

Contents

1	important	1
----------	------------------	----------

2	codes	1
----------	--------------	----------

1 important

- codes should be applicable to other ecosystems, like Eclipse or Debian
- we should consider concerns/activities/challenges for the ecosystem as a whole, not for the release team's way of working
- things that are specific to OpenStack, will be examples/quotes for paper

2 codes

- release planning (dates and stuff)
 - initial planning of next release
 - last planing of current release
 - * post release checks
 - release processes with unmanaged delays
 - follow-up of planning
 - * communication to stakeholders
 - * (semi-)automatic validation of progress
- check conformity and inclusion in milestone/official release
- deliverable consolidation => central point for end users
 - release note consolidation
 - central repo/site for releases
- dependency/constraint management
 - ensuring backward compatibility
 - * determining (stable) versions to depend on
 - * testing compatibility

- freezing upcoming library release to allow concurrent development of dependent projects
 - * communication to stakeholders?
- managing automation/tool support
 - tool development
 - change of workflow
 - * communication to stakeholders => infra integration part of this?
 - * deprecate no longer used tools
- maintenance of stable releases
 - determining mandate/scope of maintenance team
 - involve trusted liaisons
 - tag release as end-of-life
- managing code review
 - priority code review
- managing release model/governance
 - changing/adopting model => review requests
 - validating the model in use => not following right timing/procedure for their model could have consequences
 - independent, delayed releases building on older release vs. simple, consistent releases across ecosystem
- workflow to signal that a project version is ready for milestone/release (tags)
 - communicating urgent release request (IRC)
- establishing and managing network of trusted liaisons across all ecosystem member projects => includes having clear responsibilities for everyone
- uniform branch management across projects (to put stable code, etc.)?