Cheatography

LT D2 Team Guide Cheat Sheet

by ipf.art via cheatography.com/53872/cs/14481/

JIRA Epic/Story/Task statuses

Dev backlog	Work not started. Pending planning of the task.
In progress	Implementation in progress. Assigned to one of the developers. *Ideally max one issue in this status per developer.*
In review	Implementation complete but in code review. Waiting for acceptance from tech leaders. Assigned to one of the tech leaders.
On hold	Implementation/testing has been in progress, but currently on hold. Keep assigned.
Ready for QA	Implementation complete and reviewed. Waiting for testing. Assigned to one of the testers.
In QA	Testing in progress. Assigned to one of the testers. *Ideally max one issue in this status per tester.*
Ready for deployment	Both implementation and testing complete. Return to owner.
Deployed/-Closed	Deployed in pre-production environment. *Ideally deployed in production*
Rejected	The task was cancelled or rejected. Return to owner.

ℹ These describe the lifecycle statuses of a major item in JIRA. It is important to follow up the workflow as it helps to organize the work and keep track on the tasks.

JIRA Sub-task statuses

To do	Work not started yet.
In progress	Work is in progress.
Done	Work is done.
Rejected	Work is cancelled or rejected.

ℹ These statuses are used for developers/testers in JIRA sub-tasks to break the bigger tasks into smaller chunks of work. The sub-task resembles a logical organization of work.

💡 *A task is to create a user web service. A set of sub-tasks could be:*

- *Design SQL data model*
- *Code DAO/Repository objects*
- *Code a REST endpoint using DAOs*
- *Write unit and integration tests*

JIRA worflow explained

1. When you start implementation, set status to **In progress**

2. When you pause implementation, set status to **On hold**

3. When you resume implementation, set status to **In progress**

4. When you switch assignee during implementation, set status to **On hold** and assign issue to the other developer

5. When you complete implementation, set status to **In review**

6. When review has been approved, set status to **Ready for QA** and assign issue to a responsible tester

💡 *If no responsible tester, ask for one in testers team*

7. When you start testing, set status to **In QA**

8. When you pause testing, set status to **On hold**

9. When you resume testing, set status to **In QA**

10. When you switch assignee during testing, set status to **On hold** and assign issue to the other tester

11. When you find a defect during testing, set status to **On hold** and assign to the responsible developer

💡 *If no responsible developer, ask for one in developers team*

12. When you complete testing successfully, set status to **Ready for deployment** and assign to a release manager.

💡 *Current RM:* 👤 ***Bartosz Adamik****.*

💡 *In case of an urgent production fix, set label to* ***RC0***