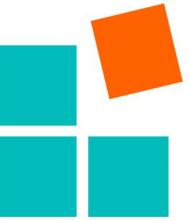




User study



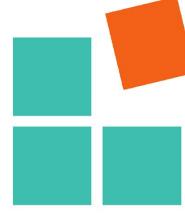
The objectives of the study

Hear experiences from users who have contextual information i.e. are using Jira.

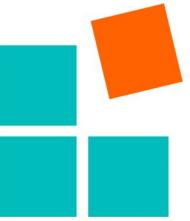
User study, not usability study.

- Validation rather than verification

No “right” answers but opinions and experiences.

OpenReq 

Visualization



Link Map

For what purpose do you use the link map?

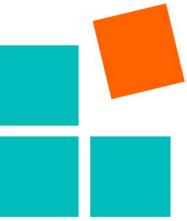
- What is the objective or information searched for?
- What is the use context or situation?

How do you use link map?

- What is the sequence of actions?
- How often each “feature” is used? (visualization, depths, filter, proposed, consistency check)

Does it have all relevant information?

- What could be improved?
- Add/remove/change?



Link map behavior and filters

What filters are relevant?

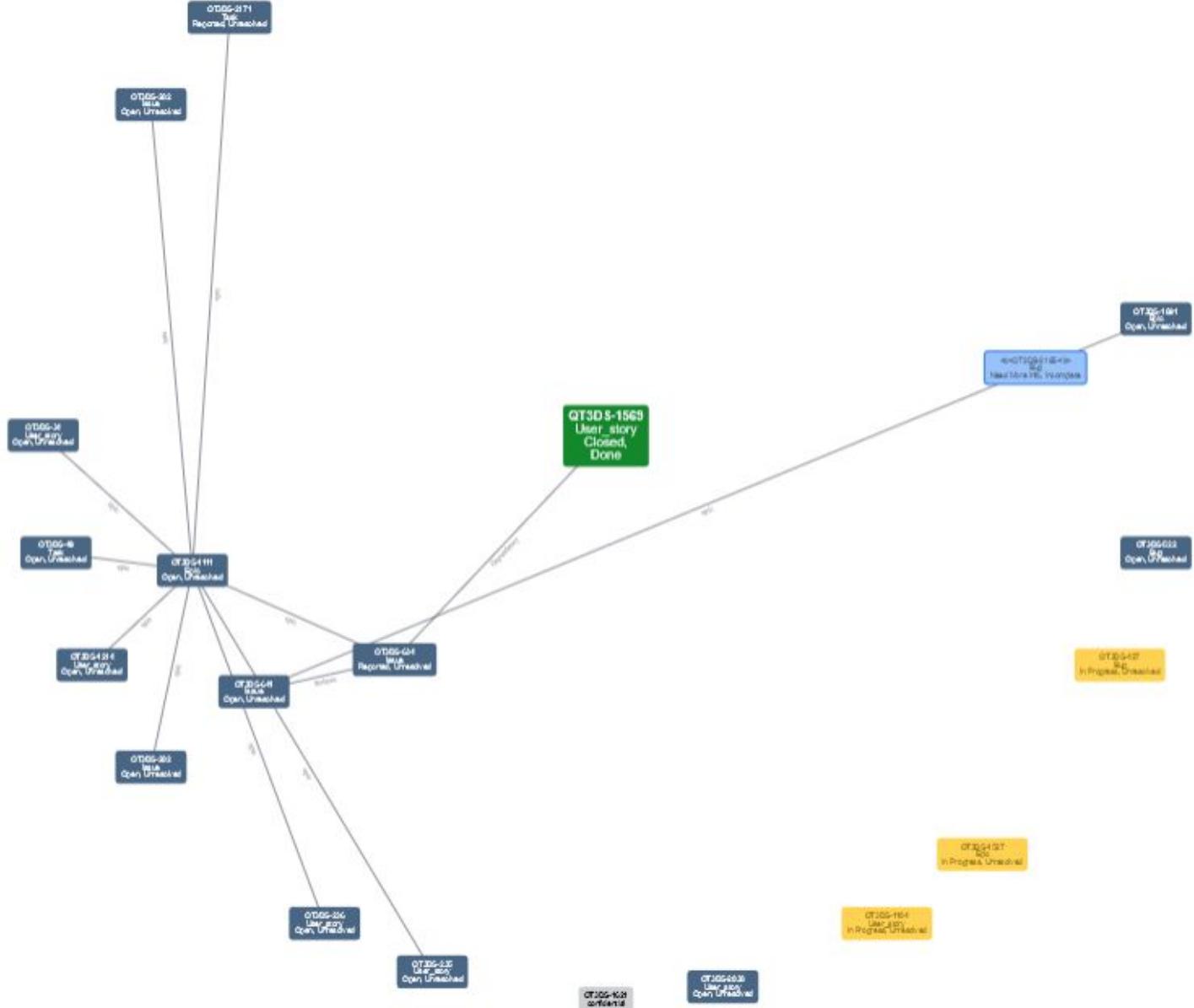
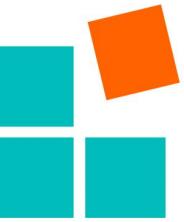
How filtering should behave?

- Does the information make sense?
- Can the graph be in “pieces”
- Text-search or some other filter?

What to select by default about filters?

Is depth 5 enough? Is focus on transitive closure reasonable?

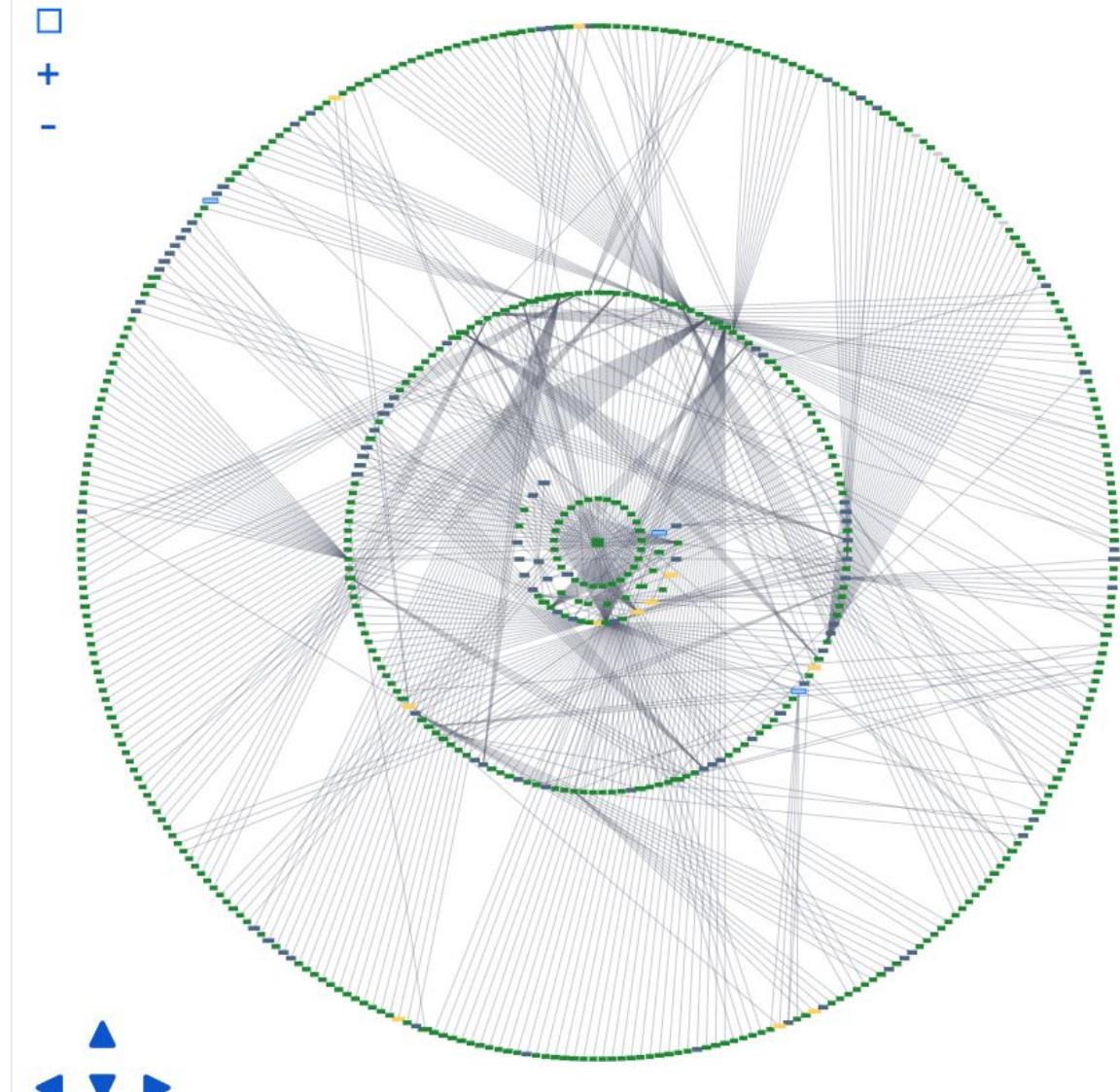
Useful to indicate (by ... or something) that some links are not shown, i.e., there would be more on the next level?



Issue Links of QT3DS-1569

Depth 1 Depth 2 Depth 3 Depth 4
Depth 5

Filter: Type ▾ Status ▾ Priority ▾ Link Type ▾





Link typology

Is hierarchy needed to be emphasized: Epic, user story, task?

Semantics to edge visualization (arrows, dashes, whatnot) to differentiate?

- Relevance of direction of link? For all types?

Are the current link types sufficient

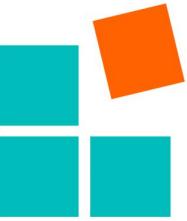
- More or less types? More complex links?
- Or is it more relevant to just see connections

Is “excludes” (or incompatible) relevant between issues: Bugs or requirements?

- Not supported in Jira



Consistency check & analyses



Consistency - status now

Is it clear what consistency means?

Is consistency a problem?

How do you check it now and earlier?

How much are there inconsistencies?

How often inconsistencies are noticed?

How much effort is spend on finding inconsistencies?

What are implications of inconsistencies to practice?

Are the rules same for all projects that cannot be checked by Jira itself

- Any required issue (A requires B; A->B) must not have later fixversion (or lower priority).
- All child Jira Issues, which have the same or higher priority, must not have later fixversion
- Duplicates are “merged”

Are other rules needed?



Consistency check

Does visualization help per se/alone?

Does formal consistency check help?

Is consistency check usable and understandable?

Would a user understand what the lists provided by the consistency checker contain?

Is consistency relevant information for “Done” issues?

What is good context for analysis. Is transitive closure good?

Should there be other such as release?

Is default depth=5 right restriction? Should it be parameterizable?

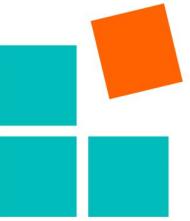
Consistency Checker

Release plan is inconsistent.

Inconsistent items +

Releases found +

Ignored links +



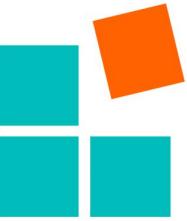
Inconsistencies?

What to represent about inconsistency to a user?

- Currently, incorrect links.
- The problematic link, fixversion or priority?
- Visualize in the graph?

What to do with inconsistencies?

- Change priority, link type, release. Which ones?
- Do not care? It's OK to be inconsistent as long as it is known!



Cross-project consistency check

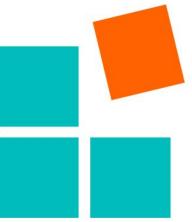
Are ignored links clear?

Is it relevant to check links between projects? (like between QTBUG and QTCREATORBUG)

Is it understandable why cross-project ignored are now listed?

Ignored but listed.

- Links not possible to check because it on fixversion (which are not comparable between projects)



Diagnosis (currently not visible)

Inconsistency (*What is wrong*) or diagnosis/repair proposal (*What to change to fix it*)?

Different diagnosis/repair proposals that result in consistency?

- E.g. remove link, change priority, change fixversion?
- How many to show?
- What is optimal?

Are these relevant or obvious from the context? Is it enough to show link that causes problems?



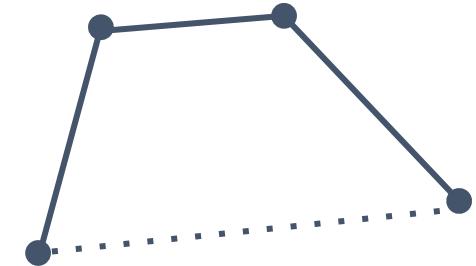
Redundant links?

Are there redundant links? Are they harmful?

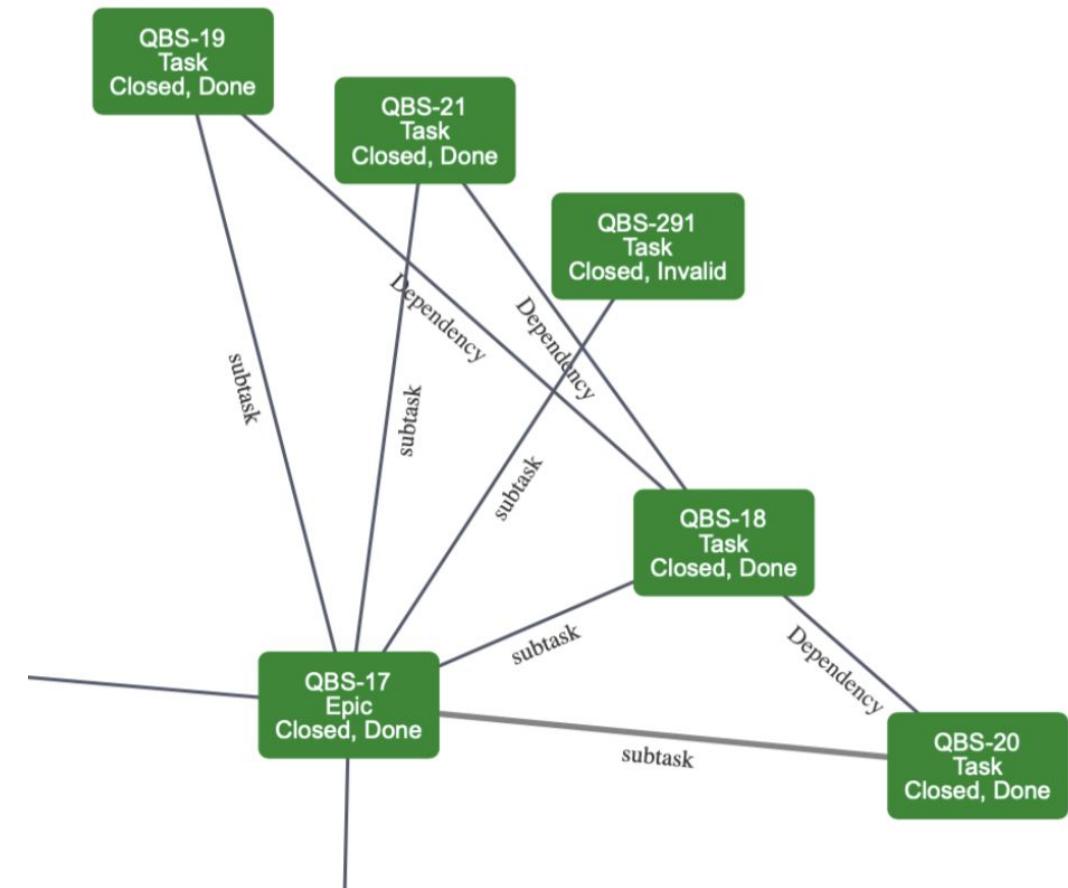
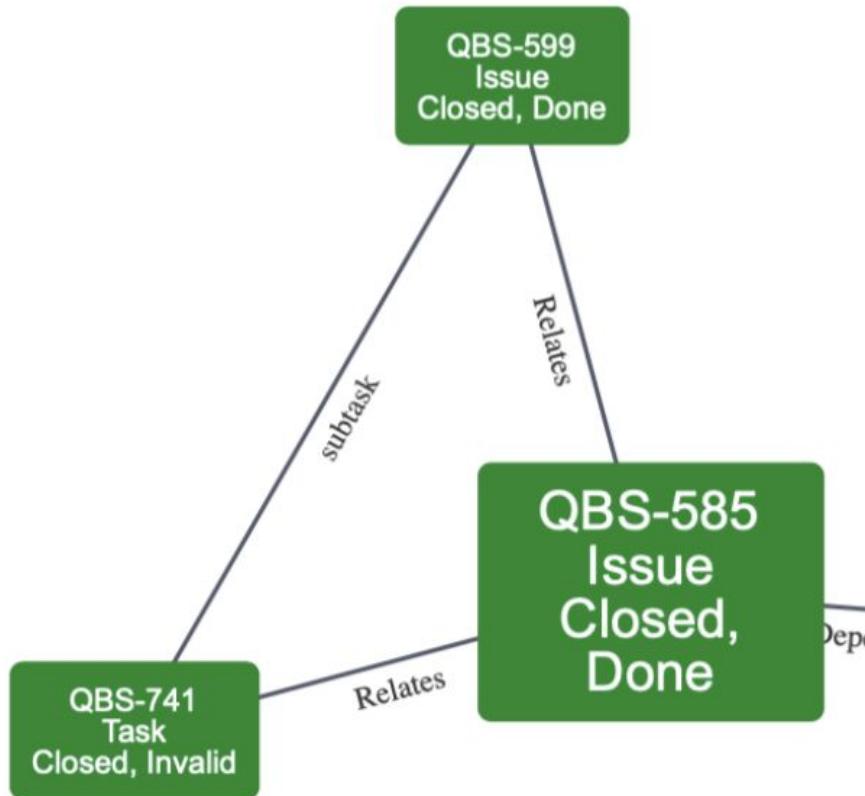
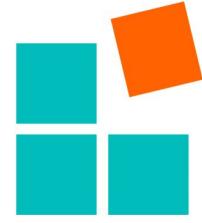
Does semantics matter in the chain of links?

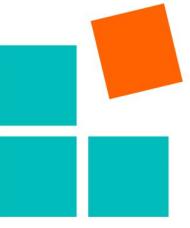
Which one is redundant -- E.g. in the figure the dashed or one in the chain?

Should potentially redundant links be highlighted, removed or something?



Redundant links, examples?





Other analyses?

What else should be analysed?

- Characteristics such as size, number of X?

What other information is inconvenient in Jira?



What is “Consistency check”

QTBUG-17305
Bug
Closed,
Done

Dependency

QTBUG-18795
Bug
Closed, Done

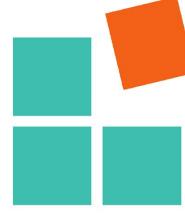
Fix Version 4. 8. 0

QTBUG-17305
requires
QTBUG-18795

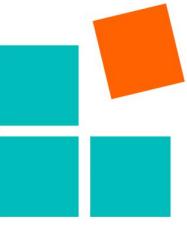
but: required cannot
be in later fixversion

Fix Version 4. 7. 4

- Detect inconsistency between two linked issues
 - Link incorrect
 - Fix Versions incompatible and/or
 - Priorities incompatible
- Currently implemented
 - Dependency (or requires)
 - A requires B
 - B must be in same or earlier release
 - Decomposition (or parent-child)
 - epic - user story, epic - bug
 - user story - task, user story -- subtask

OpenReq 

Link proposals



Link proposals

How big problem is missing links?

Are orphans or pairs problem or links in general?

How hard is it to find links or related issues by hand?

How inconvenient are false positives?

How to order proposals?

Should scores be visible? Other justification?

How many links should be proposed?



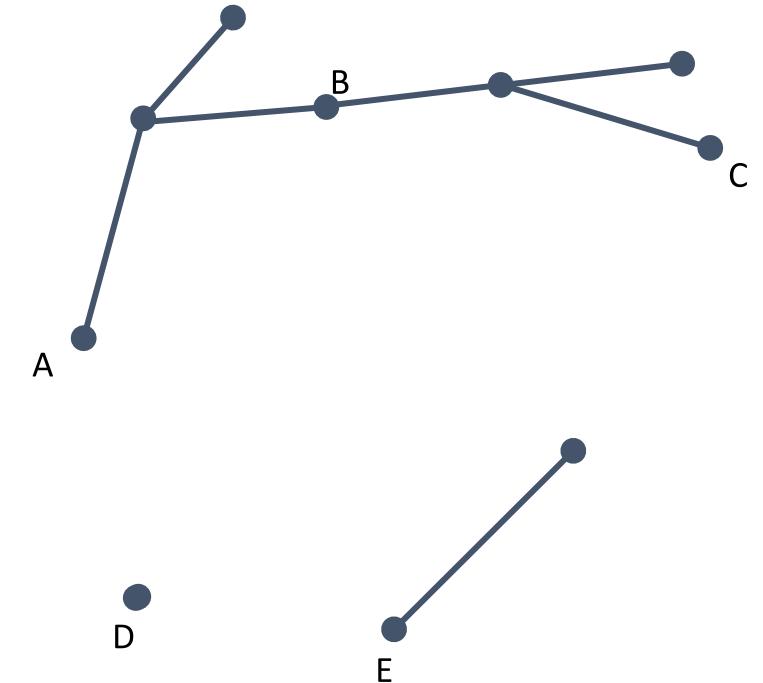
Link proposals?

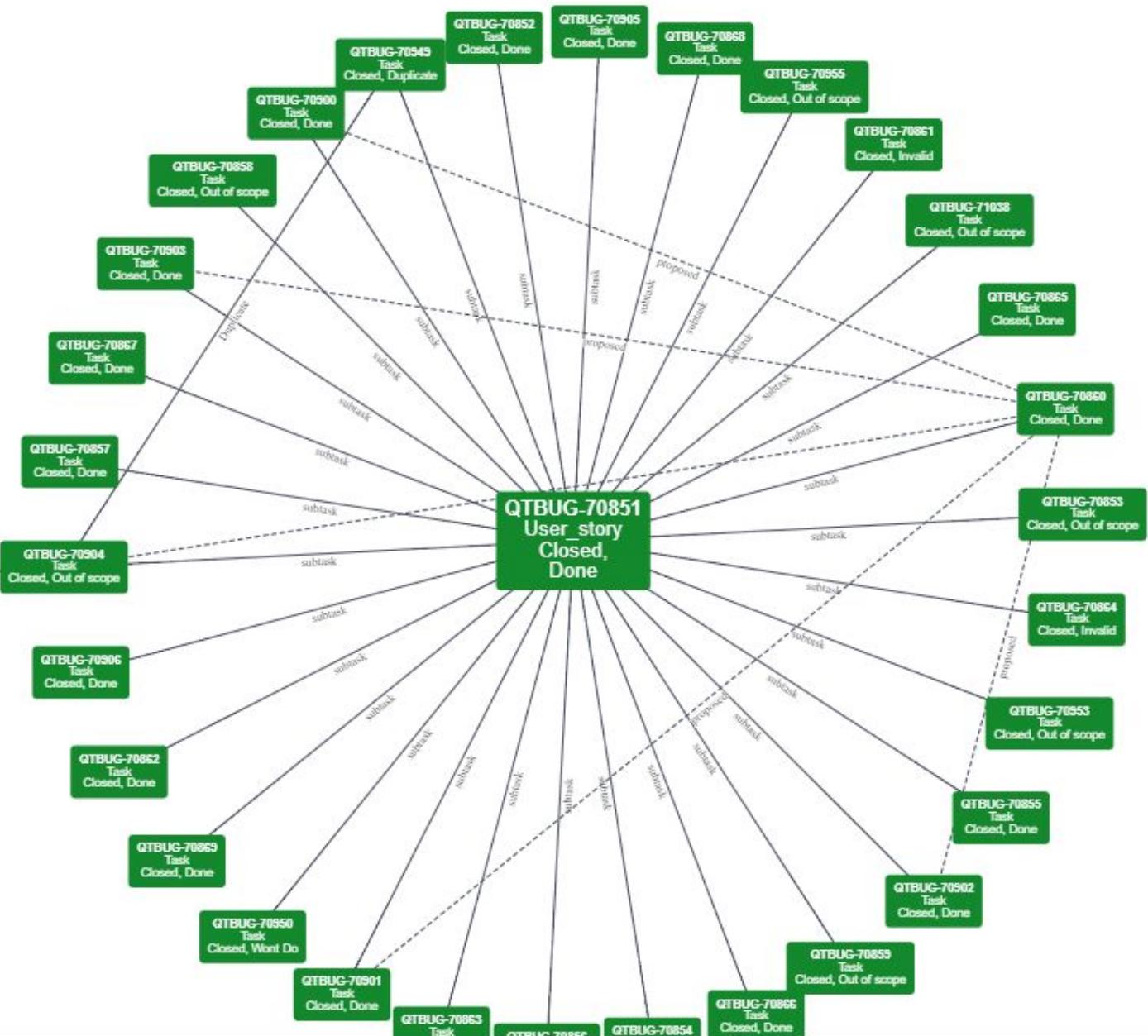
How close ones are relevant? (A-B?)

How important are orphans i.e. issues that have no links yet?

Does size of the network matter?

What if the starting point itself is an orphan?





Example: The tasks of user story are proposed to be linked to each other because they are not directly connected.

Now only 5 highest scores
are proposed but probably
all other task would be
proposed (dashed line).



Link proposals

How to weight/prioritize
(up/down) proposals?

OR

Filter for proposed ones?

Is score weights understandable?

What properties should be taken
into account?

Weight the results

Orphans	<input checked="" type="checkbox"/> Yes	1.5
Distance	3	1.5
Project	QTBUG	1.5
Component	WebEngine	1.5
Label		
Created	2019-09-11 - 2019-11-11	1.5

Apply



Accepting/rejecting proposals

What should happen when rejected?

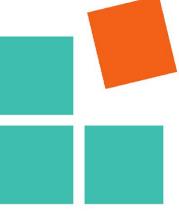
Is it decided by one accept/reject decision?

Is accept/reject too easy? Do same rights apply as for editing links?

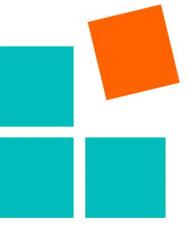
What if left unchecked?

Is type selection clear?

Proposed Links of QBS-1009			
Issue Key	Link type	Accept	Reject
QTCREATORBUG-11866	dependency	<input checked="" type="checkbox"/>	<input type="checkbox"/>
QBS-543	dependency	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Save			

OpenReq 

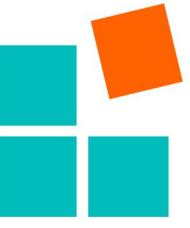
Other



Update scenarios?

How often should updates take place?

Is it tolerable that if update takes place once an hour (or every 15 minutes)?



What else?

Release planning (defining) help?

Editing possibilities?

Check specific fixversion or focus otherwise