# Code Submission Guidelines

The Development Procedures and Policies section of this wiki documents the general procedures and policies for submitting code to ONAP projects.  For the benefit of CCSDK project submitters, we ask contributors to follow the following specific guidelines, in line with best practices for most other ONAP projects.

### Guidelines for contributors:

1. Each code submission MUST include the JIRA identifier of the issue being addressed by this change.  That issue is specified in the Issue-ID footer, as documented in the ONAP development procedures and policies page noted above.
2. Each code submission that includes new code branches MUST include automated jUnit test cases to exercise the new code branches added.
3. The groupId in each pom.xml must match the repository name defined in Gerrit.  For example, all projects in ccsdk/features must have groupIds that begin with org.onap.ccsdk.features  .  This is needed because, if the groupId does not match the repository name, the merge build will fail (since the write to Nexus will fail).
4. Each pom.xml should contain a parent.  That parent should be either:
    a. One of the ccsdk parent poms defined in ccsdk/parent, if this code is intended for deployment within OpenDaylight or as a Spring Boot container.
    b. The ONAP oparent pom.
    c. Another external parent, where appropriate.
    d. The pom.xml in the directory immediately above the current directory.

### Guidelines for committers/PTL:

1. When performing code reviews, committers should check the following items:
    a. Verify that the submission contains appropriate test cases.  If the committer believes this submission could reduce the projects' code coverage, he/she should vote -1 on the submission and add a comment advising the submitter that they need to add appropriate testing.
    b. Verify that the Jira ticket listed in the code submission:
        i. Is appropriate for the code change (for example, verify that the change described fits within the description in the Jira).
        ii. Is in either SUBMITTED or IN PROGRESS state
2. After code is merged, the committer who merged the code should verify that the merge build was successful in Jenkins.
3. After code is merged, the PTL will review code coverage in Sonar.  If the merge reduces the unit test coverage, the PTL will ask the developer who submitted that change to submit a new change adding appropriate testing.