Jenkins Hudson Reconciliation Requirements

Introduction

With Oracle proposing to donate Hudson to Eclipse, the question of what would be necessary for reconciliation between Jenkins and Hudson has come up. Many users are definitely interested in the possibility of reconciliation, so it seems worthwhile to address. This is not an easy question to answer - but hey, we're going to try to answer it, here! Please provide your suggestions for requirements below, and use the comments to, well, comment on other suggested requirements.

Note that this is not asking what would be required for Jenkins to join Hudson under that name, at Eclipse, etc. It's more general: what needs to be done, from either side, in order to actually get the groups working together in a viable way? If that turns out not to be possible, that's fine. The important thing is to give this a shot and be sure we understand the differences and whether they can be resolved.

Also, please leave your name/id with your suggested requirement. Thanks!

Proposed Requirements

- Kohsuke as lead/co-lead (abayer) community to vote on other co-leads (magnayn) (+1 jieryn) (+1 drulli) (+1 olamy) (+1 karianna) (+1 kutzi) (+1 dafreels) (+1 bap) (+1 domi) (+1 mfriedenhagen) (+1 pelegri)
- Development through github (can mirror back to other systems as neccessary) (magnayn) (+1 jieryn) (+1 abayer) (+1 olamy) (+1 kutzi) (+1 domi) (+1 mfriedenhagen) (+1 pelegri)
    - I think gerrit could be acceptable as well (could mirror in GitHub? Not sure of technical details there) (karianna) (+1 mole125) (-1 mfriedenhagen: IMO we do not need the complexity, github's pulls are just fine. Having a nonformal Gerrit workflow seems to be achievable.).
- MIT (or MIT-ish, e.g., ASL, BSD, EDL) license (magnayn) (+1 jieryn) (+1 bap) (+1 mfriedenhagen)
    - Is dual-licensing possible? (karianna)
- Retention of Jenkins' development style and process (liberal distribution of commit bits). Not using the eclipse process; encouragement of 3rd party contributions, wherever they come from. (magnayn) (+1 jieryn) (+1ish abayer) (+1 drulli) (+1 olamy) (+1 bap)(+1 kutzi) (+1 mfriedenhagen)
    - Is the Eclipse process so bad? Can the Jenkins community help streamline that process? (karianna)
        - Yes. It's very bad (for developers). Bad enough to end many contributions. (who's comment has this?) _(history says it was magnayn - abayer)
- Weekly release for Jenkins core (jieryn) (+1 abayer) (+1 olamy) (+1 dafreels) (+1 bap)
    - For early adopters, sure. But I can understand that enterprises want a slower more stable release cycle. Choice is good. (karianna)
    - I don't necessarily need a weekly cycle. IMO a bi- (or maybe even 3-) weekly cycle would be okay, too. Hotfixes can be released earlier, of course. The point is that I don't want a release cycle of > 1 month (kutzi)
    - I think the weekly release is not exclusive - I'm a big fan of the idea of the stable branch (say, every 6 weeks or so) as well as the rapid iteration weekly releases in parallel. Especially with the idea that plugin devs use stable branch releases as the parents for their own releases. (abayer)(+1 domi)(+1 mfriedenhagen)
        - That last part would suck if a plugin needed a change to the core to work, and the plugin author had to wait until that version of core or newer became the basis for stable branch before the plugin could be released. (dty)
        - Rapid-iteration release/hotfix: weekly. Stable release: Monthly or 3 to 4 weeks. 6 weeks is too long. (dorothyvaliga)
- All Maven artifacts pushed to central (jieryn) (+1 karianna) (+1 domi)
- Jenkins name and logo remain with Hudson being absorbed (dafreels)
- Neither Jenkins nor Hudson remains as a name but a new name is created for the merged community (to prevent any we won, you joined us ill feeing continuing (Jenson / Hudkins) (teilo) (+1 dorothyvaliga)
    - I think this could be a nice Olive branch if the main players feel strongly about it (karianna)
    - Bunch was a female butler in literature (see wikipedia), that would be nice and recognizes that we get a bunch done with this tool (dorothyvaliga)
- Clear and very open governance model with discussion in public not behind closed doors (mole125) (+1 abayer) (+1 bap) (+1 dafreels)(+1 domi) (+1 kutzi)(+1 mfriedenhagen)(+1 akostadinov)(+1 karianna)(+1 pelegri)
    - Some variation of meritocracy being part of the governance model (pelegri)
- PlugIns should not be part of the consolidation. They need very flexible development process and commit organization (pelegri).

- (insert requirements here)