

Policy for Observing Systems Software Development

Objective

To limit changes in observing software to only those required to fix critical bugs, meet SDSS Science Requirements, or improve operational efficiency.

Tracking Approved Changes

The GNATS database will be used to identify and track observing software bug fixes or improvements that have been approved for work during a particular development cycle. One of the fields in the GNATS database, *submitter-id*, has been modified for this purpose. The title of the field is a little misleading, but is not easily changed and so it will be used as is. More important are the definitions of the categories within the field. The category descriptions for the *submitter-id* field are defined in the on-line GNATS documentation at: <http://www.astro.princeton.edu/~gnats/meanings.html>.

Critical/High PRs

New critical/high Problem Reports (PRs) may be filed in the GNATS database at any time, but must be taken seriously by all concerned, including the filer. The developer should focus on addressing critical/high PRs until they are closed, and shaking critical/high PRs preempts all other observing tasks.

Process

1. At every other Observing Software Phone-con, beginning 10/08/01, the Observing Software Development Plan for the coming shakeup will be developed. The development plan will include all critical/high PRs, any PRs whose fixes failed shake, and a reasonable number of new PRs that are approved for work.
2. The development plan will be assembled based on recommendations and input from the observers, developers, the data processing team, and survey management regarding which PRs should be addressed. For each PR under review, the values in the *Severity* and *Priority* fields will be considered as recommendations when determining which PRs get approved for work. Finally, all PRs in the state, *submitter-id* = UNKNOWN, will be reviewed. PRs approved for development work will be flagged as APPROVED in the *submitter-id* field. PRs that are not approved for development work will be flagged as NOTAPPROVED in the *submitter-id* field. At the end of the review process, no PRs should be left in the state, *submitter-id* = UNKNOWN.
3. In addition to near-term work, the development plan may include software work that will take longer than one month to complete. The target delivery date for long-term work will be agreed upon when the project is approved at the Observing Software Phone-con. Long-term development will be done on a second software branch and merged with the mainline after the most-recent branch (for bug fixes) is merged. PRs approved for long-term development will be flagged as LONGTERM in the *submitter-id* field.
4. Developers will work on addressing approved PRs, possibly renegotiating the priorities of approved PRs depending on circumstances (i.e., PRs being harder than expected, lots of work on critical/high PRs, newly discovered PRs, etc.).

5. As each PR is completed, the developer will change the PR status to “needstest” and will assign the PR responsibility to the Lead Observer. Developers should not close PRs; the observers will do this once they verify the PR has been adequately addressed.
6. All approved PRs should be in “needstest” and assigned to the Lead Observer by the shakeup request deadline.
7. Developers will provide a list of PRs ready for testing, along with test plans for shaking each PR, by the shakeup request deadline. The Lead Observer will prepare the Shakeup test plan. If a shakeup request is not received by the request deadline, the software will be considered for testing in the following shake period.
8. The observers will test revised software during shakeup. The developers will work on any PRs failing shake so they can be re-tested during shakedown.
9. Developers will provide test plans for shaking PRs by the shakedown request deadline. These should mainly involve re-testing the PRs that failed shakeup. The Lead Observer will prepare the shakedown test plan. If a shakedown request is not received by the request deadline, the software will be considered for testing in the following shake period.
10. The observers will test revised software during shakedown.
11. Return to step 1.