2008-12-04 Developers Conference Call

Date

4 December 2008

In Attendance

- Hamish Fraser
- Darius Jazayeri
- Tammy Dugan
- Mike Seaton
- Justin Miranda
- Ben Wolfe
- Burke Mamlin
- Paul Biondich
- Brian McKown
- Saptarshi Purkayastha
- Jeff Rafter

Agenda

- Implementation Builds
- Location hierarchy / tags
- Trunk version – should it be 1.4-alpha?
- Code Review
- Complex Obs

Minutes

- Implementation Builds
  - Everyone trying to run an implementation not off of trunk. Subversion not necessarily easy.
  - Git has benefits, but not enough to merit switching.
  - Brian
    - Check out release branch into separate repository (1.3.x) and apply implementation specific patches.
    - Continue to patch further changes until decide to switch to the next minor revision (1.4.x)
    - Then check out 1.4.x and re-patch implementation specific changes.
  - Tammy
    - Cannot keep pulling latest from trunk because we need a semi stable release.
    - We can pick a stable maintenance tag - but Arden is changed frequently.
    - If we commit to tag it doesn’t get populated to trunk.
    - Check out 1.3.x, make Arden changes, merge to trunk later?
  - If you branch off of 1.4.x, you don’t want to merge back into 1.4.x but to trunk.
  - Is an implementation branch ever merged back into core?
  - Lots and lots and lots of microphone static.
  - Feature branch vs. Implementation branch.
    - Implementation branch is not designed to ever be merged into core.
    - Feature branch is designed to be merged into core.
  - Branches should be short-lived, off of trunk, aimed at a release, and addressing an issue on the road map.
  - To be in the repository it should be a feature branch.
  - The alternative is that people download openmrs code and put it in their own repo. The changes they make are completely separate from us and never seen. They push patches back to us at their own discretion.
  - If we give those people a place in svn.openmrs, then we see what is going in there and can potentially ask for patches to come back into trunk.
  - Make a distribution (or build) folder in the OpenMRS repository?
  - Implementation branches are not allowed. OpenMRS repository should not be for creating a separate branch to do implementation specific development.
  - Build branches to be used to work on customized features that are intended to go into core.
  - TODID: Create place in repository for custom builds.
- Code Review
  - Google doc contains the review notes from the review call. Sometimes have trac tickets.
  - Code review open for suggestions for structure.
  - Should have a code review tool.
    - Confluence and crowd.
  - Mike 2 points
    1. When finished with piece in code review, there is not place to check it.
    2. Assigned/resolved is not sufficient. Should use Trac ticket or similar.
  - Justin
    - New tickets should be created with “required for merge””is not a must to be merged”
• Google Doc was/is intended to be to take notes during the review
• Is good to make tickets from it.
• May be good to record the calls.
• Don't want extremely high standards to be prohibitive of new features being able to be added.
  • Can prioritize items in the code review.
• Long discussion on coding standards and commenting code.
  • Every class should have a class level comment in order to be committed to trunk.
  • If what the code is doing is not obvious, it should be commented.
  • Problem of resources vs. standards.
• Weekly e-mail to the developers mailing list with high/low priority tickets where the community can help.