# 2012-05 Reflections & Brainstorms

This is a set of brainstorms and not a formal proposal

The following are a series of brainstorms around the 3.0 release. They are meant to generate discussion amongst the developers/committers around various future development options. As such these ideas should not be considered as being in line with the eventual 3.0 plans or the views of the Committer Team.

## Some thoughts on the DSpace '3.0' release. Richard Rodgers, May 2012

Our decision to end the 1.X line of DSpace releases should encourage us to stand back and 'take stock' of where we want the platform to go and how we intend to get it there. One aspect of this is dispassionately to appraise the major components of DSpace with respect to currency, power, sustainability, etc, and also to assess our community development resources, both active and potential.

From this standpoint, the XMLUI raises significant questions. XMLUI is built upon the Apache Cocoon XML publishing framework, specifically version 2.2 released in May 2008 (4 years ago). Since that time, no Cocoon point releases have been issued, and scant attention paid to reported bugs, etc. This was presumably because the focus had shifted to Cocoon 3.0, whose first alpha appeared late in 2008. The fact that 3.0 has not progressed to beta status in over 3 years should raise many flags, and we should ask ourselves whether there currently remains a viable developer community, or there is a strong likelihood that one will emerge.

What does this mean for DSpace, and specifically the 3.0 release? We cannot, of course, simply 'unplug' XMLUI in favor of another UI, since there are no realistic alternatives of comparable power and maturity. And we will in any case continue active support of XMLUI for some time, as it has been widely deployed and customized since its appearance in DSpace 1.5. We can and should, however, ask questions about what our 'strategic investment' (i.e. future development efforts) in XMLUI ought to be going forward given the reality sketched above. And the next DSpace release gives us a tangible, concrete, near-term context in which to explore alternatives. We must be mindful throughout that our own technical resources are rather limited, and must be directed to where they an be most effective. For this reason, options like forking Cocoon and maintaining it ourselves, or reimplementing XMLUI without Cocoon, etc, seem dubious to me.

What follows is a series of rough alternative models for a 3.0 release, and some notes about risks/benefits of each (thanks to Tim Donohue for most of these insights), that attempt to capture some of the possibilities and trade-offs. I'm sure there are many others, but these should suffice to provoke reflection and discussion.

1. Status Quo or Evolutionary Release -> Release 3.0 as another evolution of DSpace with XMLUI (as default UI) support (similar to all 1.x releases).
    - Pros:
        - It's what is expected. It's the norm.
        - Provides a straightforward upgrade path for current adopters.
    - Risks/Costs:
        - If not now, when do we do a UI/architecture redesign?
        - Will DSpace development activity start to "die away" because the platform is less exciting to develop on?
        - Resources must be devoted to XMLUI when our intention may be to eventually replace it (i.e. robbing cycles from new UI development).
2. "RESTfUL Rapid Redesign" -> Redesign the underlying architecture of DSpace, to develop a "next generation" platform that would allow us to more easily support alternative UIs, inter alia. As this redesign would be a large project - it would only be worth doing to add significant new system capability (like metadata on all DSpaceObjects) - the end result would likely lack a UI by the 3.0 scheduled release date, but it would have a REST API (on which one or more UIs can be built/backported), and migration/upgrade path.
    - Pros:
        - Potentially exciting project? Can we get others excited?
        - Could be re-energizing – increase development on DSpace platform (new UIs, apps/widgets, etc)
        - Needs to happen sooner or later. May be difficult to do incrementally?
    - Risks/Costs:
        - This binds evolutionary features (metadata on all DSpaceObjects) that can be delivered in a 3.0 timeframe into a larger project to re-envision dspace. Its going to create a bottleneck and not utilize existing work in the community as effectively as it could.
        - Past attempts at redesign DSpace show that unless there is a complete solution that is adopted wholly by the developer community, there is a significant risk of non-adoption at the end of the project.
        - What will broader Community think if it has no UI? How do we "sell it" to them?
        - How does it relate to "DSpace with Fedora Inside"?
        - Do we have enough developer time?
    - Valid Questions:
        - Can we even call this 3.0? Is this more of a "beta" release?
        - Is this still "DSpace"? Or is it something else?
3. "RESTful Rapid Redesign + UI backporting/creation" -> Rapidly build the new platform described in #2, but also backport at least one UI (likely XMLUI). The backported UI may have limited/no support for all new platform features (especially if the new platform were to enable features that were not currently present in the UI).
    - Pros:
        - Same pros/risks as #2 above
    - Risks/Costs:
        - May not be possible by Fall? What is the backup plan?
        - Does this mean 3.0 in 2013? No release in 2012?
4. Dual Release: 3.0 + RESTful Rapid Redesign (Almost a combination of #1 and #2 above). The idea would be to release a 3.0 using minimal work /resources (just releasing features that are "ready at hand"), while simultaneously having another developer team work towards a rapid redesign of the DSpace platform (as described in #2 above).
    - Pros:
        - Allows us to continue to support broader community with 3.0, while also working towards the next generation platform
    - Risks/Costs:
        - Stretches resources even more. Would need to limit 3.0 to features that are "ready at hand".