Portland State UNIVERSITY

Bruce Marron <bmarron@pdx.edu>

Re: LANDIS Project Update (Marron)

1 message

Robert Scheller <rmschell@pdx.edu> Mon, Sep 12, 2016 at 8:35 AM
To: Bruce Marron <bmarron@pdx.edu>

Bruce,

This is fantastic! This all makes perfect sense.

Can I share the QuikStart on the user BB? Actually, better if you did.

I like the testing plan. There are two levels of testing (in sequence):
1. Against the test data sets. These ensure that the revisions run against a sample data set, to completion, and produces reasonable results. Typically a log file and/or maps that I can examine and see whether they pass the sniff test.
2. Beta testing against actual data sets. We typically recruit users to test fixes against their data. This is particularly true when a particular user submitted the original issue. For example, Eric wants to see a harvest bug fixed and he will be well prepared to test it against his data. This takes longer, of course. Other folks can be recruited for other issues. And we will likely lump some issues, e.g., core tweaks, together for this final level of testing.

If it passes both of the above, we release.

Cheers,

R

On Sun, Sep 11, 2016 at 9:39 PM, Bruce Marron <bmarron@pdx.edu> wrote:

> Hi Rob,
>
> Quite a bit here so I'm going to switch to itemized mode.
>
> 1. Attached is the first round of project management tools: a .zip file containing a sample of the proposed project tracking system (using TaskJuggler), a proposed project timesheet, a sketch of the expected GitHub workflow, and a LANDIS-II QuikStart for newbies (myself included).
>
> 2. Unzip the .zip, open the TJ_SampleProjectTracking_20160911 folder, and double-click on the LANDIS_Project_Status_Sample.html file. You should see a Task Status report, a Expected Milestones report, and an Accounting Status report. The Task Status tracks weekly status of project events such as task assignments, percentage of tasks completed, trainings, and mtgs for myself and each of three programmers. Actual hrs worked per task are entered from timesheets. Programming tasks (PROJECT GitHub ISSUE SETS) are keyed to LANDIS Foundation repos. So, for example, during the week beginning 10-17-2016, Programmer 1 and Programmer 2 worked on task1 (a GitHub Issue) while Programmer 3 worked on task2 (another GitHub Issue). Programmer 2 has the lead for task1 and Programmer 3 has the lead for task3. During the following week, Programmer 1 finished their portion of task1 and moved on to task2 (taking the lead for this new task), Programmer 2 finished task1 and

went on to start task4 while Programmer 3 continued working on task3 having it nearly completed by the end of the week. Milestones are self-explanatory. Accounting shows aggregated charges per individual with new charges added to the week following the actual work. We can go through this more in detail if you'd like. The TaskJuggler script is provided in .pdf and .tjp files.

3. The timesheet is the primary document for updating the project tracking system. Tasks are coded for mngmnt:
MT1 = Management Structure
MT2 = Recruitment
MT3 = Trainings
.....

and for programming:
GHB1 = Extension-NECN-Sucession
GHB2 = Library-Climate
GHB3 = Extensions-Succession
GHB5 = Tool-L2-Site
GHB6 = Extension-Biomass_Extension
GHB11 = Core-Model
GHB27 = Extension-Biomass_Harvest
GHB30 = Tools
GHB36 = LANDVIZ-Development

Subtasks (I'm calling them tickets for TaskJuggler but they are in fact the individual GitHub Issues) are self-selected by the programmers, as discussed above. I like real timesheets; I'm old-school.

4. The anticipated GitHub workflow is sketched out in, 'Workflow_sketch_20160911.pdf.' GitHub is amazing. Pretty sure the outlined sequence will give us what we want (ie excellent version control, excellent change history, and excellent firewalls). The sketch tracks the life of a single Issue. Note how by having me (bmarron18) fork each of the to-be-updated LANDIS Foundation repos, we have the programmers working from my fork until all changes are made and checked. Only then will I send you a Pull Request. At some point, we should do a test run on dummy (or real?) updates to make sure the system functions correctly.

5. The LANDIS QuikStart hopefully gives the programmers a rapid intro to running LANDIS. (Writing this was super helpful for me too since I haven't done a LANDIS run since the training last year!)

6. I split model testing/verification into two, QA/QC levels. The first includes a peer-review of the code and the second will require actual model testing with defined metrics for acceptability. Maybe you already have LANDIS scenario sets that can be used as 'model testers'? Or maybe it's case-by-case? You can let me know how you would like to proceed.

7. Early this coming week I'll have a draft job description that we can hammer out so that I can start contacting folks. Of course, there is the university job posting service (http://www.pdx.edu/careers/posting-jobs-internships) if you would like to post there.

> 8. I went ahead and added one of the core Issues per your comments on the USFS request list. If that's what you had in mind I'll go ahead and add the rest.
>
> 9. I'll begin organizing materials for the introductory training session(s) later this week.
>
> Give me a holler if you'd like to go over any of this or you can just send me corrections.
>
> Thanks Rob! It's sure great to be volunteering.
>
> Bruce
>
> Bruce Marron, MS
> NSF IGERT Fellow
> Earth, Environment, and Society PhD Program
> School of the Environment
> Portland State University

--

Robert M. Scheller, PhD
Associate Professor
Dept. Environmental Sciences and Mgmt
Portland State University

Office: Science Research & Teaching Center, Room B1-04C
Phone: 503-725-2497
https://sites.google.com/a/pdx.edu/dynamic-ecosystems-landscape-lab/

Mailing Address:
Dept. Environmental Sciences and Mgmt
SRTC, Room 218
1719 SW 10th Ave.
Portland, OR 97201