Documentation: Write the Docs

Date
23 Sep 2015

Attendees
- Kieran Etienne - University of Washington in Saint Louis
- Carla M. Galarza - Dartmouth
- Heidi Waterhouse - Documentation Specialist
- Mark Bussey - Digital Curation Experts
- Michael Bond - West Virginia University
- Anjanette Young - University of Washington

Goals
- Address issues with regard to documentation within the community.
- Decide on some actionable items to take to the larger community which may help to alleviate issues surrounding documentation.

Summary
- projecthydra.org vs. confluence vs. github
- documentation being out of date
- paths to appropriate documentation channels for specific audiences
- docs can save incredible amounts of time
- treat documentation like code
- exploration of doing pull requests for documentation
- policy: gem releases include a snapshot of github wiki
- "good for newbies" tag is helpful
- mark can propose on hydra call what best practices we come up with
- some documentation lives in gem's readme.txt/md
- is readme.txt/md production only? does release tagged code come with production documentation?
- can we keep docs near code?
- if you don't know what you're looking for, doc/answer could be in one of many different areas.
- silversurfer could take a long time to search docs, and only if you know exactly what you're looking for.
- can documentation be compiled up from code/readme into a higher level? can it be automated?
- cannot issue pull request to change documentation code. answer is probably to edit doc to include the question / issue.
- can questions / issues for documentation be tagged?
- what is the best way to bring issues on documentation to the community? email? tagging in github repo documentation?
- no feasible way to tag documentation inline (color code, etc) with documentation questions / issues.
- dce borrows "Dive Into Hydra" documentation format from "Rails Bridge"
- what can we report back? lessons from Heidi?
- coders use testing frameworks to automate testing. documentation maintenance is huge. how can we automate portions of documentation writing? "Review" is a bad word. "Testing" is a good word for evaluation documentation quality.
- is there a way to time-expire things on Confluence?
- is there a way in github wiki to find all pages that link to a page?
- what are the bit-sized or manageable pieces that we can get tangible solutions for?
- kill an out-of-date page and receive a bounty?
- how to invite newbies to knowledge based projects -> Sumana Hariharan
- slack vs. IRC