

## Fast Tracking Critical Bug Fixes for RUCore

The following steps outline a scenario that will help us address RUCore problems that need to be fixed quickly. Our target is to fix critical problems in a 24 hour period.

### The Fast-track Scenario

1. A user or someone acting on behalf of a user identifies a problem that might require a quick fix. The user should notify a member of the ad hoc fast track group . It is proposed that the ad hoc group consist of Kalaivani, Rhonda, and Ron.
2. To qualify for a quick fix, the following criteria will be considered: a) a critical audience (user or group of users) is experiencing the problem or b) an RUCore subsystem is not working, e.g. the Handle server, etc. Other criteria may be identified as we get more experience with this process.
3. An ad hoc member enters the problem in software libraries .
4. The ad hoc group meets (probably via a conference call) to determine if the bug warrants a quick fix. (If some members of this ad hoc group are out on vacation, conferences, etc the other members are authorized to proceed.) If “yes”, we proceed with the fast track with the following steps. If “no”, the ad hoc group recommends that the fix be part of an upcoming bug fix or full release.
5. The ad hoc group identifies an “advocate” for the bug fix, probably a member of sw\_arch. The advocate will help facilitate the fast track process.
6. The advocate notifies the appropriate software developer (s), tester, and system administrator that we need a quick fix. The software developers will raise the fix to a high priority and provide an estimate of when the fix is ready for testing.
7. At this point, the 24 hour fast track process begins. (Note: the assumption is that the fix can be implemented quickly).
8. The SW\_ARCH team is notified that a quick fix is in progress.
9. The software developer fixes the problem, performs unit testing and notifies the advocate that the fix is ready for testing. The software developer also submits the unit tests that were used on the development server.
10. The software developer notifies the tester who does testing on the development server. Assuming success, the tester requests the software developer to package the fix and forward to the system administrator for installation on the staging server.
11. The software is packaged to support efficient installation including backup of files.
12. The system administrator installs the fix on the staging server and the tester completes testing on the staging server, notifying the system administrator when testing is complete.
13. The system administrator installs the fix on the production system and notifies the advocate.
14. The advocate monitors the activity throughout the 24 hour period to facilitate the process and help remove obstacles that might be impeding progress.
15. The advocate notifies the impacted user community, CISC, and MDWG that the fix is in place.