

# Final Report

---

## Flip Systems

### Modelling Editorial Workflows

Annie Chen | Adam Gaumont | Leanne Johnson | Jenna Newman | Michelle Simms

April 3, 2009

---

Please refer to these slides and screenshots:

[Flip Presentation.pdf](#)

---

## Introduction

The goal of Flip Systems was to draw from various publications' current practices and develop a model for a workflow software system that would allow different users to process, manage, and collaborate on content from submissions to editing to output, resulting in increased efficiency and turnaround time.

In order to gather as much information on editorial workflow as possible from a variety of publications, we conducted eleven interviews with twelve people representing ten different publications or presses. We would like to thank Susan Safyan, Brian Lam, and Robert Ballantyne, Arsenal Pulp Press; Tyler Lokken, Athabasca University Press; Marilyn Bittman, *Canadian Journal of Communication*; Ian Cockfield, *Event*; Leanne Johnson, *FRONT Magazine*; Mary Schendlinger, *Geist*; Joy Gugeler, *Room*; Melissa Edwards, *Room* and *3-Day Novel*; Geoff D'Auria, *The Tyee*; and Holly Keller, UBC Press. Their generosity and willingness to share information about their work were instrumental in the success of our project.

Flip Systems created a standardized questionnaire to use in our interviews which was based on the questionnaire created by Wesley Fok, MPub graduate, in his interviews of several magazines (mostly Ontario based); we are grateful that he shared his work with us. From our interviews we created a diagram for each publication's workflow. We identified and compared strengths and challenges, and we uncovered common difficulties, practices, sequences, roles, relationships, and variations. Based on these findings, we created a general model of the workflow process. Our model consists of a master diagram which integrates the basic components in flowchart form, and a series of wireframes that explain these components in more detail and show various interfaces that would be needed throughout a workflow software system.

From our interviews, we discovered that publications have adapted or developed various specific sequences and practices in acquiring submission material and preparing it for publication. However, we concluded that at a general level, certain broad steps always proceeded in a linear

order: items submitted were evaluated, entered a period of negotiation, were edited, then prepared for output (i.e. went through layout), and finally could be published. We rely on this linear progression to drive material through our model. Within each stage, we allow an individual user to select the specific tasks or steps relevant to their process. These can be as hierarchical or controlled as the user wants, and can be configured easily by the publications' managers. This flexibility within the broad linear sequence enables our model to apply to a variety of publication practices. We hope this approach will encourage buy-in among the managers, editors, and other staff who use the system by allowing flexibility that will not make them adhere to superfluous or over-managed reporting processes.

---

## **The Model**

### **An Overview**

In our model, we have colour coded the four stages of the workflow for ease of reading. The diagram (page 3 of the attached PDF wireframes) shows a basic workflow sequence of the evaluation, negotiation, editing, and output stages. It's important to keep in mind that a publication could expand or collapse each stage as needed to capture explicit tasks, sequences of steps, and approvals. The diagram shows a number of other options that could be included in the editing stage. Similar sets of options exist for the other stages as well.

The wireframes demonstrate some of the possible variables suggested by the diagram, such as number of tasks, requirement for approvals, or use of automated approvals or advancements. Again, any combination of these variables are possible.

We have imagined three levels of user permissions: reader, editor, and manager. (Please see the attached glossary for a discussion of these and other terms we use.) Most of the publications reported a flat organizational structure where a single person may take on various roles at different times, so we did not see a need for many different types of access restrictions. Publications with more formalized staff structures can of course add new types of users. At the other end of the spectrum, a publication could make all its in-house staff “managers” and rely on their knowledge of the house practices to guide their behaviour, with slightly restricted access (e.g. editor roles) for freelancers.

The following narrative explains the step by step process represented in the wireframes. The relevant sections of the attached PDF file are referenced by page number as they arise.

---

### **Submission and Evaluation**

Based on our research, we found that although several content management systems exist, few are geared towards submission-processing. Of all the publications we interviewed, only two had a digital submission system; the rest relied on either post or email, with documents tracked via spreadsheet or a checklist. However, most interviewees said that they were not entirely satisfied

with their submission processes, which suggests that some could benefit from the more automated and less paper-intensive digital system we've designed.

In order to submit a document, contributors would normally have to create an email-based account, which could be saved for the next time they returned (page 5). After entering their contact information and uploading one or more files (and/or a batch of text), they would then receive an automatically generated email confirming the receipt of their submission and indicating an anticipated response time.

Once the system receives submissions, it automatically generates a unique document ID for them, and marks them in the submission bucket for readers to access and review. For example, acquisition editors at a book publisher could sign an item out, give it a rating, and sign it back in for the next person to read; at an academic journal, external peer reviewers could review each others' comments and write their own; for a literary magazine contest, volunteer readers could each log in and rate a submission without seeing what others have said.

Each staff person using the system has their own user ID, and all of their actions would be tracked (an automatic log file can record users, time stamps, and actions). They have to log in to use the system, and have to check the item out in order to maintain version control, although anybody can view an item at any time. Freelancers, external reviewers, and other non-employees might have more limited access than in-house editors, and managers would have the ability to toggle whether information such as comments and ratings would be visible to others.

All submission items arrive in the submission bucket, which means they are accessible from the Evaluation Home Screen (page 6). They may or may not have a project ID already; if not, they will be most likely associated with one during this stage. (Project IDs are defined by a manager in the administrative interface, but can be associated with individual documents from the task screens; departments are treated the same way.) The "All due" date is the deadline for the project as a whole, and can be displayed as a reminder. This date could be set and changed only from the administrative interface.

The task displayed in the home screen for an individual document is the current task: as soon as a task is complete, the home screen will display the next task for that stage. ("Next task" is set by a manager for the entire system or for individual projects as required.) This means that every submission will enter the system in evaluation with the first task already listed, in this example Review 1. The task may or may not be assigned to a specific user; unassigned tasks can be claimed by whomever is eligible and available to do so. A task's due date can be automatic (e.g. perhaps no submission can be in review or evaluation for more than 60 days) or can be set each time. The availability column indicates whether the document is signed in or out to prevent version conflict. A signed out item could be viewed as read-only by other users if necessary.

Martha Reader, a user, comes to the Evaluation Home Screen and selects document 01010 (page 6). By clicking on this row, she will go to the task screen for the current task: Review 1 (page 7). The task screen describes where the user is at the upper left (project, department, stage, and task); the user's name will appear whenever they are logged in at the upper right, next to a button to return to

the user's dashboard. The actual text of the item appears in the centre area: we recommend including a text editor function within the workflow system rather than using files as attachments. This will help ensure the integration of the workflow management process with the actual work being done; we wanted to avoid having a user go to a separate management program as an additional step.

Martha can read the entire submission item, write comments, and finally rate the item. In this example, we have allowed yes, maybe, and no; this could be a ten-point or some other scale instead. A manager would specify what ratings would lead to the red bucket (rejection), while everything else would lead to the yellow bucket (possible acceptance). Some of the item's metadata is displayed in the lower left area (title, author, type of file, word count; department and project can be changed to another option from a drop-down list here too). When Martha selects her rating, she is returned to the home screen.

Now the evaluation home screen displays the next task for document 01010: Review 2, assigned to George Reader (page 8). George can log in to the system, navigate to this home screen and select document 01010. He will then go to the task screen for Review 2 (page 9). George has access to all the same functions Martha did. In this example, he can see who did previous reviews and what they scored, but this information could be wholly or partially hidden. The “No further reviews” check box allows a reader to identify the item is clearly not appropriate for the publication at any point, potentially bypassing the rest of the task sequence for this stage. By checking this box, the document will automatically go to the Decision task (i.e. yes or no must be selected from the “Advance?” column in the home screen). If a publication has a policy of two reviews and then a decision, no further reviews would be automatically checked once two reviews are complete.

Once sufficient reviews are complete, or the no further reviews box is checked, the home screen will show the next task as Decision (page 10); it may be that an editor or manager must perform the decision rather than a reader, and these permissions can be set in the administrative interface. The final “Advance? Yes / No” column records the decision that completes the evaluation stage. Checking the yes check box will mark the item as in the yellow bucket and move it to the negotiation stage.

Checking the no check box (page 11) will mark the item as in the red bucket. In this case, a rejection letter template could appear as a pop-up to allow an editor to email the contributor and inform her that her submission is declined (page 12). The template is defined through the administrative interface, but can be personalized before sending. This rejection letter could be fully automated (e.g. three no's and it's out) or require explicit action by designated users.

---

## **Negotiation**

After a submission item has advanced from the evaluation stage, it is in the yellow bucket and will appear in the negotiation stage (page 13).

The negotiation stage involves some form of discussion between two or more people: an informal

discussion between an editor and publisher; an editorial meeting where the merits of a submission are discussed in relation to its inclusion on a season's list; a conversation or series of emails between a publication and contributor regarding possible changes or additional material; contract discussions—any negotiation that takes place before a submission is formally accepted for publication and editing can begin.

Individual negotiation tasks may be specified by the manager if desired, but we appreciate that even within a single publishing house or publication, the negotiation stage can take many forms. The only task we have set for negotiation in our general model is that a negotiation of some kind takes place and a formal yes or no decision is made before the editing process begins. We have also included a comments log that will facilitate tracking communication and decisions.

In our wireframes, we have set the first (and only) task as "Discussion". This will appear as a default once a submission has reached the yellow bucket; it may not be assigned to anyone, but usually will be assigned to the person responsible for contacting the contributor or the person presenting it at an editorial meeting. Often this is an editor and the user who is most familiar with the submission. At the negotiation interface, a user may see all submissions at this stage, or just the submissions that have been assigned to a specific project by selecting the appropriate filters from the drop down lists in the upper left.

From the Negotiation Home Screen the user can see all the submissions in negotiation for acceptance in Issue 4 (page 14). Submissions could be sorted by author, title, or by the person assigned to them. Here, Alice Editor can click on a submission—in this case, one that she has been assigned or assigned to herself and hopes will advance into editing. This takes her to the task screen for Discussion (page 15). Here, she can review the submitted text and can see that it has been marked as a "fiction" submission for "Issue 4." If during negotiations it has been decided that the submission would be better suited to Issue 5 or 6, Alice can change that here from a drop-down menu (lower left); she can also move the submission to a different department if needed. Any comments from the initial readers appear on the right, and if Alice wants to track requests to the contributor or future editing notes, those could be noted as well. (This would be useful if her publication worked with freelancers not involved in the negotiation process.) All of the activity of Discussion will, in this example, take place inside the comments field.

The "Accompanying data" tab tracks metadata (page 16): whether an author bio or photo has been received, contact information, the status of a contract; any keywords that have been marked, and accompanying documents associated with this submission, such as images. This tab is accessible from all the task screens.

If an editorial meeting has taken place and this submission has been accepted—or whatever other steps that are required—Alice clicks "Discussion complete," and that brings her back to the Negotiation Home Screen where she advances the submission to the editing stage by clicking "Yes" (page 17). In this case, Alice has permission to decide that discussion is complete and to formally accept a submission. (The manager may configure the system so that Alice has permission to note that discussion is complete but not to formally accept or decline a submission.) From this screen, clicking "Advance? Yes" or "Advance? No" could trigger an email to the author or to another editor.

Here, once Alice clicks “Advance? Yes,” the submission moves to the green bucket and is ready for editing.

If a publication had a formal structure for negotiations, such as a scholarly press, the negotiation stage could include and track each task: author's replies to peer review comments, receipt of revised submission, acquisition meeting, review by publication board, submission of grant funding request, etc.

If the submission is declined after these negotiations, it is moved to the red bucket, and could trigger the same form rejection letter as followed from the evaluation stage.

---

## **Editing**

All submission items that are being accepted for publication—those marked “Advance? Yes” in the negotiation stage—are now in the green bucket, which means they will appear in the editing stage (page 18).

The individual tasks in editing are specified and ordered when the manager configures the workflow system for their publication. In our wireframes, we have set the first editing task as substantive edit and the second task as copy edit. Note also that in our example, a manager has configured the system so that after these two rounds of editing the document is then automatically marked “ready for layout.” The manager could also have configured the system to explicitly track a decision step where someone would have to approve the document by checking off the box from the Editing Home Screen.

Alice clicks on a document that is assigned to her (page 19). This will sign the document out to Alice, and take her to the task screen for substantive edit (page 20). The document is available for text editing, with changes tracked as they would be in a wiki. The system will auto-save as she works. She can also collapse her change history by clicking on the “save as new revision” button. If she needs to compare versions, she will click on the “revision history” button and a list of versions will appear; she will select two and be able to see their differences.

The comments area is where Alice records her actions with respect to the document: that she has queried the author, that additional information will be supplied. Specific edits or queries will be contained within the text, but notes and statuses (i.e. that Alice is waiting to hear back from the author) are recorded in the comments field.

When Alice has finished the substantive edit, including resolving any outstanding issues with the author at this stage, she will check off the Substantive Edit Complete check box (page 20). By checking this check box, Alice moves the document to the next task. She will return to the editing home screen, and the document will be signed back in (page 21).

The second task in this example is copy edit, and it has been assigned to Bob. He logs in and clicks on the document assigned to him (page 21). This will sign the document out to Bob, and take him to the task screen for copy edit (page 22).

Bob can see that it was Alice who did the substantive edit. He can also see a collapsed view of whatever comments Alice might have made during her edit. If he needs to review them, he can click to expand them. Bob has the same access to revision history, save new revisions, and write new comments functions as Alice did. When Bob has completed his copy edit—which may also include his own queries to the author—he will check off the Copy Edit Complete check box. This moves the document to the next task and signs the document back in to the system. Because the manager configured this system to automatically move to “ready for layout” once the copy edit is complete, “Advance? Yes” would be checked off automatically and the document would then appear in the output stage.

---

## **Output**

All items that have advanced from editing are marked in the layout bucket and appear in the output stage (page 23). Once the document is ready for layout, the designer will either be notified that documents are ready, or she may log in to the system and check if documents are ready. The Output Home Screen lists all documents that are ready for layout (page 24). When the designer selects a document, instead of going to a task screen, this action can export the document into a design program. Once this is done, the document is checked out (permanently). This means that a user can still see the document but cannot edit or write to it. (This can be changed from the administrative interface if necessary).

The document might now leave the system, and the tracking can end here as well depending on the needs of the publication. If the manager has the proofs in front of her, she won't need to notify herself, but if the layout and proofing are happening remotely, then she can continue to track or have the proofreader track the proofing process. In order to streamline this process, these tasks (first proof, second proof, etc.) can be tracked from the Output Home Screen in the final Advance? column and can be set to automatically apply the same status to all items in a project (page 25). Once a project is complete, then it is archived in the system.

---

## **Setup and Administration Interface**

All staff will first log in to the system (page 26). Their landing page would be their personal dashboard: it looks very much like the stages' home screens, but gathers all the documents, in all stages and all projects, assigned to that staff person (page 27). A status bar on the right shows how far along in the entire process that document is, allowing the user at a glance to see progress to date. In the personal dashboard, any notifications directed to that user will appear.

Previous screens have shown documents filtered by stage; a user may also filter by project (page 28). In this example, when “All Stages” is selected, the status bar appears on the right instead of the Advance check boxes. A manager can select which columns appear in the various views from the administrative interface.

From the back end of the system, a manager can define and change the settings for the system and

for the projects/documents (page 29). These settings are:

System:

- Activate/deactivate users
- Browse /search users
- Set user permission levels
- Manage tasks (configure task sequences)
- Set triggers for notifications
- Assign tasks
- Set submissions requirements
- Set allowable file types
- Set rating system
- Set submission expiry dates
- Toggle data fields

Projects/Documents:

- Generate report
- Create new project
- Manage projects
- Delete a project
- Delete a document
- Empty red bucket

For example, in the “Manage tasks” setting, the manager would be able to customize the task sequence for each stage. She can add a task to the existing sequence or remove a task from it, as well as create a new task or rename it (pages 30–33). The sequence can be saved as a preset, so there is no need to rebuild it again for the same task sequence in a new project. In the “Set triggers for notifications” setting (page 34), the administrator can determine whether or not a task completion would trigger a notification, and which user will receive the notification for that specific task. In the “Assign tasks” setting (page 35), the administrator can either assign tasks to different users, or leave them unassigned (let users assign the tasks to themselves).

Through these settings, a manager can customize the model to meet her publication's needs in terms of editorial workflow.

---

## Conclusion

The model we developed combines a tracking and notification system with a collaborative text editing system.

An important feature of the model is that it's a centralized repository. Each person comes to the system and does their work inside it; there's no need to confirm or schedule activities or track progress anywhere else.

The model also covers the workflow from submission all the way to output. As a single integrated system, our model eliminates the need for editors to re-key submission metadata or to track accompanying documents (such as an author bio) separately.

The four linear stages are broad enough that they are common to all publications, and these provide the structural backbone of our workflow model. Within these stages, we have allowed individual publications to completely customize the configuration of discrete steps or tasks. A manager can add, remove, and re-order tasks at any time through the administrative interface.

The manager can also specify what level of detail she wants to track. She can set notification triggers on certain tasks. She can choose if a task automatically triggers a sequence of actions (other tasks), or if every task must be explicitly performed. In this way, our model lets the manager decide how workflow software can support her publication's existing process. The model does not make anyone jump through superfluous hoops.

We hope that a model such as this one is flexible enough to apply to any type of publishing proposition, and that it offers benefits to everyone involved in the workflow by streamlining administrative tasks and promoting efficient collaboration at each stage.