# Definition of Done

Like any other Apache project, the Geode community should agree on a common understanding of when a new feature, improvement or a bug fix is **Done** (Definition of Done). Following are a few ideas contributed by members of the community (starting Nov/ 02/ 2015):

Kirk Lund - from Spring Data Commons (additions in blue)

## CONTINUOUS DEVELOPMENT ACTIVITIES

- + Design reviewed on Wiki, especially for larger features
- Implemented according to specification in the ticket
- Unit tested
- Integration tested
- + Public/ Peer-reviewed
- + Merged into 'develop'
- + JIRA Ticket marked 'resolved'

## PRE-RELEASE (TIE-UP) ACTIVITIES

- Reference documentation updated
- Changelog updated
- Code formatted appropriately
- JavaDoc updated/ polished

## RELEASE ACTIVITIES

- + License Audit
- + Crypto Audit
- Feature branch merged into master
- Snapshot build deployed
- Ticket marked 'closed'

## COMMENTS

John Blum -

DoD is used continuously especially before a release goes out.  DoD occurs continuously with activities such as design, implementation, testing, documenting, merging (topic branches), getting feedback, iterating, etc. Other activities (Code Formatting, Polishing, Ticket Closing, etc) are "tie-up" tasks and necessarily happen at the end, right before a release.

As for the "contribution process", there is not much difference.  A JIRA ticket is filed for the PR (especially for not committers) and the ticket progresses as any other ticket would.

Kirk Lund -

The DoD is a checklist with everything that needs to be done in order to say you're done with a User Story, Iteration, or Release. Most contributors would be primarily interested in DoD for User Story, but some projects add additional steps to the DoD for Iterations and Releases (such as Capturing Release Notes).

Defining "done" simply formalizes what it means to be done. Without defining done, important work can easily be skipped accidentally or purposefully (Javadocs, User Docs/Manual, UnitTests, IntegrationTests, AcceptanceTests, product examples, license audit, crypto audit, etc). For an open source project, it's extremely valuable to have a DoD so that new contributors know what all needs to happen to contribute successfully.

The DoD should be a live document that is continually reviewed and improved as needed. It would simply be a page on the wiki that mentions things like writing UnitTests, reviewing changes prior to committing, etc.

## REFERENCES:

Good definition of Done in Jira:
http://blogs.atlassian.com/2013/10/8-steps-to-a-definition-of-done-in-jira/

Interesting blog article about Multiple Levels of Done:
https://www.mountaingoatsoftware.com/blog/multiple-levels-of-done

Good article about DoD:
https://dzone.com/articles/definitions-done-practice

Another good article:
http://sauceio.com/index.php/2015/09/what-is-your-definition-of-done/

**Work in Progress**

Please Comment/ update this page, as appropriate