Jira usage conventions

§1a+ $1b+ $1c + §1d: RICP development is done by teams, each with their own PO (Product owner). §1a: Each team has its own scrum (not kanban) backlog which is generated based on filter that is based on a mapping of components to that team. This mapping of component-to-team is visible in the RICP repository list page. §1b: For this to work - by convention - we mark each JIRA item only exactly belonging to one component; the primary component. We do not have teams as "JIRA components" - even if that might come in handy for items that are not mapped to a component yet (the PTL can create components in JIRA quite quickly). Two exceptions: the component ricp-integration for team T6 and the component ricp-e2e for team T7. §1c: In board views you see only epics mapped to a component of your team. User stories of your team that are mapped to epics of other teams are only visible if you click on "all issues" in the epic selection box. §1d ...

- T1 (Antti): board #16 (search filter: link)
- ~~T2 (Juha): board #17 (search filter: link)~~ team dissolved in Q12020 and work moved to T1 and T3
- T3 (Abdulwahid): board #18 (search filter: link)
- T4 (Avinoam): board #19 (search filter: link)
- T5 (Matti): board #20 (search filter: link)
- T6 (Lusheng): board #21 (search filter: link)
- T7 (Bharath): board #22 (search filter: link)
- ~~T8 (Ojus): board #24 (search filter: link)~~ team dissolved in Q32020 and work moved to T5
- subteam-s1 (Rahul+Arun): list of all items subteam S1 is driving or participating in (link).
- subteam-s2 (Timothy): list of all items subteam S2 is driving or participating in (link).
- subteam-h1 (Alok+Sangeetha): list of all items subteam H1 is driving or participating in (link).

$1e: Subteams are typically small (1-3 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 (T1..T8) . 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-s1, subteam-s2, subteam-h1) to the item. Subteam members may use the filters in §1d for listing what they are currently planned to work on.

§2: Backlog sprint planning is done for JIRA (user) stories. Stories are linked to higher-level JIRA Epics (which from team **planning** perspective are not particularly interesting) via the "Epic link" property of a story. Epics can be either project specific, e.g., a RICP epic, or RSAC-defined (O-RAN-SC's Requirement steering and architecture committee) epics that touch many subprojects. Epics are primarily for very high level release planning and not that relevant in sprint planning.

§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 Jan-2020 this state only used by team T4 (Avinoam). 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 Jan-2020 are only used by team T4 (Avinoam). For all other teams a JIRA items go through creation and three transitions passing through four states.

§5.1: JIRA has a concept of releases (aka versions). We use this naming convention: Bronze-R3, Bronze-R4, Cherry-R5, Cherry-R6. R3 and R4 are the first and second half of Bronze. 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: Each team creates their own 3 week sprints (and names) as per the naming scheme below which is aligned with the Bronze (B) and Cherry (C) timeline. P1-P3 are mapped to "release **p**lanning sprints". D1-D6 to "release **d**evelopment (and integration) sprints".

- RICP_T1 Sprint B.P1 (last week of previous release plus week 1 - this is exceptionally only 2 weeks),
- RICP_T1 Sprint B.P2 (week 2-4),
- RICP_T1 Sprint B.P3 (week 5-7),
- RICP_T1 Sprint B.D1 (week 8-10), Bronze-R4: Sprint B.D1 (Feb-03 ... Feb-23)
- RICP_T1 Sprint B.D2 (week 11-13) Bronze-R4: Sprint B.D2 (Feb-24 ... Mar-15), Cherry-R6.1 (Aug-31-Sep-13)
- RICP_T1 Sprint B.D3 (week 14-16), Bronze-R4: Sprint B.D3 (Mar-16 ... Apr-5), Cherry-R6.2 (Sep-14-Oct-4)
- RICP_T1 Sprint B.D4 (week 17-19), Bronze-R4: Sprint B.D4 (Apr-6 ... Apr-26), Cherry R6.3 (Oct-5-Oct-25)
- RICP_T1 Sprint B.D5 (week 20-22), Bronze-R4: Sprint B.D5 (Apr-27 ... May-17), Cherry R6.4 (Oct-26-Nov-15)
- RICP_T1 Sprint B.D6 (week 23-25), Cherry-R5.1: Sprint B.D6 (May-18 ... Jun-7)
- RICP_T1 Sprint C.P1 (last week of previous release plus week 1 - this is exceptionally only 2 weeks), Cherry-R5.2: Sprint C.P1 - EXCEPTION in 2020 (3 weeks): Jun-8 ... Jun-28
- RICP_T1 Sprint C.P2 (week 2-4), Cherry-R5.3: Sprint C.P2 (Jun-29 ... Jul-19)
- RICP_T1 Sprint C.P3 (week 5-7), Cherry-R5.4: Sprint C.P3 (Jul-20 ... Aug-9)
- RICP_T1 Sprint C.D1 (week 8-10), Cherry-R5.5: Sprint C.D1 (Aug-10 ... Aug-30) *exceptionally added to Cherry-R5 (2020) thereby shortening Cherry-R6

§7: Some useful direct links to Jira:

1. Filter: All items mapped to sprint "Sprint B.P1" : filter10308 (adapt filter string (13,14,...) to point to sprint names for the sprint your are interested in. Note you need to list seven sprints, as each team (RICP_T1..7) has its own sprint).
2. Filter: All items mapped to release Bronze mapped to your team: use filters from §1 and add this to the search filter: "AND fixVersion in (Bronze-R3,Bronze-R4)" before the "ORDER" part of the filter.
3. Filter: All RICP items (epics, user stories, ...) mapped to release Bronze: filter10400 and Bronze-epics only (i.e. no user stories or others) filter10401 and Bronze-R4-epics only filter10509.
4. Filter: All RICP items (epics, user stories, ...) mapped to release Cherry: filter10601 and Cherry-epics only (i.e. no user stories or others) filter10602 and Cherry-R5-epics only filter10603 (incl. stretch goals) filter10605 (excl. stretch goals) and Cherry-R6-epics only filter10604
5. Filter: All RICP items that are not yet done: all = epics only + non-epics.
6. 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 .
7. All Epics (not only RICP): link
8. 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.