Jira usage conventions

§1a+ $1b+ $1c + §1d: DEPRECATED (curious? -> check old text from wiki page history) During 2019 and 2020 we used team-specific JIRA Scrum boards with RICP component-based assignment to teams. We do not use this concept anymore and instead do as per §1e.

§1e: Subteams are typically small (1-20 developers) and do not focus on specific RICP components. Therefore they do not have their own scrum backlog, but rather the items are managed (JIRA states, subitem management) by the "owning" team (subteam-s, subteam-h, subteam-n, subteam-a) . We mark the fact that a subteam has picked up a JIRA item by marking the item as assigned to a person from that team and at the same time we also add one of the subteam labels (subteam-s, subteam-h) to the item. Subteam members may use the filters in §7.5 and §7.6 for listing what they are currently planned to work on.

§2: DEPRECATED (curious? -> check old text from wiki page history) No backlog sprint planning is done, i.e., we do not plan finer than per-release

§3: To access JIRA or to be assigned JIRA items you must have a free LF (Linux Foundation) account and you must have logged in at least once into LF's JIRA. Please ask your team mates to do so.

§4: The workflow is generally (a) TO DO (b) selected for development (c) in progress (d) in review (e) done. The state "in review" is a bit badly named, but it is optional for teams to use and it describes the situation that a development team considers the item as done, but the PO (product owner or similar representative/delegate of the team) has not formally accepted it yet. As of Sep-20201this state is not used by any team anymore. JIRA also has a name for each state transition under the "workflow context menu) as follows: Assign near-term schedule (TO DOselected for development), start work (selected for developmentin progress), done, but waiting for approval (in progress in review), approved (in review done), done and no further approval (in progress done). Again, note that the "grayed out" state transitions as of Sep-2021 is not used by any team anymore, i.e., JIRA items generally go from creation through three transitions before ending in "done" state.

§5.1: JIRA has a concept of releases (aka versions). We use these names **G**, F, E, Dawn, Cherry-R6, Cherry-R5. A release ZZZ_future is used to indicate items that are not yet planned to be worked on any time soon. A release ZZZ_never is used to indicate items we never plan to work on (Jira's Resolution field (with the state "won't do") is not used for anything). This way at least (new) epics that do not have any release version as "fix version", must be analyzed still on when they are to be worked on.

§5.2: For epics that were originally planned for release X, e.g., R4, and that were started, but could not be completed, please keep the original "fix version" field, i.e., “R4-Bronze” and add to that the "fix version" for release X+1, e.g., R5-Cherry. This way we know that it is continued (left-over) work (after this the item will have two "fix versions"). We know the exact leftover content from the user stories that are mapped to the Epic, but not yet done. You could also add a summary of post-R4 leftovers into the description of the epic item. Note that this way the item will show up automatically in X+1 planing. If parts of an epic originally planned for release X were not completed in X, and are unlikely to be continued in X+1, extract these left overs into a new epic marked "ZZZ_future". This way you can mark the epic that was done in X as "done". In the new epic make a reference to the original epic using a "relates to" JIRA link and add a label "movedoutofbronzeR4" (see filter mentioned in §5.3). Also explain in the description of both epics what has happened.

§5.3: Complete items (for parts of items see §5.2) that were originally planned for a release X, but were not even started yet simply (1) mark with the label ""movedoutofbronzeR4" (related "movedoutofbronzeR4" filter) and (2) move to the next release by removing the old value in the "fix version" field and replace it with the new plan, i.e., the label for X+1 if that's very likely the plan, or ZZZ_future if it is not yet clear when the item will be picked up again.

§6: We plan with the following sprints (as per Releases):

- Sprint J release dev sprint 1: Feb-19 to Mar-10
- Sprint J release dev sprint 2: Mar-11 to Mar-31
- Sprint J release dev sprint 3: Apr-1 to Apr-21
- Sprint J release dev sprint 4: Apr-22 to May-12 (last planned development sprint)
- Sprint J release dev sprint 5: May-13 to Jun-2 (only for unplanned last-minute development items)
- Sprint I release dev sprint 1: Aug-21 to Sep-10
- Sprint I release dev sprint 2: Sep-11 to Oct-1
- Sprint I release dev sprint 3: Oct-2 to Oct-22
- Sprint I release dev sprint 4: Oct-23 to Nov-12 (last planned development sprint)
- Sprint I release dev sprint 5: Nov-13 to Dec-3 (only for unplanned last-minute development items)

§7: Some useful direct links to Jira:

1. Filter All RICP items: **E** release only
    a. committed (90% sure) and tentative (90...30% sure) items: subteam-n, subteam-h, subteam-s, subteam-a and all teams.
    b. committed items only: all teams (committed)
2. Filter All RICP items: **F** release only
    a. committed (90% sure) and tentative (90...30% sure) items: subteam-n, subteam-h, subteam-s, subteam-utfpr, subteam-p and all teams.
    b. committed items only: all teams (committed)
    c. items that are known to require work, but are currently not in F release: link
3. Filter All RICP items: **G** release only
    a. committed (90% sure) and tentative (90...30% sure) items: subteam-n, subteam-h, subteam-s, subteam-utfpr, subteam-p, subteam-c and all teams.
    b. committed items only: all teams (committed)
    c. items that are known to require work, but are currently not in G release: link
4. Filter All RICP items: **H** release only
    a. committed (90% sure) and tentative (90...30% sure) items: subteam-n, subteam-h, subteam-p, subteam-s, subteam-utfpr, subteam-c, subteam-ag, subteam-cg
    b. committed items only: all teams (committed) and all teams (committed + tentative).
    c. items that are known to require work, but are currently not in H release: link
5. Filter All RICP items: **I** release only
    a. committed (90% sure) and tentative (90...30% sure) items: subteam-n, subteam-h, subteam-p, subteam-s, subteam-utfpr, subteam-ag, subteam-cg, subteam-r, subteam-gs
    b. committed items only: all teams (committed) and all teams (committed + tentative).
    c. items that are known to require work, but are currently not in I release: link

6. Filter All RICP items: **J** release only
    a. committed (90% sure) and tentative (90...30% sure) items: subteam-n (Nokia), subteam-h (HCL) , subteam-p (Parallel Wireless) , subteam-s (Samsung), subteam-utfpr (University UTFPR), subteam-ag (Abhijit Gadgil), subteam-cg (CapGemini, subteam-r (Rakuten), subteam-gs (gslab.com)
    b. committed items only: all teams (committed) and all teams (committed + tentative).
    c. items that are known to require work, but are currently not in J release: link
7. Filter: All RICP items that are not yet done: all = epics only + non-epics.
8. Test Epic and test user story Use RICP test epic RIC-42 and the linked test user story RIC-43 to experiment with JIRA. ~~Right now these are mapped to team/board for team T1: board #16 .~~
9. All Epics (not only RICP): link
10. Search for a word in any of the RICP items: link

§8: Deleting JIRA items using DELETE_ME

To delete an item please add the string "DELETE_ME" (with an underscore in the middle) into the **summary** of the item. The RICP PTL (Thoralf Czichy) will delete such items every once in a while using this filter: filter10510. Additionally, you may also send an e-mail to the PTL asking for deletion of an item.

§9: Security bugs (see also Tools (mailing list, JIRA, Gerrit))

We currently expect all security bugs to be reported publicly as JIRA issue of type "Security Bug". Other then classified as type "Security Bug", they are managed in the JIRA tool the same way as other bugs.You can use this filter (requires login) to view all security bugs: https://jira.o-ran-sc.org/issues/?filter=10701 (same filter without login: link)