REQUIREMENTS REUSE & SYNCHRONIZATION PLAYBOOK:

A how-to guide to sharing across products/projects within Contour.

What is requirements reuse and synchronization? Many organizations are looking for ways to streamline inefficient processes by reusing requirements that are common across projects, platforms, or releases. Leveraging these already existing artifacts from one project to another greatly increases efficiencies while decreasing time to market.

However, it is important that projects maintain their ability to manage aspects of these common requirements within their own lifecycle. Contour solves these seemingly conflicting goals by synchronizing only specific details, such as the requirement name and description. This frees up the detailed attributes such as status, priority and release to be managed within the specific project. Within this playbook, we provide a step-by-step guide to help you use this capability, which has been broadly enhanced in Contour 3.6.

⸭ Requirements Reuse: An Overview of the Process

1. Create a product line with multiple sub-projects
2. Reuse requirements
3. Make changes to core requirements
4. View changes across projects
5. Synchronize changes across projects

⸭ Example Scenario

Jason Benson is a Product Manager for a mobile phone company. His team will be developing multiple types of phones for a new product line. These different phones all share a core set of requirements. Jason would like to avoid duplicating efforts and losing track of changes to the core requirements. He will use Contour's Reuse & Synchronization capabilities to manage the core requirements from within a central location.

Kristi Wolf is a Project Manager who will be responsible for the Mobile Phone X, which is a specific phone in Jason's product line. She will need to synchronize the core set of requirements while continuing to manage them through her project's lifecycle.

**Step 1:** Create a product line with multiple sub-projects.

To begin, Jason will create a core project in Contour. This project will contain the core set of requirements. Kristi will create a separate project, which will represent a single variant of the core project.

Jason will create a project in Contour called "Mobile Phone Domain" to store the core requirements for the entire product line. Jason will also use this project to define the scope and vision of the mobile phone product line.

Kristi will create an individual project called "Mobile Phone X", which will be managed separately but will need to reuse and synchronize requirements from Jason's project.


    www.jamasoftware.com / 1.800.679.3058
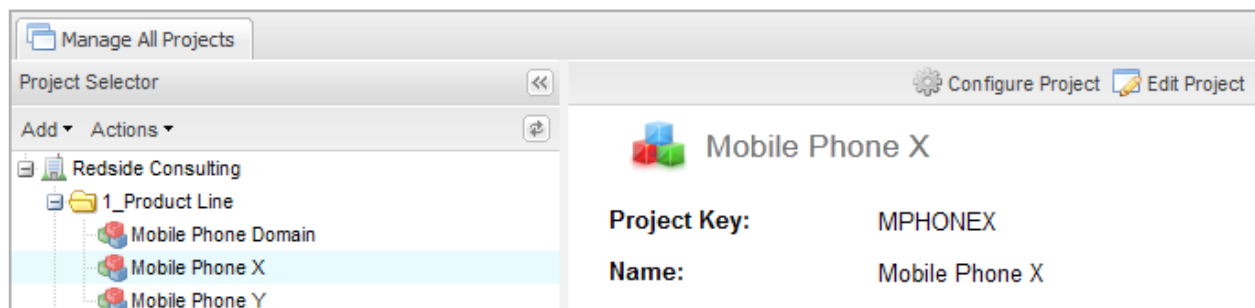
Jason will begin by defining and/or importing his core Requirements into the Mobile Phone Domain project.
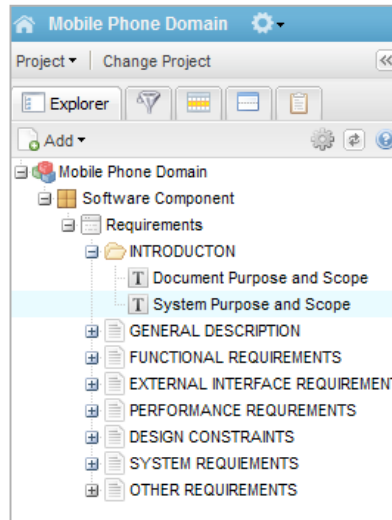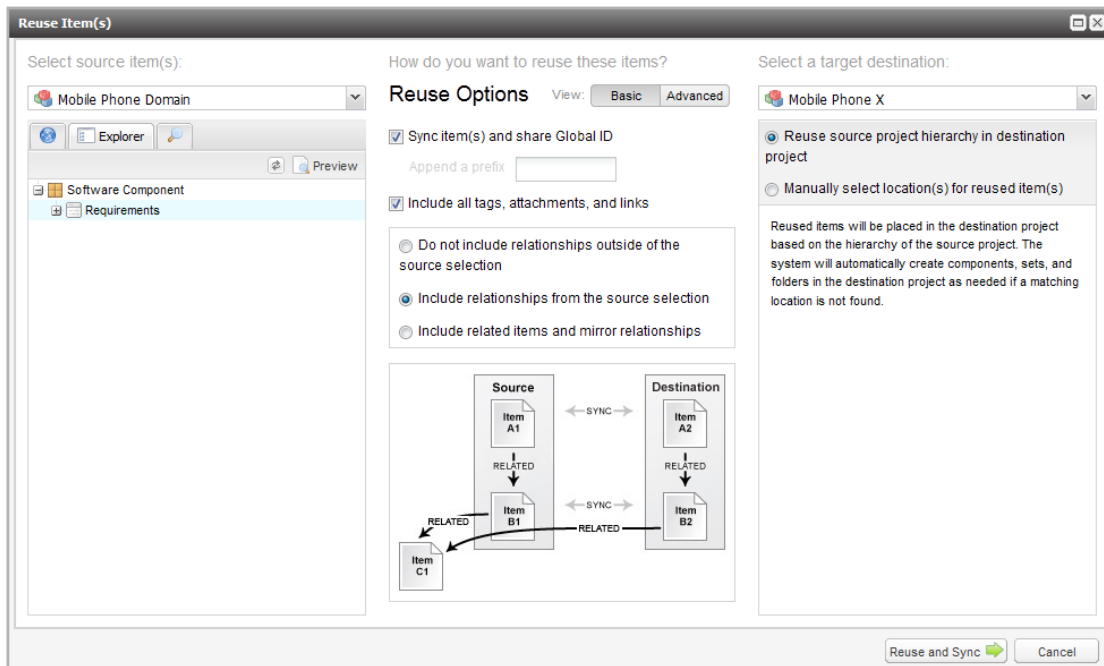
Step 2: Reuse requirements.

Jason's team of Project Managers (e.g.. Kristi) will begin managing their individual types of phones. They will reuse and synchronize requirements from Jason's product line repository. This will avoid duplicating efforts.

As part of setting up her project, Kristi will reuse all necessary requirements from Jason's core project repository.
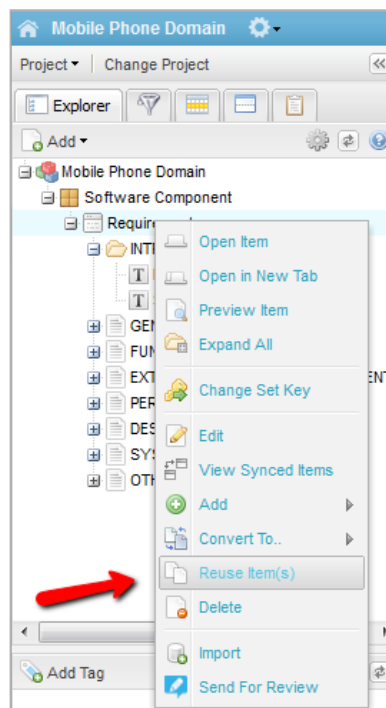
Kristi will do this for all components her particular mobile phone type needs. In this case, she will reuse all the software requirements in the domain (or library) project.

Via the Reuse Item(s) screen shown just below, Kristi has a lot of options for reusing these requirements. In this instance she has decided to sync the requirements into her Mobile Phone X project while bringing along all the items' tags, attachments, links, and even continuing to use the relationships (traceability) to any items outside of the source set of requirements.

For her project she chooses to reuse the source project's hierarchy, which means the layout in Mobile Phone X will equal that seen in the domain project.
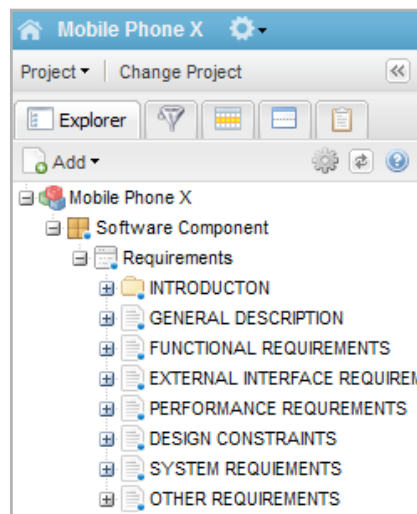
Now that Kristi has the necessary core requirements she can begin adding requirements that are specific to her type of phone. The advantage is that the core requirements are now part of her project and are synchronized with Jason's (represented by the blue dots in the screenshot above). This enables her to manage the requirements within her project lifecycle.

Step 3: Make changes to core requirements.

Jason will continue to manage the core requirements, making changes based on new rules, feedback from customers or internally driven changes. Jason will need to make sure projects that reuse these requirements are aware of these changes.
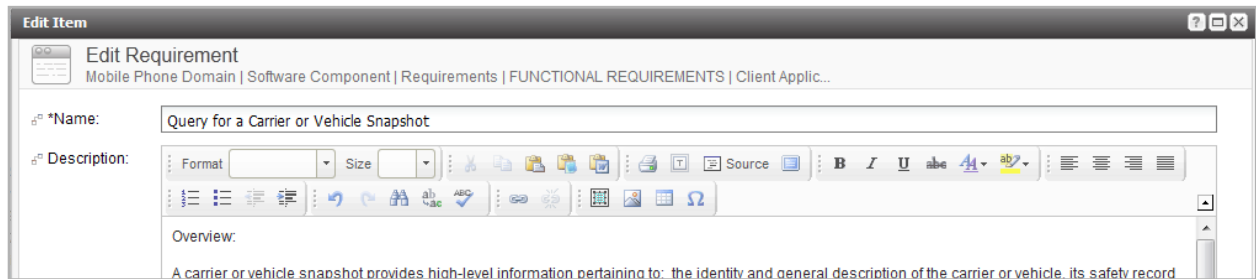
All the projects that are synchronized will be aware of this change. Kristi can quickly see that her copy of this requirement and others are out-of-sync. The screenshots below walk us through this flow.

While Kristi could quickly go to a specific requirement she was curious about, Contour also allows her the option to view the synchronized items across a set, a component or even the whole project.
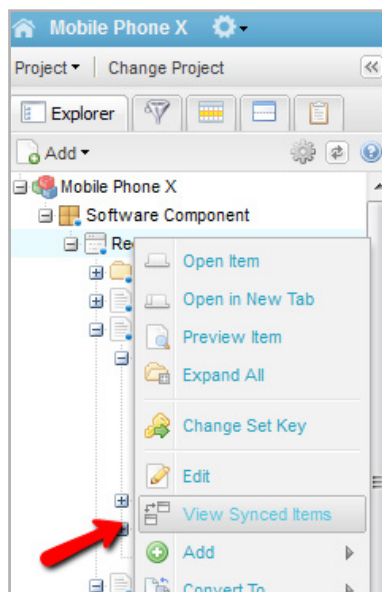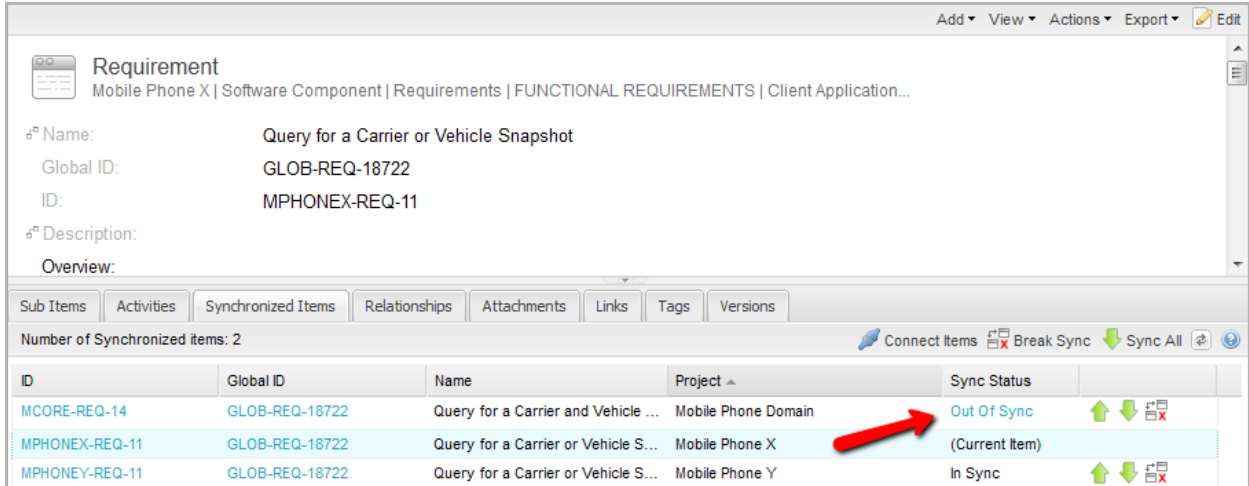
Once she has selected the item(s) from her project, in this example her requirements set, she then chooses which of the other synced copies of this set to compare. In this case she selects the domain project's set.
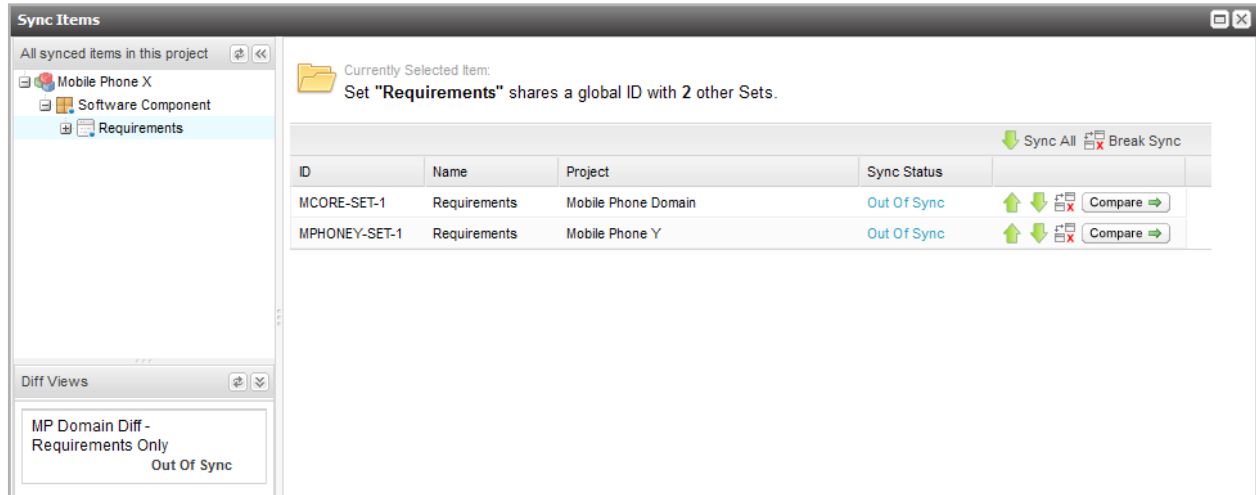
While Kristi could have immediately clicked one of the green arrows to immediately bring everything in sync, she instead chooses to view the details of the changes in a convenient side-by-side comparison window shown below. Here she has many options available to her like choosing which details of the items to display, whether to only display out-of-sync items, or to even use a saved Diff View that she created earlier.
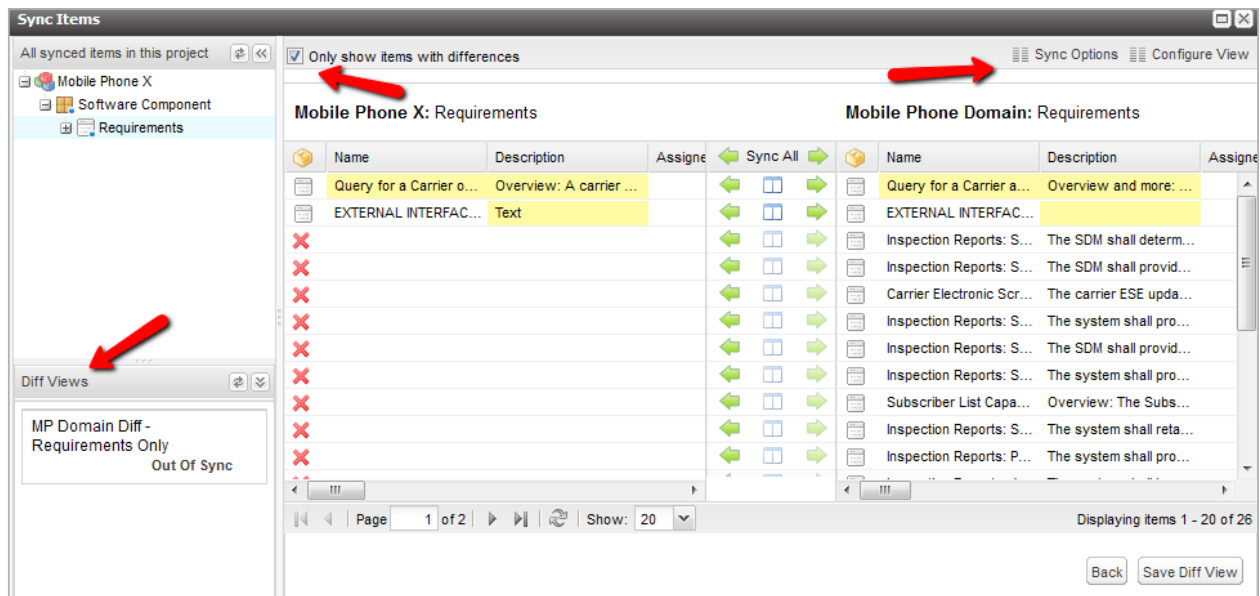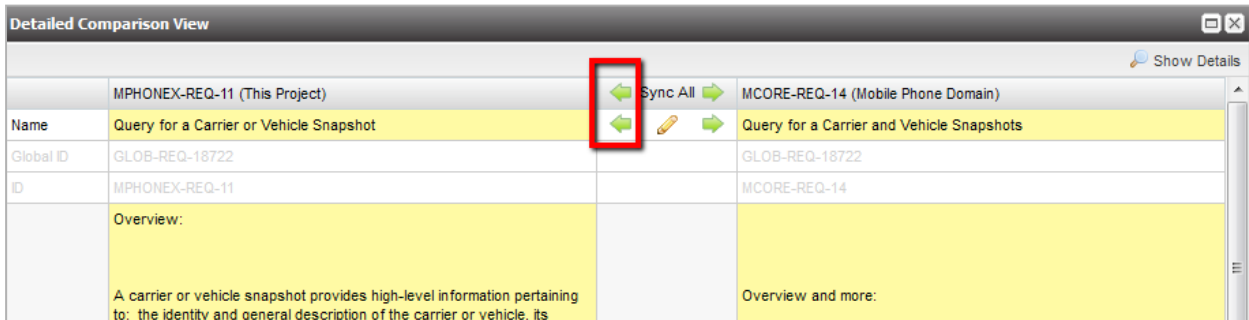
Since Kristi really just cares about one specific requirement right now, she goes to that item's Synchronized Items tab and works from there.
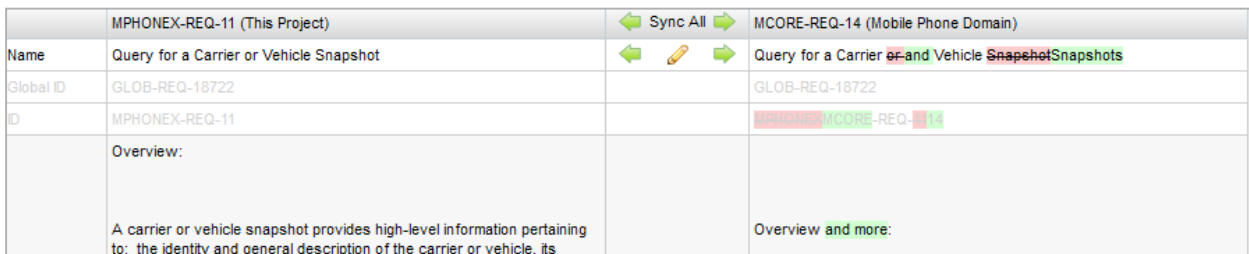
Clicking the Out of Sync link displays a detailed comparison view where Kristi can see exactly what changes Jason made to the domain's copy of this requirement.


Details:

Clicking the top left green arrow brings Kristi's Mobile X copy back in sync with the domain's. Alternatively, as you'll see in a moment, Jason can also manage those changes from his core project.

Steps 4 & 5: View and synchronize changes across projects.

Once Jason has made several changes to his core domain project, he will view the changes across all impacted projects. Here he could immediately click Sync All and bring all connected projects back in sync, but for this illustration he will focus on just syncing Kristi's project.
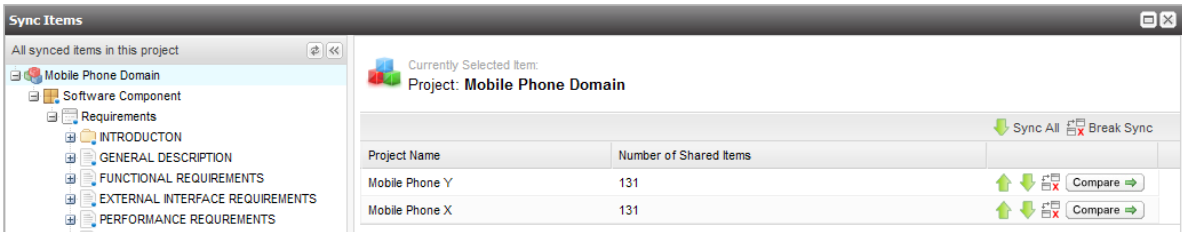
Jason decides to compare the requirements set between his project and Kristi's in a similar fashion that she did above. In order to ensure that Kristi is working on the most current requirements he clicks a green sync arrow to push his changes to the connected artifact(s).
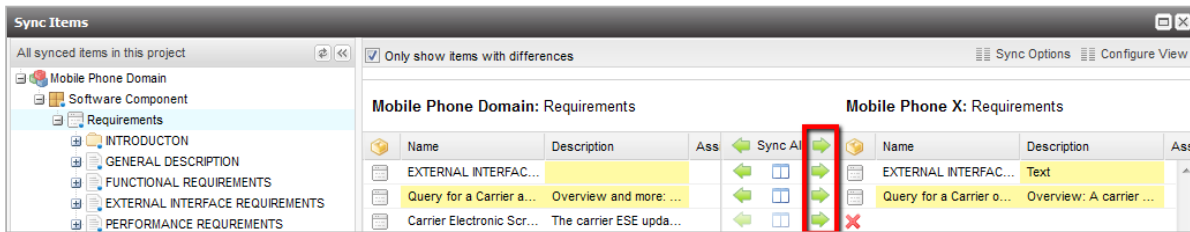
Kristi can now see that her Mobile Phone X version of the requirement is back in sync with Jason's core domain copy.