Cheatography

Relaymed & Rally Cheat Sheet

by Marryat via cheatography.com/17840/cs/1713/

Rally Schedule states

Idea	(I) Provisional User Story or Defect, requires design work
Defined	(D) User Story or Defect has either been designed or is a simple story
In Progress	(P) Once work has been started by a developer
Complete	(C) Work has been completed by a developer including code reviews and updated documentation
Accepted	(A) Work has been signed off by the testing team.

How to Complete a User Story

- Review User Story and Acceptance Criteria
- Document design in the ADD document
- Take the Task within Rally, and check that you agree with the time estimate
- Write Unit & component tests to confirm the behaviour of the first part of the first task
- Write code to pass the tests you have already written
- Review tests and make sure they are correct
- Update todo estimate if sensible
- Iterate previous 3 steps till task is complete
- Refactor code to ensure it high enough quality.
- Update actual time spent on task
- Iterate previous steps for tasks until User story is complete
- Do a final refactor step to ensure code is well designed etc.
- Review ADD & update if required
- Review SMG & update if required - mark the checkbox stating that documentation is uptodate.
- Ask someone to code review work.
- Action all MoSCoW items as required and give back to reviewer to do the merge
- Mark task as complete
- Testing team will test and Review and reopen if required or mark as Accepted

Checklist for a Code Review

- Check code runs
- Does design match documented design
- Can you break by changing test values etc
- High level review to ensure methods aren't too long
- High level review to ensure classes are named properly
- High level review to check its clear what the code does

Checklist for a Code Review (cont)

- Check test coverage
- Check tests are in the correct area of code and seem sensible
- Check naming of methods and variables
- Check no libraries are in use unless agreed
- Check abstraction is required
- Check naming conventions

The code should be branched and all comments committed in the relevant part of code, and then committed back to Github. Each change should be classified by:
(M) Must Change
(S) Should Change
(C) Could Change
(W) Won't address

Raising a defect

- Enter a description of how to reproduce it
- Enter the Git Commit Hash it was found in
- Set the severity and priority (ask Marryat if unsure)
- Check if it should be scheduled for this iteration.
- Ideally link it to a Test Case
- Ideally link it to a User Story

Fixing a defect

- Git commit messages should include the DE### so it links in Rally
- Enter the Git Hash that fixes the defect in the Fixed in Field
- Enter any notes that are relevant
- Set the Defect State to Fixed
- Set the Scheduled State to Complete
- Set the Resolution

Verifying a Defect

- Enter the Git commit hash that you are verifying the fix in
- Change the Defect Status to Closed
- Change the Scheduled State to Accepted
- Add in any notes that are relevant
- Re-run the test case and record the result if present/relevant

By **Marryat**
cheatography.com/marryat/

Published 16th May, 2014.
Last updated 16th May, 2014.
Page 1 of 1.