

CommitterPhoneConf01Nov2004

Notes from committer phone conference, 01-Nov-2004

Jim+Downing, Richard Jones, Ralph LeVan, Gabriela Mircea, Richard Rodgers, Robert Tansley (notes), Scott Yeadon

Committer's available resources to work on DSpace+2.0

Richard Jones:

Theses Alive project coming to an end – probably take about a month or so to finish up. Then becoming official library staff.

Continuing to support & work with DSpace – most ongoing work is going to involve DSpace – e.g. SHERPA Edinburgh research archive

Not sure how much time to work on DSpace+2.0 – would like to be involved in discussion – may be able to contribute development if in alignment with local project

Ralph LeVan:

Continues to have interest in interoperability – OAI & SRW
Longer-term interest in authority control
DSpace+2.0 – Interest in discussion, though probably not development

Jim+Downing

Very keen to be part of DSpace+2.0 effort – likely 1/3 of time to DSpace+2.0 until the end of 2005

Gabriela Mircea

Only DSpace tech person on Toronto
Soon starting 2nd project – can spend 1/2 time on DSpace
About 20% of this (10% total) could be spent on DSpace+2.0
Main interests are modularity & internationalisation

Scott Yeadon:

Very interested in DSpace+2.0 – can spend 75%-100% of time on DSpace 2.0 (.75 -> 1)
ANU also has 2nd developer – Australian Partnership for sustainable repo's based around DSpace

Richard Rodgers:

Richard hopes to spend 2/3 -> 3/4 of time to DSpace+2.0 development, particularly storage layer (as part of project to integrate DSpace + SRB)

other MIT folk: hopefully 25% of FTE – dependent on alignment with local requirements

Potentially 50% FTE from SIMILE team, as DSpace+2.0 is intended deployment channel for SIMILE technologies and they have vested interest

Robert Tansley:

Full time DSpace+2.0 – likely to be only HP contributor to DSpace 2.0

Discussion of process

Jim: don't think we have enough committers for a real Bazaar-style approach

(post-meeting addendum) I think the *development* should be open. If and when the 4-5 man project materializes it will inevitably have a profound effect on the direction of the code development. It doesn't need exclusive access to the code, and 2.0 doesn't need to wait for the project. Giving a closed group exclusive access would alienate (and probably annoy) anyone who'd already been working on the 2.0 code, and would result in the same community building problems we've been experiencing with version 1, once the end product was handed over.

Not too many developers. 3-7 developers would be good, so decisions can be made quickly by vote – e.g. 24/48 hour turnaround to do that (backed up by Scott)

(post-meeting addendum) I'm actually thinking that it would be good to have a periodically elected group of 3 (or 5) architect or senior developers to whom decisions are passed for arbitration if the entire development community (which I don't think should be limited in number) can't agree quickly. For people familiar with python, I'm thinking that this "architecture senate" would play the role Guido Van Rossum plays for python - making clearly backed up decisions made in good time that the community is by and large content with.

Ralph – developers' opinions should count more than 'theoreticians'.

Rob – Architect has to be developer – developers not necessarily have to be architects

Richard R: Mentions prototype-based approach

Jim: Should prototype only selected pieces, not 100s of prototypes testing every direction – need to be focussed

Scott: Question about 1.x and 2.0 – do we branch them? Try and feed/migrate components from 1.x to 2.0?

Richard R: We need to manage them in parallel, important that 1.x has carries on to maintain momentum

Ralph: manage in parallel – customisations to 1.x not assumed fit to 2.0 – we should be bunt to developers about this, rather than make promises that can't be fulfilled

Rob: Likely to end up similar to Apache HTTPD 1.3 & 2.0 threads – 2.0 starts off with less functionality, but catches up over time. 1.3 is alive and continue to be maintained, but increasingly, new features tend to be added to 2.0 due to improved architecture

Jim (post-meeting addendum): It's worth pointing out the similarities and differences here. HTTPD was modular both before and after the major version change, I believe it was the new threading models in 2.0 (and the variety that had to be supported) that gave module developers a headache. In DSpace the nature of extensions will (hopefully) change completely, but the code required for any particular extension should be better defined and simpler to understand.

Basically I agree with Ralph. As a maintainer of a customised version of DSpace myself I'd regard the effort of porting my mods into plugins for 2.0 easily worth it, if it meant I didn't have to maintain a whole parallel branch anymore.

Ralph: Committers should flesh out an architecture by prototyping the body of it with a particular 'seed' application, e.g. e-publishing

Jim: building around storage layer prototype would be a good 'seed'

Rob: Consensus seems to be that there needs to be a longer on-ramp/overlap period of architectural discussion and prototypes and development project

Richard J: question of how easy to obtain 4 or 5 dedicated developers?

Ralph: How many open source repo efforts can this community sustain? Have we talked to the FEDORA people? Shouldn't we make a common cause?

Richard R: FEDORA & DSpace originally addressed different niches – but DSpace+2.0 means this is less the case now

Ralph: There is fair amount of common interest: for example, DSpace could provide the UI, FEDORA the rich dissemination procedure

Agreement that more communication and cooperation between projects should occur