accept. This has led to many interesting workflow, technical, and philosophical discussions about what the community is trying to achieve.

At other times, we have needed to build in abstractions to allow for configurable options when workflows are clearly divergent and require new features.

Thankfully, the Workflow Sprint design and build processes have lent themselves well to meeting the needs of each community's individual visions of PRC workflow. The process has allowed us to front-load discussion and discovery in an expedient, collaborative manner. It would be a mistake to adopt a development process that designs too far ahead when it comes to features that support disparate emergent workflows.

5. PR, RP, and PRC can all coexist in the same platform

One of the design philosophies of Kotahi is that all submit-review-publish processes look the same from a 'bird's eye view', and it is possible to build a system that can support a 'domain agnostic' approach. From a certain distance, for example, funding review looks very similar to both journal workflows and preprint community review workflows:

1. Submit
2. Review
3. Improve
4. Share

Kotahi was originally designed as a journal system through a series of Workflow Sprints with the journal Collabra: Psychology.[22] eLife recognized that the 'bird's eye view' design philosophy could work for supporting PRC workflows, and the collaboration began. These experiences have really tested the ability for a single system to support a domain-agnostic approach (above) and we are happy to validate this approach so far. Kotahi can support an enormous amount of variance in Journal and PRC workflows. With Kotahi quickly becoming the platform of choice for PRC communities, and with a growing number of organizations (notably publishing service providers) using Kotahi for journals, we have a lot of faith in the Workflow Sprint processes and the Kotahi platform to fulfill and support these disparate needs going forward.

Summary

Designing scholarly technology to support emergent workflows is initially a conversational process that leads to shared concepts, documented (speculative) requirements, and an evolving shared language. When these foundational elements are, at least partly, established, the process can begin to incrementally move forward with initial technical designs.

The technologies built from these designs, in turn, lead to 'real-world' experiences which feed back into the conversations, and we witness the gradual but progressive discarding and replacement of legacy ideas, language, and processes.

One of the challenges with this process is designing and building a technical platform that can cater for all these divergent workflow pathways within the one system. This has required a lot of careful technical design decisions by the Coko (<https://online.ucpress.edu/collabra>) development team. The pressure of these real world disparate requirements is welcomed and has benefited the platforms architecture and utility immensely.

PRC is a series of connected and emergent workflows with no fixed approach. This presents many challenges for platform development intended to encapsulate these processes. The eLife-Coko collaboration has explored this in depth through the design and build of features to support a large variety of PRC workflows in Kotahi. Our findings essentially validate that it is important to front load the workflow co-design process before building features and testing in the real world. We have found that patient discussion and co-design have been the main attributes of our success in building a platform through which these communities can collaborate to expedite the sharing of reviews and research.