Subject: Re: [LANDIS-II Developers] GitHub Configuration
From: James Domingo <jdomingo@green-code.com>
Date: 7/27/15, 6:25 AM
To: landis-ii-developers@googlegroups.com

Technical Issues

On July 24, Robert Scheller <rmscheller@gmail.com> wrote:

> The GitHub repository was configured as it was for expediency; ease-of-transition; and variable activity of different sub-directories.

Unfortunately, those reasons don't justify multiple projects in a single Git repository.

> Expediency: Self explanatory given the impending Google Code shut down.

This isn't a factor because even now, there's plenty of time to execute a collective plan to migrate each of projects from the Subversion repository into its own Git repository. Using the GitHub import link that I shared previously, any developer -- even those without write-access to Google Code -- can create a single-project Git repository in their GitHub account.

So the work can easily be divided among L-II developers working on different projects -- put the project list into a Google spreadsheet, and have developers sign up for the ones they'll migrate. Once all the projects are assigned, the developers can simultaneously start their imports.

It takes only a minute to launch the import process for a project. Once a repository is ready in their GitHub account, the developer can begin using it right away for development. She doesn't have to wait until the repository is transferred to the Foundation to continue working on her code. The repositories can be transferred as time permits, later on when it's convenient for the Foundation admins.

> Transition: We wanted to centralize some of the wikis that are very general (say, for all libraries).

True, just because each Git repository can have its own wiki, doesn't mean they should. For some library projects, a simple README.md file (which replaces its individual page in the Google Code wiki) may be sufficient. Some more complex libraries like Succession may warrant having their own separate wiki.

But common information that applies to multiple components (like library packaging) does not justify putting those components into a single Git repository. A more appropriate home for this shared developer information is the SDK project. After all, every developer, whether working on a library or an extension, needs to visit that project to obtain a release of the SDK. The wiki of the SDK GitHub project is the logical choice to put information from the Google Code wiki that applies to multiple components.

> Variable activity: Some extensions are very static and it was convenient to store them in one place.

No extension or library is permanently stable. Even a fairly mature component will eventually need modification. It doesn't even have to be a scientific change; it could be purely technical (for example, upgrading the target framework from NET 3.5 to 4.5). But the slow pace of a component's evolution does not warrant making all future development more difficult (as in, beyond inconvenient) by forcing it to share branches and tag with other components.

In short, none of these reasons -- alone or collectively -- justify shoving a multi-project square Subversion peg into a round Git hole. That migration approach will not allow developers to use Git branching and tagging properly.

Communication Issues

Although these serious technical issues still haven't been adequately addressed, this conversation has raised even more alarming issues about communication. I raised those technical concerns in 2 posts on another topic on this list. **But those 2 posts have been deleted.** For the record, here are links to PDFs of them (3rd post and 5th post in that topic).

This censorship is unjustified. Removing posts from a public mailing list is only justified in rare circumstances -- spam, security (e.g., inadvertent disclosure of user credentials), or abusive language. My posts clearly don't fall under any of these categories.

Why were they removed? Why didn't you post your response on that original topic, instead of creating this new topic? Their removal is a totally inappropriate way to deal with criticism. Their removal is an abuse of the trust that the developer community puts into those who manage the list.

I've been one of those managers, along with Rob (who is the list owner) since the list was created years ago. **But I no longer am.**
I just discovered last Friday (July 24) that **my administrative access to the list has removed without any notice or explanation.** Another unjustified action that flies in the

face of all the talk about transparency.

It'll be very interesting to see if this post meets the same fate as those earlier posts. Or if it will be met with the professionalism that this community forum deserves.

Jimm