Managing Release and Integration Cycles

The topics on this wiki page will be discussed in a Special Topic Meeting on July 28, 2010 in the #duraspace IRC channel at 20:00 UTC. Additional information in announcement to dspace-devel listserv.

Utilizing Instability and Breakage to Promote DSpace Evolution, Code Review and Developer Participation

Proposal for less constraints on Trunk:

To Give DSpace Developers opportunities to work on the trunk without restrictions that everything going into it be completely perfected gives the developer community and the codebase opportunities to evolve. We recognize that sometime disturbance and breakage has a stimulating effect on developer participation.

Therefore it is proposed that the DSpace Guidelines for Committing to trunk allow for period of instability and stability with clear deadlines as to allow for the introduction and testing of new features which have been decided to be included into the next release by the community.

Committer Rights and Google Summer of Code Participation

Background Discussion of GSoC Commit Rights:

This proposal originally arose out of the Google Summer of Code project with the need to actually have GSoC students help with the code merge of their projects into Trunk. But, as students are not "Commiters", they currently are limited in receiving rights to Trunk (under our current policies). It is recommended that we hold a Special Topic meeting next week (July 28, 2010), and try and determine what to do in this scenario. Obviously, this proposal then leads to other questions of how we may view committer rights.

Discussion of how best to begin preparing/merging "ready" GSoC code in preparation for DSpace 1.7 began in the Developer Meeting on July 21, 2010.

Options for merging of GSoC code into Trunk for 1.7:

Four options were discussed (there may be more options available):

1. Require students to create JIRA issues for their projects? (Preferable to have several tickets for a single project, breaking it into components in varying states of readiness)
   * We'd probably want to encourage this, no matter which merging option we choose below [Tim Donohue]
2. Allow "ready" projects (or parts of projects) to be merged & committed immediately to Trunk (by the students themselves)
3. Allow "ready" projects (or parts of projects) to be merged & committed immediately to a Branch (by the students themselves). After a quick review process, this Branch would be merged back into Trunk for DSpace 1.7 release.
4. Keep the GSoC projects where they are (in SVN). Schedule them for review, and have a Commiter (GSoC mentor?) merge into Trunk later on.
   * This option is how we've done things in the past. It is worth noting that, to date, no GSoC project code has ever been accepted into Trunk, as oftentimes they are "abandoned" or left in an uncertain state after the students complete their work.

Key Questions:

1. How liberal or conservative do we want to be with allowing GSoC students to commit/merge "ready" code in preparation for DSpace 1.7? This includes:
   a. How liberal/conservative do we want to be about giving students temporary commit rights to Trunk? Or, would we rather they merge their code together elsewhere (e.g. a common branch based on Trunk)?
   b. How liberal/conservative do we want to be about allowing for temporary "breakage" of trunk (which could happen as several projects attempt to merge code)?
2. How much extra reviewing do we want of GSoC projects whose Mentors feel the code is "ready" for broader distribution/release in DSpace 1.7? If extra reviewing is warranted, how do we want to ensure this review is done in a timely manner (i.e. in time for DSpace 1.7, as necessary)?

(Feel free to add your own comments to these questions directly on this wiki page.)

Issues In Relation to Asynchronous Release and Modularity

Robin Taylor says

Having completely forgotten about yesterdays developers meeting (doh !) I just read the log and was interested in the chat about committer rights etc. I don't want to preempt any further discussion but just wanted to say that it might be logical to have the discussion about modularisation and asynchronous releases before the discussion on committing to trunk. The envisaged shape of the svn repo might affect how we view committer rights eg if Sword lived in its own module with its own release cycle then people might be less concerned about an individual with an interest in Sword having commit rights just to that module.

Original Dialog In IRC Meeting

(Also see the meeting notes/transcript at DevMtg 2010-07-21)

[20:03] <tdonohue> mdiggory: are there any GSoC upcoming meetings to know about (I know this was mentioned as still being worked on at/before
Monday, August 9

Suggested 'Pencils Down'

on the other hand, i've argued that broken trunks aren't the end of the world

I challenge that this community needs to change its perception of trunk.

commit to a branch first may be less controversial

mdiggory_ i understand your challenge (and honestly, I agree to a point) -- it's just something that needs to be voted on

but and do so respectfully, I know why the first blush is to be conservative.

ok, I was going to say that the REST work, if we are truly targeting 1.

We talked about temporary commit rights etc at the or10 meeting...

DSpace Summer of Code

Monday, August 23

Final Results Announced

hi kshepherd -- still talking GSoC right now

that's great news all around!

I see Unit testing as a good first candidate

(yea -- it's better to split up into manageable "chunks"

but simply getting everyone used to creating tickets around the GSOC projects is the first challenge

and if building is harder to "approve" a JIRA issue if it's really just a placeholder for the whole project -- i imagine for some projects only "part"

may be trunk-ready, and the rest may need a bit more work

I agree with that. That's what I was trying to get at by saying that a "JIRA placeholder" is less useful than splitting up the project and adding separate (but related) JIRA issues into each of the appropriate modules in JIRA

might even want several tickets for a single project if it usefully breaks down into components in varying states of readiness.

(FYI for all -- this schedule is also embedded on our GSoC wiki page: DSpace Summer of Code)

I think a goal with the REST work is that it support legacy DSpace at the same time as new services portings... IMO it should be pluggable to that degree and I know that Aaron would probably recommend such a target.

I agree that not all the projects will be good candidates for trunk, but ticketing for other projects needs to target appropriate modules projects

so for storage and semantic web projects we have the "DSpace Services" JIRA project that might be a good home

midgigory -- yes, I agree with that. That's what I was trying to get at by saying that a "JIRA placeholder" is less useful than splitting up the project and adding separate (but related) JIRA issues into each of the appropriate modules in JIRA

mdiggory_ Might even want several tickets for a single project if it usefully breaks down into components in varying states of readiness.

mhwood -- in my opinion, several tickets is much better than one ticket (as most of these projects touch several parts of the codebase)

mdiggory_ ok, I was going to say that the REST work, if we are truly targeting 1.

Time to go fishing in syslog.

Commit to a branch and then merge slightly later?

But, simply getting everyone used to creating tickets around the GSOC projects is the first challenge

mhwood: unless there is a review between the commit to the branch and the merge... I don't see usefullness in that strategy.

All students did receive passing grades for the midterm.

Friday, August 20

Final Evaluation Deadline/Final Payment Issued

I think a goal with the REST work is that it support legacy DSpace at the same time as new services portings... IMO it should be pluggable to that degree and I know that Aaron would probably recommend such a target.

I think there is a point where some destabilization of trunk needs to happen to get the work in... and we need to agree on that period as much as on the feature freeze dates etc

I agree, that not all the projects will be good candidates for trunk, but ticketing for other projects needs to target appropriate modules projects

so for storage and semantic web projects we have the "DSpace Services" JIRA project that might be a good home

midgigory -- yes, I agree with that. That's what I was trying to get at by saying that a "JIRA placeholder" is less useful than splitting up the project and adding separate (but related) JIRA issues into each of the appropriate modules in JIRA

mdiggory_ Might even want several tickets for a single project if it usefully breaks down into components in varying states of readiness.

mhwood -- in my opinion, several tickets is much better than one ticket (as most of these projects touch several parts of the codebase)

mdiggory_ ok, I was going to say that the REST work, if we are truly targeting 1.

Time to go fishing in syslog.

Commit to a branch and then merge slightly later?

But, simply getting everyone used to creating tickets around the GSOC projects is the first challenge

mhwood: unless there is a review between the commit to the branch and the merge... I don't see usefullness in that strategy.

All students did receive passing grades for the midterm.

mhwood> Commit to a branch and then merge slightly later?

mhwood> Time to go fishing in syslog.

Commit to a branch and then merge slightly later?

But, simply getting everyone used to creating tickets around the GSOC projects is the first challenge
the longer they do so, the lower their probability for inclusion

Agreed: *prompt* review and decisions.

So where should we be looking? The SCM tree seems to have grown like kudzu lately.

and I agree with kshepherd, breakage on the trunk is actually a good thing

is the disturbance that promotes change/improvement

Agreed -- I think the goal would be that the branch is very temporary -- a place for immediate review and then hopefully move to trunk before 1.7

and I just think thats a not such a good idea, this is really what a trunk is for

So where should we be looking? The SCM tree seems to have grown like kudzu lately.

and I agree with kshepherd, breakage on the trunk is actually a good thing

is the disturbance that promotes change/improvement

yeah we do need to get less precious about trunk, i think ;)

that is why we are here as mentors, to assist them in learning such things if they do not

ie mentoring = helping

not mentoring = judging

depends on the way you want to view it - yes, we could get less precious and more experimental with trunk... but that would mean

Yes, be as experimental as we want, so long as there is an island of relative stability "somewhere".

or we work in phases/periods of stability /instability

ie... controlled chaos

so, i'll also note here that "breaking trunk" currently is against our recently approved "Guidelines for Committing" -- Guidelines for Committing (obviously, we can change these guidelines as we see fit, but we need to have larger discussion)

So, table for special topic meeting that needs to happen asap?

more developers that have an interest in such a policy have time to respond

yes, we could even schedule mtg for next week -- but, we need to also send an email proposing these changes (for those unable to attend)

I volunteer kshepherd ;-)

will you take lead on proposing this to dspace-devel today/tomorrow? We can put on agenda for next week's mtg.

hmm? for what?

you are the one who brought this all up, mdiggory_ ;)

" kshepherd runs svn delete on trunk and runs away cackling

haha ;) ;)

ok ok ok...

yes -- and explain the background as to "why you propose this" -- which is that the proposal is that the GSoC students commit ready code to trunk

those are two separate issues, I know -- but we might be able to cover both in one special topic mtg

my "break trunk" opinions have mostly been in the context of the way we tend to leave bugfixes going stale in JIRA instead of just committing... this is a slightly different context so i'd like to read/think more

ok -- anything else immediately on GSoC? I notice we only have 20mins left (not that there's much more substantial on the agenda though)

that is a bit different. I think many of us tend to want more comment than we give, so changes rot....

kshepherd nods

but I think it is related

no better way to encourage peer review than accidentally breaking trunk, though ;)

because it has to do with getting things into trunk, then hardening them when errors arise

so like mdiggory_ says, it can actually force us into action

true -- but we also don't want to harm productivity of our widely distributed group by having a broken trunk for too long ;)

rather than waiting long periods of time and only exceppting the most perfect merges

and will get us used to using atlassian code review tools more

which raises the bar for contribution

Do those tools work now? Must go look again.

heh true, of course.. be real in mind that i think 95% of commits won't actually result in a broken trunk ;)

yeea agreed ;)

mhwood: heh, i hope so! i think there is a test review in crucible

yes, crucible is still working