20110215 Developer Integration Meeting Notes

Agenda

10-11:30 Plan structure of integrated code base

1. What structure do we want for the source tree and what are the distinct build targets
2. What becomes a library?
3. Framework and patterns for unit testing

11:30-12:00 Plan Subversion migration – moving code into a single place

1. How to maintain continuous integration (Hudson/Jenkins)
2. Any impact of project structure on Selenium testing
3. Who does what in what sequence for the merge

1:30-3 Collaboration

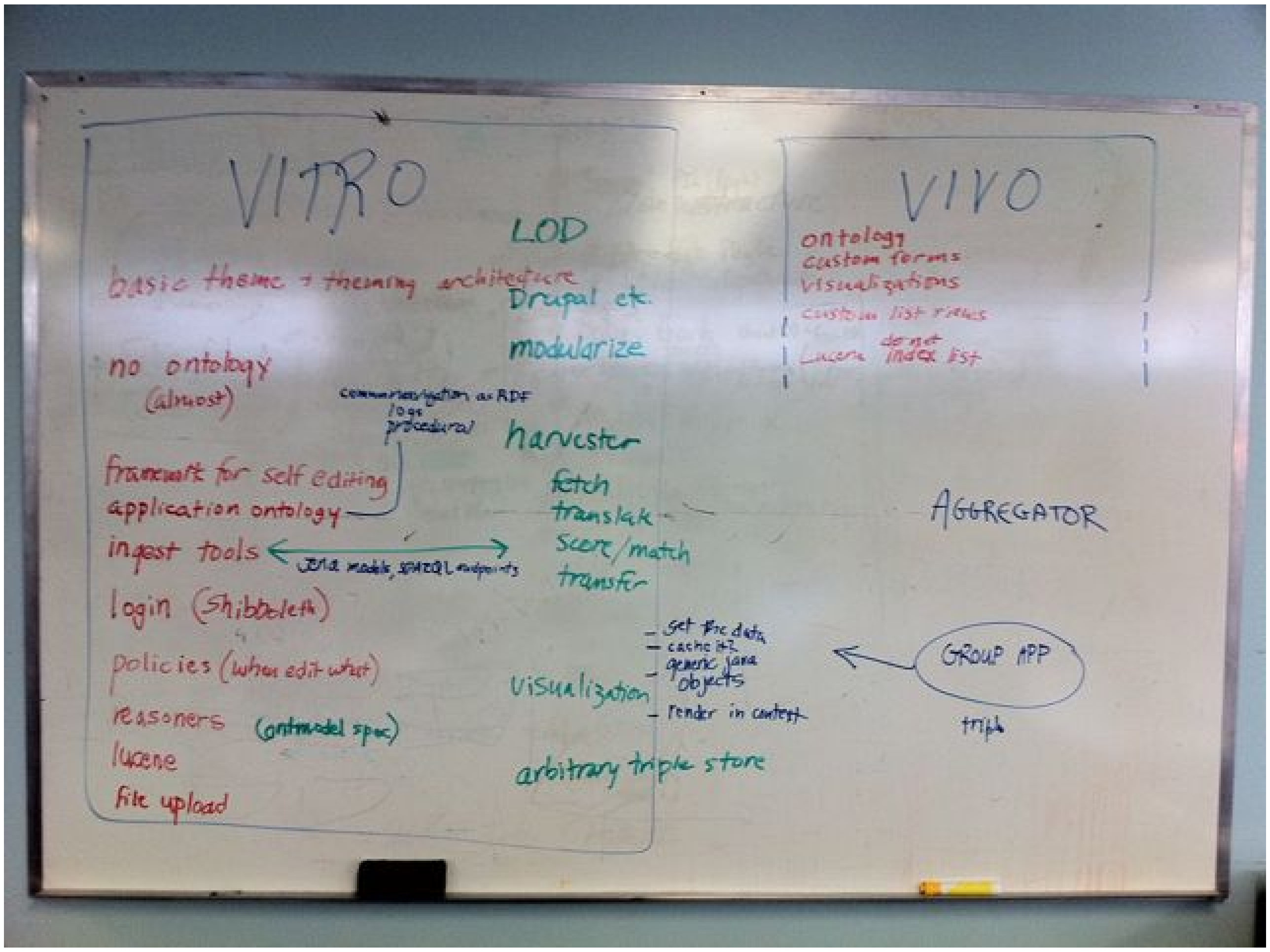
1. What is effective collaboration for us now and what are the barriers to us collaborating better
2. managing the lines of communication
3. assignments, division of labor #
4. Managing the code base

- Reviewing submitted code – who has what responsibility
- Deciding that we will have a standard
    - Standards for code structure – packages
    - Standards for coding style

1. How to submit a patch
2. How do I become a VIVO developer?

Plan structure of integrated code base

VIVO has two distinct subversion repositories – Vitro for the bulk of the generic functioning of the application, and VIVO for specific modifications associated with the VIVO ontology.

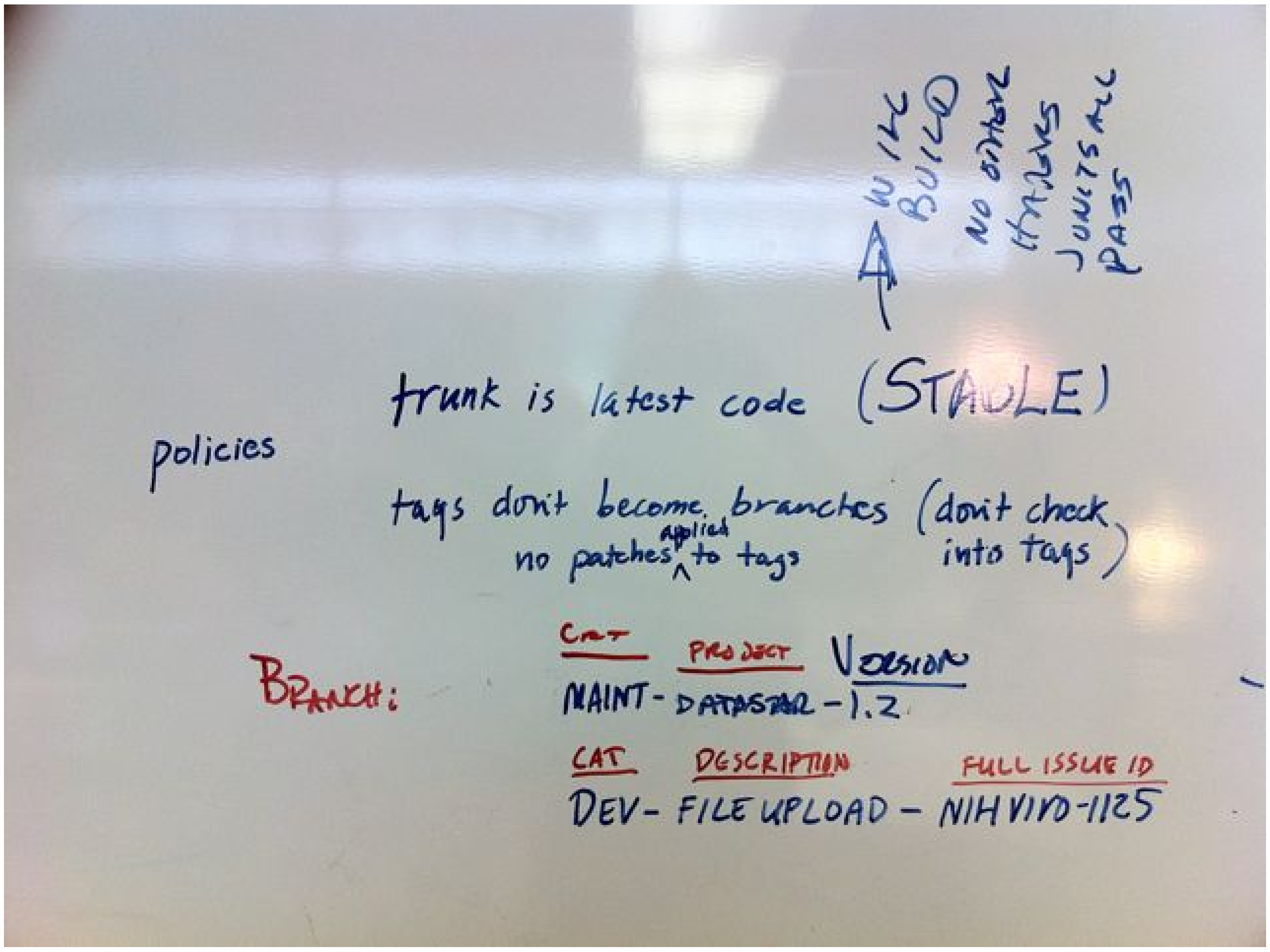


Discussion of difference between VITRO and VIVO

VITRO	VIVO

- basic theme + themeing architecture
  - connectors
    - wordpress
    - drupal
    - R
- no ontology (almost)
  - application ontology
- framework for self editing
- ingest tools
- login(shibboleth)
- policies(when edit what)
- reasoners (ontmodel spec)
- lucene
- file upload |
- ontology
- custom forms
- visualizations
- custom list views
- lucene (do not) index list |

Branch structure


WILL BUILD
NO OTHER [illegible]
JUNITS ALL PASS
policies
trunk is latest code (STABLE)
tags don't become branches (don't check into tags)
no patches applied to tags
BRANCH:
CAT
PROJECT
VERSION
MAINT-DATASTAR-1.2
CAT
DESCRIPTION
FULL ISSUE ID
DEV-FILEUPLOAD-NIHVIVO-1125

Discussion of the SVN

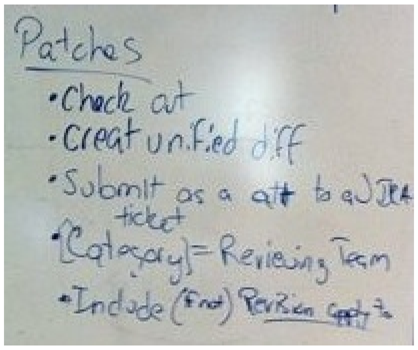
- trunk
  - latest stable code
  - code that is stable to the point other developers won't hate you
  - development occurs in trunk
- Tags
  - moments in time
  - do not become branches
  - no patches applied to tags
- Branches
  - development (that might annoy other developers)
  - release(moving)
  - very unstable
  - Naming Convention
    - development branch (dev-Description-JIRAProject-Issue#)
    - maintenance (MAINT-Project-Version)

Submitting code changes

General discussion was that we should accept code changes from the mailing list. Right now we can use a basic policy about what to do about accepting code but if we start to get a lot of submissions we will have to review this policy.

(based on XBMC's How to Submit a Patch)

1. create unified diff
2. submit as attachments to JIRA tickets.
3. category = reviewing team
4. include revison to apply to (if needed)
5. license requirements



Patches

- Check out
- Creat unified diff
- Submit as a att to a JIRA ticket
- [Category] = Reviewing Team
- Include (if not) Revision apply to

Todo: Jim Blake: create sourceforge wiki page on submitting code changes.
http://issues.library.cornell.edu/browse/NIHVIVO-2142

Code style (3:20 pm - 3:37pm)

Can we get Eclipse to help us?

How will we agree on what the style standards are?

When will we discus problems that people have with styles.

Todo: Micah will get us the xml check style config

Decision: Project will use the check style plug-in with a unified configuration. We will have to come up with a way to agree on the style if conflicts are a problem. We can talk about changes on 3pm Thursday calls.

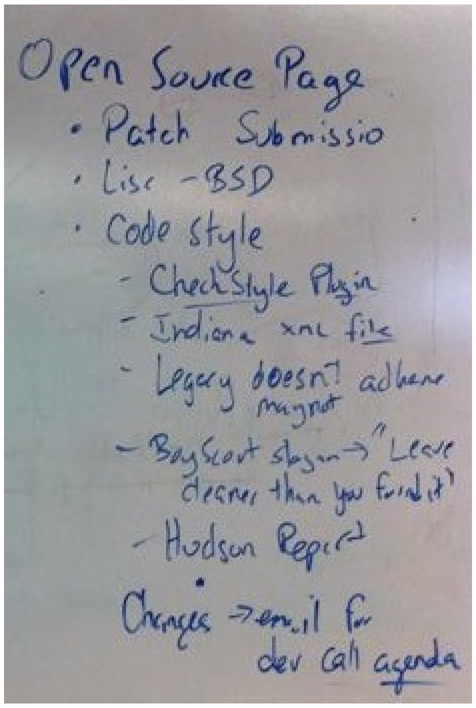
How to become a developer? (3:37pm- 3:48pm)

A developer who is already a committer can email the dev list about this person and ask to have them added to the agenda of a 3pm developers call? Have a lead add an item to the agenda to get someone approved as a committer? We could have them nominated on a meeting one week and then approve then next week. (Chris B, Brian C and Steven W like this).

What should we do about discussing someone who wants to become a committer? Will people be reluctant to criticize someone in public? They could just email Jon CR or someone with the problems in private?

What do we do about developers who are causing problems? Does someone who is having problems just ask someone like Jon CR to look into this? Should a dev lead just yank their SVN access?

todo: Steven Williams write a wiki page on becoming a developer



Open Source Page

- Patch Submissio
- Lisc - BSD
- Code style
  - CheckStyle Plugin
  - Indiana xml file
  - Legacy doesn't adhere magnet
  - Boy Scout slogan → "Leave cleaner than you found it"
  - Hudson Report
- Changes → email for dev call agenda

break (3:48 - 4:00)

4:00 - 4:30 CU people worked on release. Chris B, Brian C and Steven W. talked a little about what to to with openness. Chris B said that we can outline some documents and get some of the other people to flesh these out. Talk about "getting started guides" for projects like Debian, Ubuntu and Ruby.

Unstructured discussion (4:30 pm - 5:00pm)

What would need to change for doing release out of Sourceforge?

StevenW Likes regular releases for predictability. FL has been doing half time on development and half on testing. Are we going to need a unified cycle across the project?

Chris B. What is the process to decide on what to work on in a release?

JonCR - After the project, what will we do to decide what features to work on in a release?

BrianL - Would like to spend time working on providing data in formats like bibo (http://bibontology.org).

Nick - Would like to do more user testing to design new features.

BrianC - My priority is the national network.

ChrisB - Concerned that VIVO is seen as just a phone book. Also interested in usability testing.

StevenW - Competitive landscape review. Digital Vita has a collaborator feature.

ChrisB - Vanderbilt applied to be a national CTSA coordinating center. As a coordinator they will need to find co-PIs who work at other CTSAs to collaborate with. How can we do this in VIVO? How can I answer the questions: "Who can I work on this grant with?" "Who can introduce me to this person that I'd like to work with?" Can we write plug-ins to write mashups with Linkedin?

StevenW - Tomorrow we will decide what process to use to choose features for upcoming releases?

JonCR - Would like to be able do national network tasks (starting with search) in a local VIVO.

We looked at the Temporal visualization and people liked them. They would like to have the visualization on the profile page of organizations.