DCDT Development Process

The Direct Certificate Discovery Tool development team uses the Agile Scrum software development method. We have two-week-long sprints with sprint planning on Tuesdays. Our work is organized in JIRA with the Agile plugin.

Development Process and the Definition of Done

A JIRA issue enters the sprint needing no further elaboration. The goal is to close out the issue before the end of the sprint. The steps below list the process through with an issue is closed:

1. JIRA issue set to "in progress"
2. Completion criteria are met
3. Unit tests have been written
4. Local build succeeds
5. Changeset pushed to personal repository
6. Code review iterations completed
7. Reviewed changesets from personal repository merged to main repository
8. CI passed
9. JIRA issue set to "resolved"
10. QA completed
11. JIRA issue set to "closed"

Code Review

Per the DCDT development team's policy, code reviews must be completed within three core working hours of the time that they are opened. All assigned reviewers must complete the review within that time. Currently, the entire development team is listed on every code review. Code reviews are of the highest priority because failure to complete code reviews blocks team progress. Before a code review can be marked as complete all comments from team members must be addressed.

Commit Message Format

The commit message must contain the JIRA issue identifier that the commit addresses. In DCDT, these are in the format DCDT-#. It must also provide a brief description of the content of the commit. Example:

```
hg commit -m "fixes issue DCDT-9001. change: added implementation of Foo interface to AbstractBar."
```

Sprints

Our sprints are a week long and end at COB on Monday. The sprint review and the retrospective occur every Tuesday morning, immediately followed by the sprint planning meeting. In the sprint review the progress made in the previous sprint is tallied. The sprint retrospective is a time to review what went well and what could have gone better in the previous sprint. The discussions from the review and retrospective are useful for the sprint planning that follows immediately. Any issues that were not closed in the previous sprint are pulled into the current sprint and reevaluated during sprint planning.

Personal Repositories

A personal repository is a clone of the code base that is under one user's control. Development is done on a personal repository. This means that when working on a JIRA issue, all changes are committed and pushed to the clone of the main repository and not the main repository itself. Code reviews are created from the personal repository.

Merges must happen in one's personal repository. Merges must be done before a code review is created so that the code review does not contain stale code. Another merge must also be done before pushing to the main repository. Merges should never occur in the main repository. As a best practice, a pull to the personal repository should be done after every code commit from other team members.