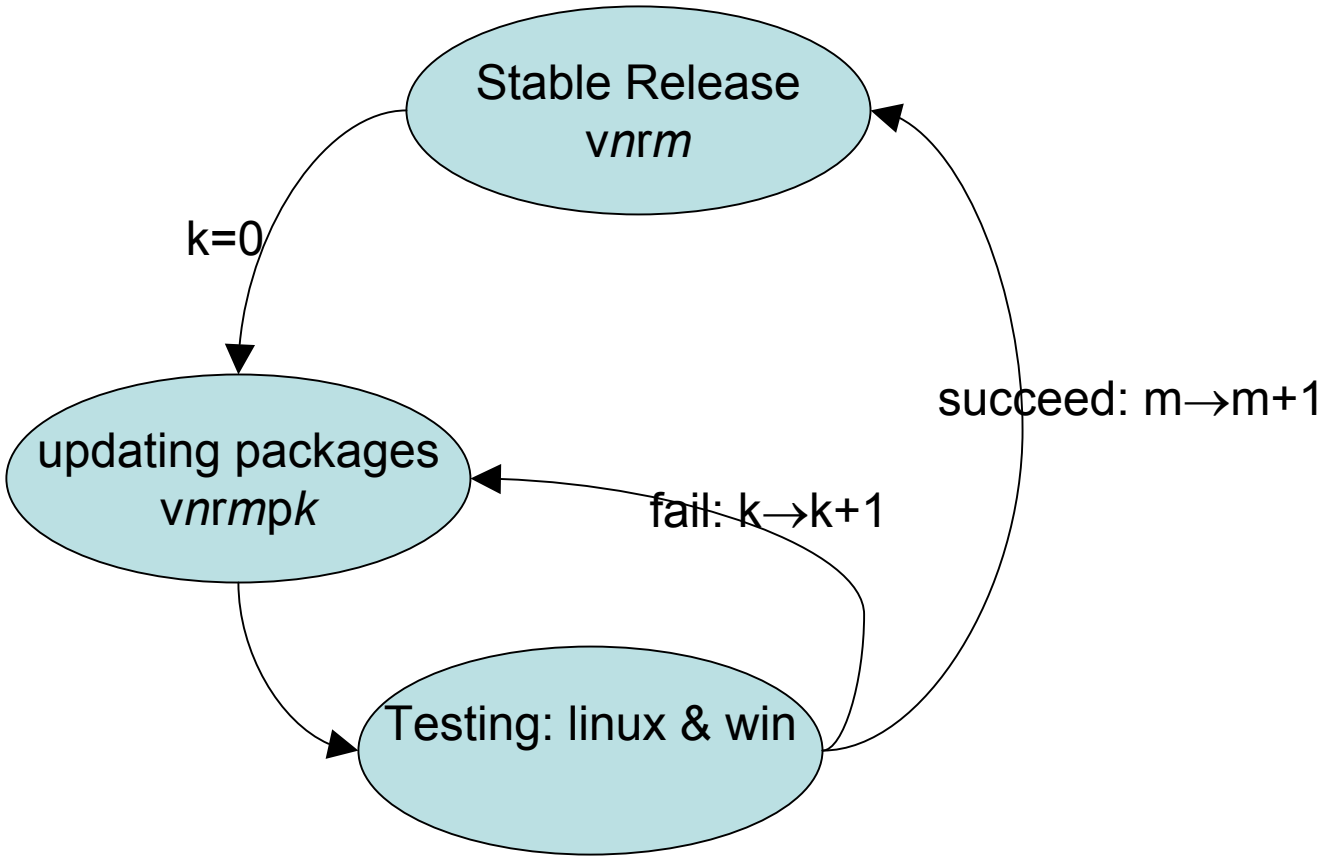
Proposal for a GLAST release system: state diagram



Details

- Release definition
  - contents of the tagged GlastRelease package, in usual form
    use *Package version*
  - for example, see the current GlastRelease package in our cvs
- Define new package set (to k=0)
  - From the *package* table in the Oracle tag database
  - Changes in response to authorized developer web requests.
  - A *developer* table has a list of developers
  - There is a many-to-many developer★package relation
- Start build
  - nightly, and on Thursday* morning
  - run on both Windows and Linux
  - results to web-accessible log files
- If fail:
  - authors notified by auto generated e-mail, requested to fix, and retag with new patch.
  - managers may intervene with fixes on **Thursday***
  - Increment release patch number
  - Any new package patches *automatically* set in data base.
- If status is successful on Thursday*
  - Retag GlastRelease with new version.
  - Logs entered into database for future reference
  - Create tar-balls for collaboration.

* "Thursday" is a place holder for the day on which we decide to generate a new release: not necessarily weekly

Tasks

- Karen
  - Create Oracle tag database with tables for packages, developers, relations, logs, and update requests
  - Design a webpage for developers to submit tag updates, with authentication.
  - Generate e-mail notification to Glast mailing list of submitted changes
- Alex
  - Redesign scripts to run builds (generating a new GlastRelease list from the tag database), save results, notify developers
  - Create script to log new patch tags in data base for next build
  - Design a system for independent nightly builds of *head* versions of packages in the list
- Toby
  - Update design as needed, maintain documentation