# *Can we do away with RFPs?*

## *Origially pulished on Upstream*

Published: Feb 1, 2022
Author(s): Adam Hyde
Illustrations: Henrik van Leeuwen
PDF Design: Harshna Haswani
Rendered from HTML using Paged.js

Scholarly publishing at times seems to want to hold on to some outdated processes. At the top of my list is the dreaded RFP (Request for Proposals) process.

In the scholarly comms world, consultants send these out on behalf of publishers that are seeking solutions or services. My career involves designing and building publishing technology and so I have been on the receiving end of many RFPs for developing new platforms.

An RFP to discover a partner to build software generally contains an endless bucket-list of necessary features ('requirements'). Some of the RFPs I have seen literally list items such as 'JATS export' or 'automatic notifications' as features and I'm supposed to check if we can supply that functionality or not.

Export

- □ JATS Export
- □ PDF
- □ other XML...

But (from this example), let's face it... all journal platforms require this functionality. I'm not sure I understand the rationale to indicate that we can provide features that belong to every publishing platform in their domain. Sure, there are sometimes unusual requirements but at the end of the day we can 'supply' (i.e. design and build) any of those too.

## *Let's Design Your House*

Let's zoom out a little. The problem is perhaps easier to understand if we reference the more accessible notion of house architects. In the process of identifying an architect to design my new home, imagine if I sent them an initial RFP that said: “Do your houses have bathrooms?”, “Can you design living spaces?”. Or, even closer to the scholarly communications reality, I sent them a list that just read 'bathrooms', 'living rooms' and the architect should tick off those they can 'do'.

House

- □ living room
- □ bathroom(s)
- □ other rooms

I hope you can see why this might be considered a waste of time for everyone involved. As an architect, the best I could do would be to answer “yes” to every question - we can design and build anything. But what a good architect actually wants to do, is to design and build a solution that meets the present and future needs of the client perfectly. To design systems like this means an architect needs to understand the commissioning organisation's legacy, culture and ambitions.

## *Function vs Culture*

Imagine it like this (now I'm going back to the house design): as a house architect, I could deliver you a design immediately after answering your checklist. That would actually be quite easy. Most houses are more or less the same from this checklist-level description. They have (for example) bathrooms and living spaces. So, going by this logic, should my design fulfill those functional requirements (i.e. the house has a bathroom and living room) then it follows that you should be quite happy with the final product.

It might be a pity that I actually put the bathroom at the opposite end of the house from the living room - you might have wanted it closer for convenience sake. Or perhaps you have a nice view and I built the living room looking out onto the back fence ("...the laundry has a superb vista!"). But all that doesn't matter because the checklist is fulfilled.

I don't think you want an architect of your new house to work like this. I imagine you probably want the architect to sit with you, to gain a much deeper appreciation of what you are trying to achieve. For instance, you like the view out the front gate but you also want some lawn out front for the kids? No problem; we set the house back on the section and put a big window at the front-facing living room...etc...in other words, to make a good design, the architect has to understand not just the functional requirements (which are more or less the same for every house), but more importantly (much more importantly), how you want to live.

'How you want to live' is a personal and cultural question, and in the same way so is the question 'how do you want to work?'. When I am asked to design systems for publishers, the technical function is pretty well understood before we meet. Sure, there are details and nuance to work out, but the more important question is 'how do you want it to work?', which is also a personal and cultural question.

Let's take an example from the journal world : reviewer invitations. Sure, inviting reviewers via email is something most journals have, but there are quite a few questions a good design must consider that may be answered differently by publishers, depending on how they like to work. For example: do you want to use templates for the emails? If not, how and where are the invitations authored and sent? If you do use templates, can they be customised? If they can be customised who can customise them and to what extent? Do you allow emails to come from 'out of system' (e.g. from an editor's personal email address)? If so, how do you then circumvent automated email invitations? How about other channels for such invitations / notifications? Would you like notifications/invitations to be injected into a Slack channel, for example? In the future do you wish reviewers to self-select from group invites? etc…

There is a lot to understand here about how the publisher wants to work that will inform the design of every aspect of the system from the trivial (e.g. email notifications) to the complex (e.g. author revisions, reviewer rounds, submission form customisation etc).

## *Constraints*

There are also questions an architect needs to understand around constraints that you, the client, may face. The trick here is that you might not realize immediately how these constraints may inform your design. Let's consider the house example once again: a good architect can only design your ideal house with you after they understand some of the major constraints you may face. Some of these constraints might be financial, some might be practical e.g. the nature of the landscape you are building on: If you want a single-level pole house built on a radical incline, the architect would do well to inform you of the level of engineering and costs needed to achieve this, or whether this 'requirement' is in tension with any of your other requirements (e.g. like having easy access to the ground). These issues are best understood through conversation - arguably the most efficient way to help everyone involved come to the best suited solution together.

Similar issues exist in publishing. If you state, for example, you wish to have both LaTeX support throughout the submission and peer review workflow AND you wish to achieve browser based Single Source Publishing…then I think it would be good if we discussed this in detail so you can understand clearly the constraints and tradeoffs, and consequent development time and costs that would be involved. You can then make your decision on whether it is worth it or not.

## *An Improvement*

I have been on the receiving end of good RFPs but they are rare. Most recently I was asked to respond to functional use-cases, i.e. as a user, I need to do x. I liked this process. Responding to this as an architect, gives me room to move. I can respond in short-essay form by highlighting the design issues that must be considered per use case; I can juxtapose some design challenges and trade-offs, and I can ask questions that will help the publisher think through what it is that they actually are trying to achieve.

This is a much better approach than the box-ticking RFP. From the beginning the RFP is trying to determine how the architect thinks and how they approach a problem. From the response provided by the architect, the publisher can also understand clearly whether the architect clearly understands the problem or domain. While the functional use-case example is better than the checklist approach there are still better ways to request proposals. In my opinion the best way is through conversation. You could learn a lot more about the suitability of an architect in a single hour conversation than you can learn from either the functional use-case or checklist approaches.

In other words, find an architect by simply talking to them and then involve your chosen architect in your requirements discovery process. This is a necessary and critical shift that would help publishers get systems that better meet their needs and open up solutions they might not have yet considered.

## *Lets Fix it*

I was recently asked by a group of publishing consultants to have a chat about 'things in general' - it was a welcome process and I had the opportunity to set the agenda. One item I put on the table was the RFP process. I made clear my reservations about checklist documents and I was very surprised to hear them respond likewise. Their point was that publishers are driving the RFP process.

That feedback was very interesting and inspired this post. It would be great to think through an effective RFP process with the scholarly comms community.

Assuming this stimulates discussion, then I will post more on this topic. I want to talk about avoiding pre-loaded assumptions in RFPs and how they can exclude good solutions and approaches from the very beginning.