Release Manager Checklist

Milestone Entry

- Clearly declare a milestone release mgr at beginning of the milestone.
- Post to the devlist the target delivery schedule of the milestone.   Get consensus from the community early.
- Nail the must-have function from the community that is required to be delivered in this milestone.
- Target (or reTarget) all of the Jira defects and new function that is required for the milestone. Move non-critical items to the next milestone early.

Milestone Middle

- Keep Jiras under control during the milestone.   Make sure new opened ones are targeted for the appropriate milestone, and the backlog is decreasing.
- Make sure the new function Jira are marked appropriately (since they will be used in the ReadMe file creation).
- Look for Jira's that have patches attached to them and get the code integrated early in the cycle.   Don't wait until the last minute.
- Make sure that you begin to obtain clean versions for all SNAPSHOTs in the build. This can sometimes be a lengthy process as dependent packages are sometimes not available.
- Double check that the TCK machines are ready for executing the TCK.   Lay out a plan for the distributed execution of the suite.
- Now is the time to ensure that all of the licenses are valid and replacements/remediation should be done.
- Declare a candidate build to the community.
- Make sure that you remind the community that all modules should have the appropriate header file with the appropriate copyright statement. http://www.apache.org/legal/src-headers.html#headers

Milestone Exit

- Declare a candidate build for the TCK testing to the community and work any issues that are voiced.
- Update the Readme file with the important New Function and Total defects.   Put the delivery date of the milestone in the top of the readme.
- Ensure that tooling is verified with the specific milestone candidate build (late changes can break tooling, don't ignore this step).
- Double check to make sure that the Graphical Installer works with the build candidate.
- Get some to verify that all of the samples pass without issues.
- Complete TCK testing for the build (this normally can take 2-3 weeks).
- Near the end of the TCK testing, create a new branch in the build tree for the milestone. Remind the community that only milestone defects should be integrated to that branch.
- Declare the TCK testing has been completed, and request for a vote from the community to make this build public.   Allow at least 3 days for the vote to complete.
- After 3 days, summarize the vote and declare the milestone build as the public milestone build.
- Make approved build available in source and binary form. Declare this to the community with excitement 🙂.
- Create Service Branch for the milestone or release (if appropriate).
- Post a congratulation note to the devlist thanking the entire team, and calling out any extraordinary efforts that were done to make it happen.