

# Subtasks in Jira - a good or not such a good thing?

Being an Atlassian Administrator and developer for 10 years, I've done a lot of Jira deployments, administration, managing and developing workflows.

This is ITIL stuff as Incident- and Change-management, Tasks etc. But also more special needs as a large CMDB, Customer, Subscription, Employee, License and Contract Databases. Internally in my Company we use Jira for all sorts of thing, including being "mailbox" for a lot of the standard company mail-addresses as job@ support@ info@ etc etc; so we don't need distributionlists and all incoming mail are handled and tracked probably by an assignee.

As an admin, You probably would know that Subtasks, defined as **an issue being a child of another issue** (we will call it parent and child) - is an option You can select to disable in Your Jira.

It can be disabled under Gear Issues Sub-tasks.

Let us look at some of the things around Subtasks.

- Its an unfortunate name, as there also is a default issuetype called **Sub-task** (with a hyphen). This is very confusing for many users and admins, that **Subtasks** is a concept for a relationship, as well as an issue type called Sub-task. The concept could have been called child-tasks or children (to resemble the Confluence page relations)
- Subtasks always are in the same project as the parent.
- Subtasks to subtasks are not possible, there is only one sub-level
- Subtasks are not useable for Jira Service Desk Request Types.
- Jira has little awareness regarding the interaction between

Subtasks are in general fine to use, but if I were to start a new Jira installation, I might tend to start by disabling Subtasks totally - and use Issue linking instead to have the same (and wider) possibilities.

## Why?

There are several reasons:

### Remove confusion and simplifying

You remove the confusion between the concept and the issue-type.

### Tidy up the UI

Subtasks and issue links are represented in 2 different styles / ways in the UI, it does not look well, and as a user You are never sure where to look for related tasks under Subtasks or Issue Links



I have grown very fond of the [Issue Matrix](#) App to display linked issues and issues that can be correlated by fields.

Its awesome, flexible - and includes inline edit 😊

### Added flexibility

As Subtasks only can live in the same project as the parent, it can't be moved to other projects, this is quite a limitation for many companies that has different Jira projects for different departments/functions when subtasks are used for delegating work.

Some users won't make a subtask, but a linked issue - other users might create a subtask and then a linked issue in another project.

## Double programming of functionality

As Jira has no builtin awareness/functionality that relates to the parent-child or linked issues relation (Links are out of the box primarily a visual enhancement to show a relation), I am often asked by customers to program such things - this is often:

- Some awareness of issues blocking other issues; like when a issue/subtasks is closed, other issues are made aware of this by transition or comment.
- Duplication of comments between issues; some issue are added an comment (could be by a customer), and parent/child/linked issues are automatically added the same comment.
- Duplication of attachments between issues; some issue are added an attachment, and parent/child/linked issues are automatically added the same attachment.

Very often, these functionalities need to be programmed twice; one for subtasks and one for linked issues.



Several Apps can perform part or all of the points above from the UI - my primary favorites are [JSU Automation Suite](#) and [Automation](#).

But I mostly use [JIRA Scriptrunner](#) - I does require programming, but extends flexibility of the function; the downside is its more time consuming and you have to maintain the groovy code Yourself (between upgrades etc)

## Limitless

As facted, subtasks to subtasks is not a possibility - in JIRA this can be partly overcome by a great App like [Structure](#)

With links, this limit is not existing; but of cause there is a danger in the fact that links can end up being more of a mesh that a structure.

Some other Apps like [Issue Links Viewer](#) or [Pathfinder](#) can help visualising the linking.

## Controlling links

In every Jira installation, issuelinks are somewhat hard to handle, as the links types created over time often has som issue minded relations.

And pr. default, all linktypes are available for all issue-types.

For some customers we have added the App [Extended Schemes for Jira](#), which give the ability to (amon other functions) to control which links can be used for issue types (Link Scheme).

## Changing peoples process

Im sure that removing or disabling subtasks can cause frustration for hardcore and experienced users of Jira, so do always remember to argument and educate for the solutions You implement. Its all a matter of the way people work and are educated to use Jira ....

## All good?

Of course not. Not using subtasks has some drawbacks, one of them being worklog and the logged work time presentation on the parent, where all work on this and subtasks can be seen and measured..

There are som ways to overcome this, I have actually written a groovy script that can travel links from a certain tasks and sum up logged work in all links.

## Real life story

Some years ago - Ive created a sub-task issue type called Sub-change