**Cross-Project Tracking**

Blessie, Tech Mahindra

Last updated: February, 2017  
Build basis: IBM Rational Team Concert 5.0.2

**Key Points:**

Work that is implemented in different projects can be tracked from a single project.

The project can be of any type of process configuration(Scrum, Formal Project Management)

Work items of different projects of same repository, as well as work items of different projects of a different repository, can be linked to each other by using the new link type, **tracks.**

Using Cross Project Tracking, you can track the overall schedule for the project as well as see what needs to be done in each of the projects to complete a particular cross-cutting task.

**Introduction:**

Large teams often deliver features where development and execution requires the work of multiple teams over multiple projects. The work must be planned and tracked in a manner that shows the progress and completion of the high-level work items.

Ex: Work is implemented in two different projects and is tracked in a third project.

**Note:** Each project can use any type of process configuration such as the Scrum, Formal Project Management template, etc.

 In order to support cross-project tracking, work items can now be linked to each other by using the new link type, **tracks**.

Let’s assume you build a diagnostic tool for a teller machine. It should consist of a user interface, a backend, and hardware additions to the teller machines. You have different teams working on these efforts in three different projects:

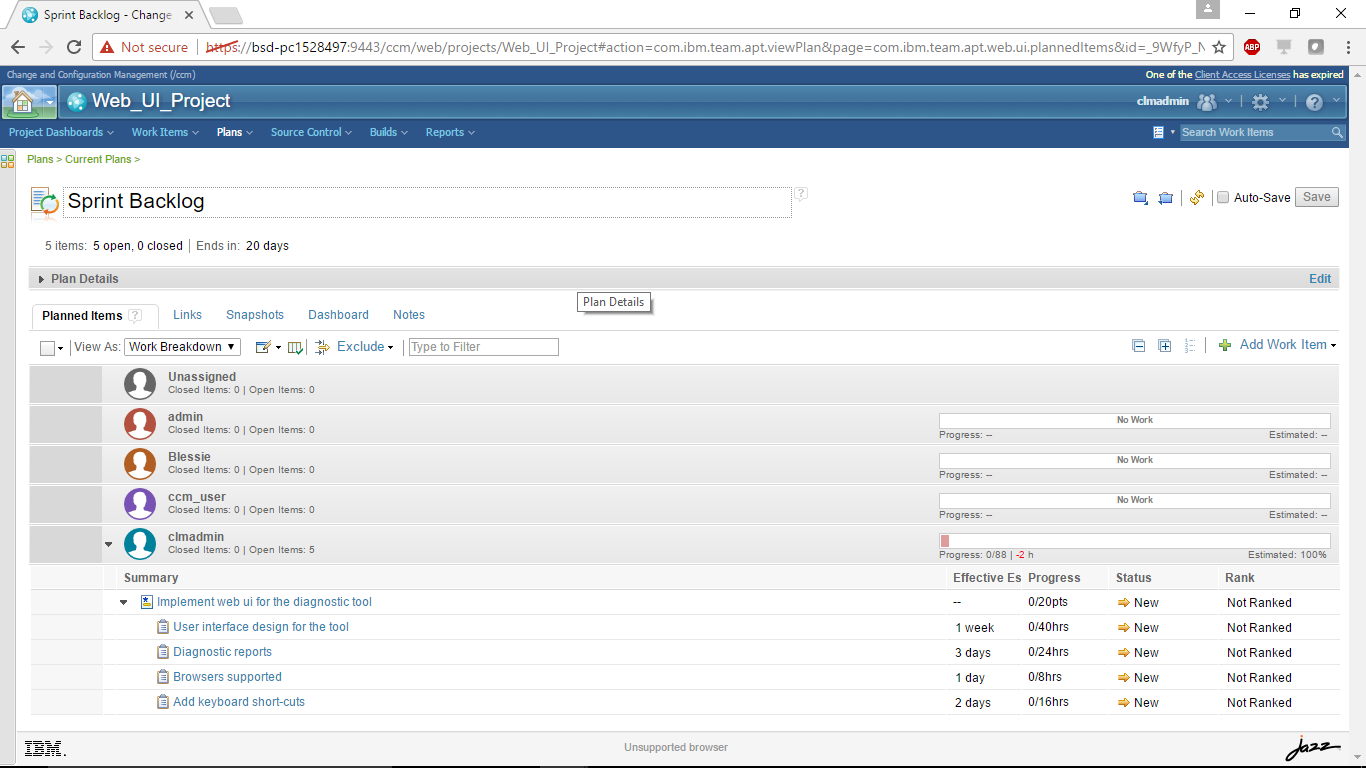
* **Web UI Project**: This project is where the web UI features of the tool are executed.
* **Repository Project**: This project is where the repository or database features of the tool are executed.
* **Hardware**: This project is where the hardware needs of the tool are tracked.

You now want to track the overall schedule for the project as well as see what needs to be done in each of the projects to complete a particular cross-cutting task. What you need is a cross-project plan.

**Implementation of Cross Project Plan:**

### **Development/Execution Plans**

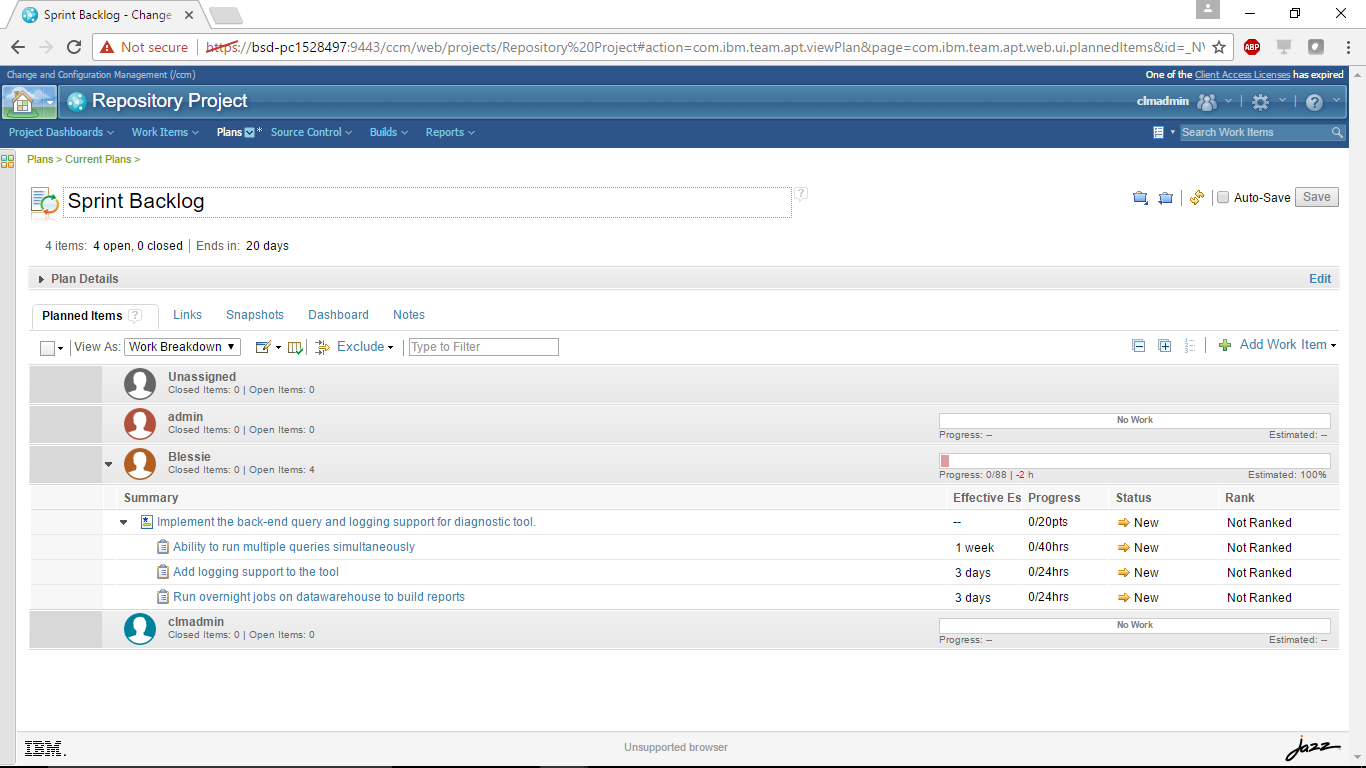
The work that needs to be tracked is in development plans that have already been created as part of execution of the work. There are plans and work items for the diagnostic tool feature in Web UI Project, Repository Project and Hardware Project.



As shown in the plan above, the following work items are created in the Web UI project:

* Story with the summary,*Implement web ui for the diagnostic tool.*
* Child work items with the following summaries and estimates:
  + *User interface design for the tool*, 1 week
  + *Browsers supported*, 1 day
  + *Add keyboard short-cuts*, 2 days
  + *Diagnostic reports*, 3 days

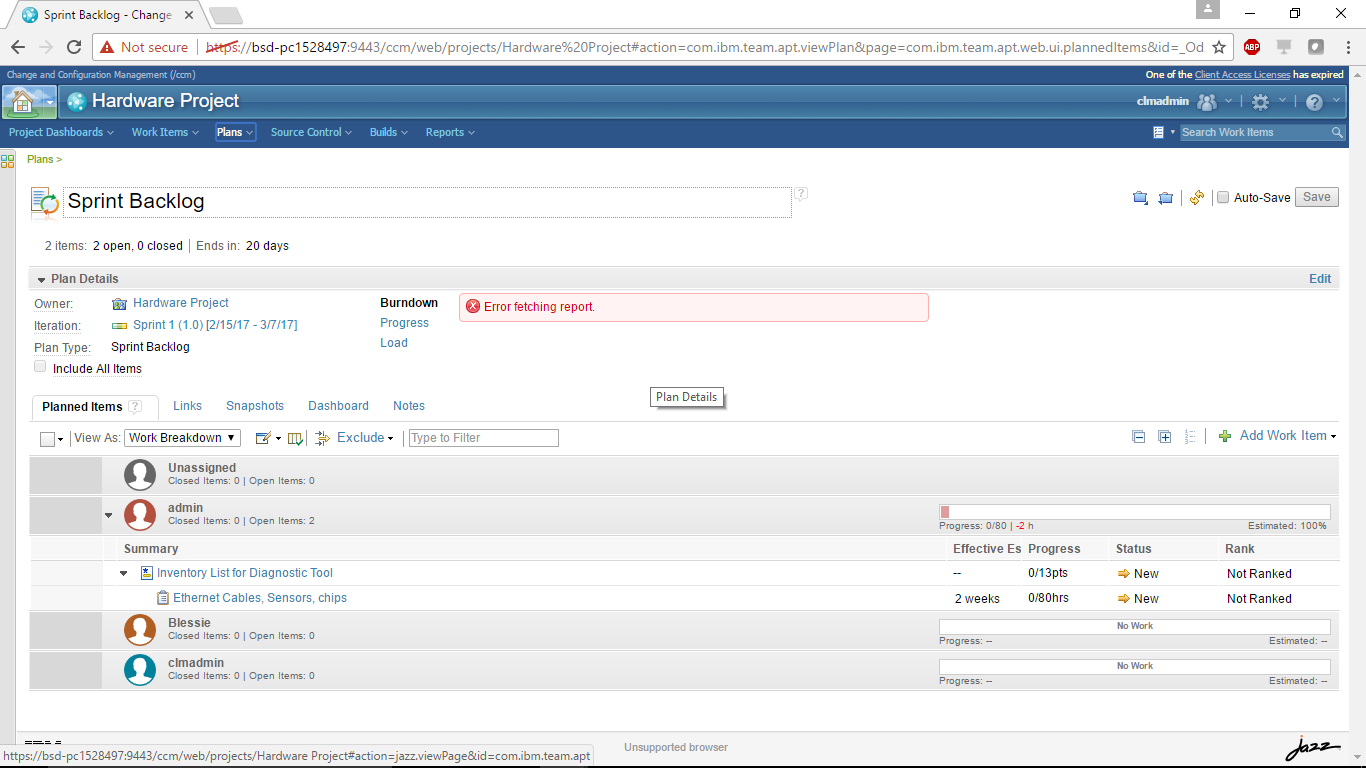
In the Repository project, create the plan and work items to implement the repository or database feature for the tool. This is handled by a different team.



As shown in the plan above, the following work items are created in the Repository project:

* Story with the summary, *Implement the back-end query and logging support for diagnostic tool.*
* Child work items with the following summaries and estimates:
  + *Ability to run multiple queries simultaneously*, 1 week
  + *Add logging support to the tool*, 3 days
  + *Run overnight jobs on datawarehouse to build reports*, 3 days

Finally the plan in the Hardware project to track any hardware requirements for the tool.

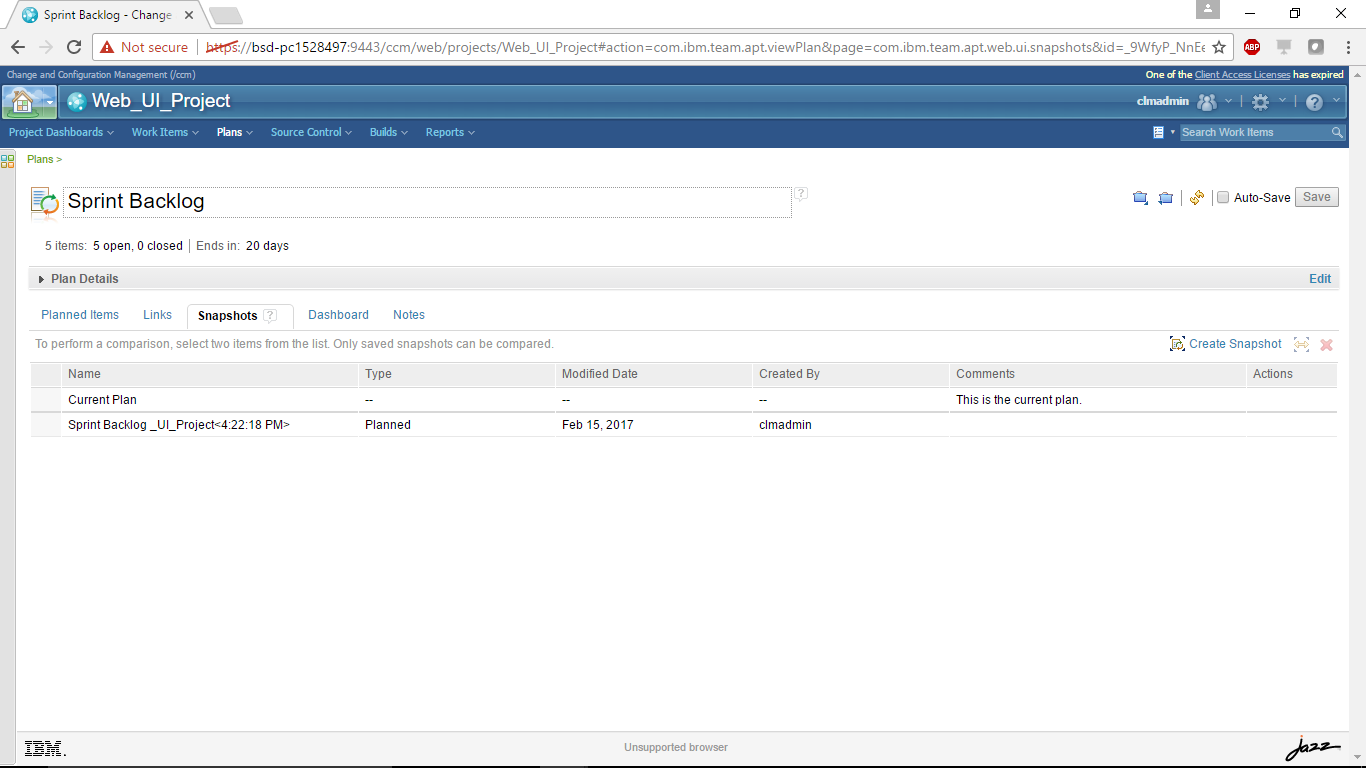


### **Planned Snapshot**

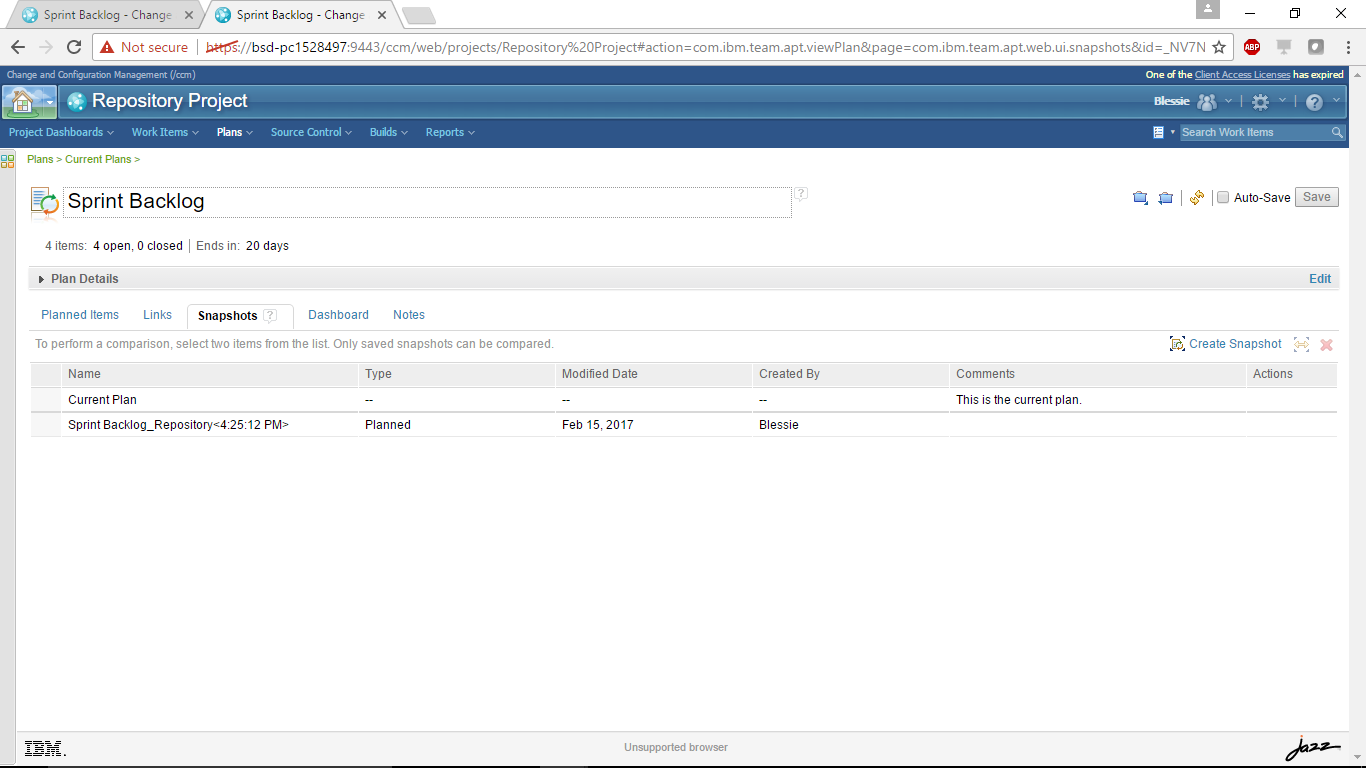
As the development plan progresses, it is important that good practices are in-place for declaring updated plans as planned snapshots by teams/projects; otherwise, the cross-project tracking plan will not be accurate.

The schedule data for the work that is executed in different plans is stored in the planned snapshot that is taken for those plans. If the planned snapshot is not available, the schedule will default to the work item’s iteration start and end date. This makes the schedule roll-up inaccurate; therefore, it is important to save the planned snapshot as the plan progresses.

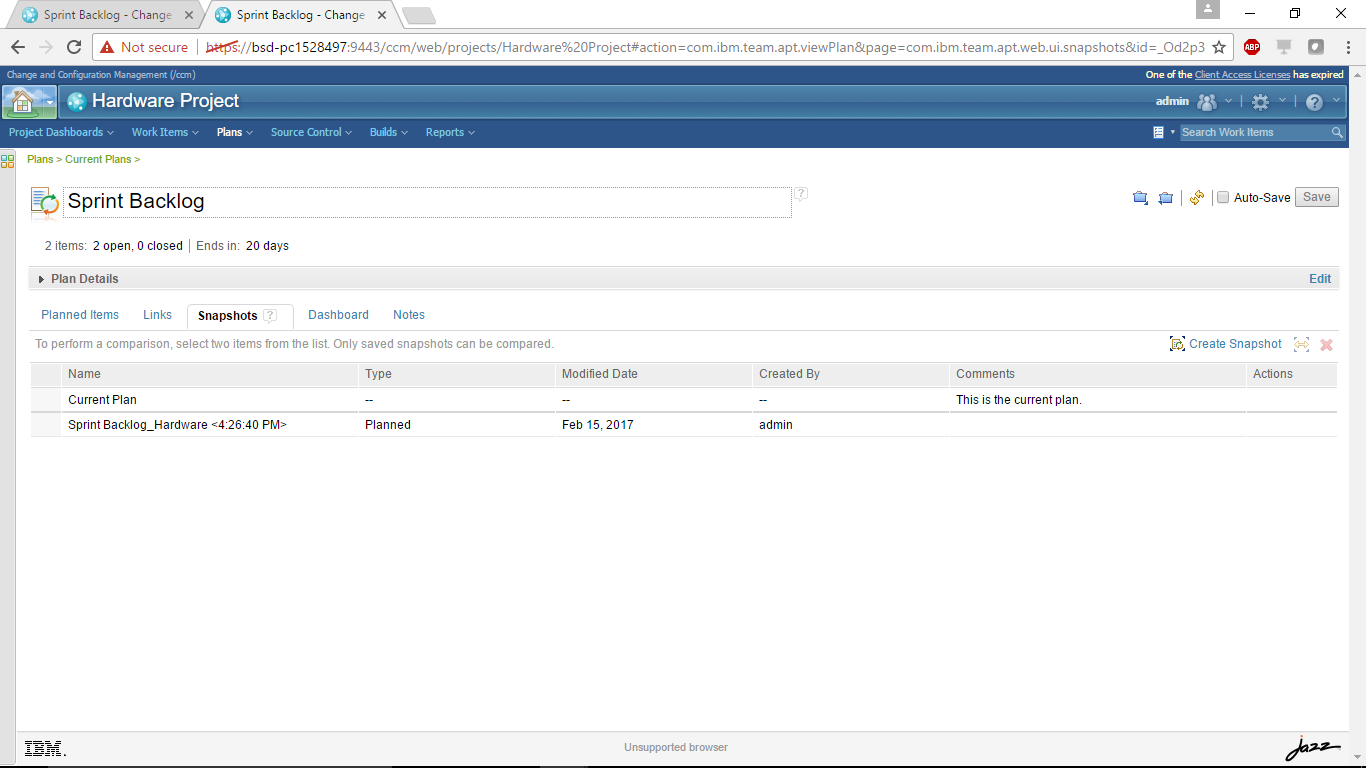
Planned Screenshot of Web\_UI\_Project:



Planned Snapshot of Repository Project:



Planned Snapshot of Hardware Project:



### **Cross-Project Tracking Plan**

The cross-project tracking plan needs to be configured in the project that tracks the work items which are part of another plan that could be in different projects and possibly different repository. A cross-project plan shows all items that belong to it locally (matching the plan query) and the items that are tracked by them.

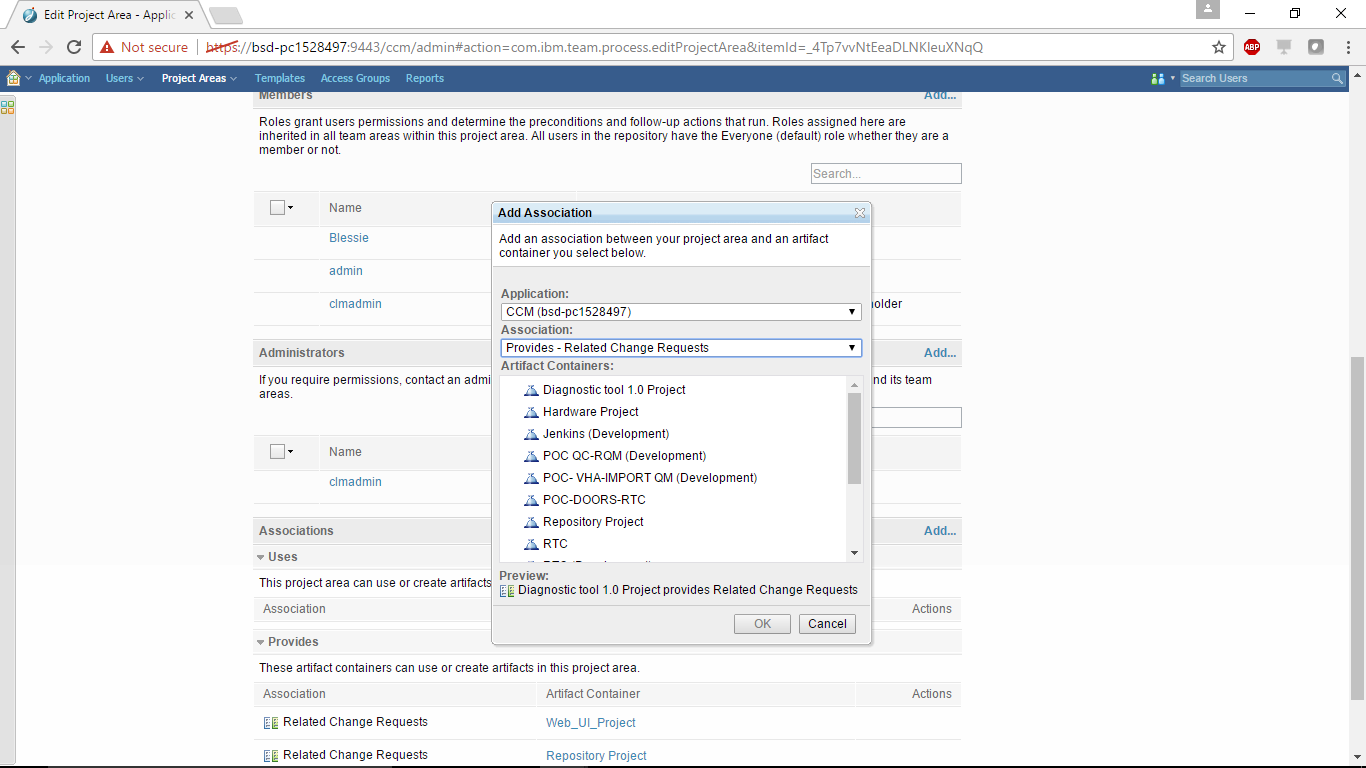
Right now, we only have the three projects (Web UI, Repository, Hardware), but we don’t have a project that “bundles” those three. First, let’s create such a project:

* **Diagnostic tool 1.0 Project**: This is a project to track the plan items across different projects that is building the Diagnostic Tool for a teller machine.

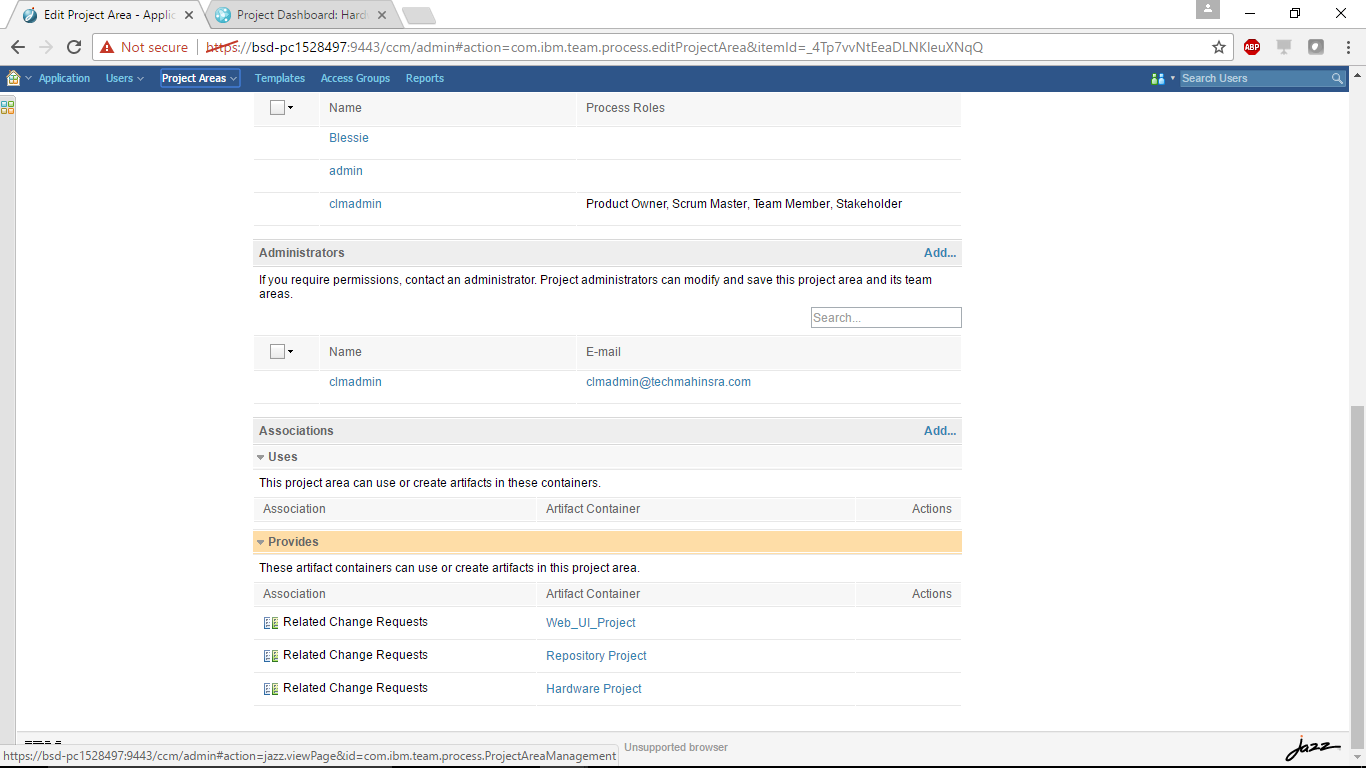
The Diagnostic project can use any process template. You now have to configure associations between the projects.

This association of projects enables you to create a tracks links from the work items in one project area to the work items in another project area.  
**Note**: If the projects exist on different servers, then you must first establish cross-server communication and approve OAuth access requests before performing this procedure.

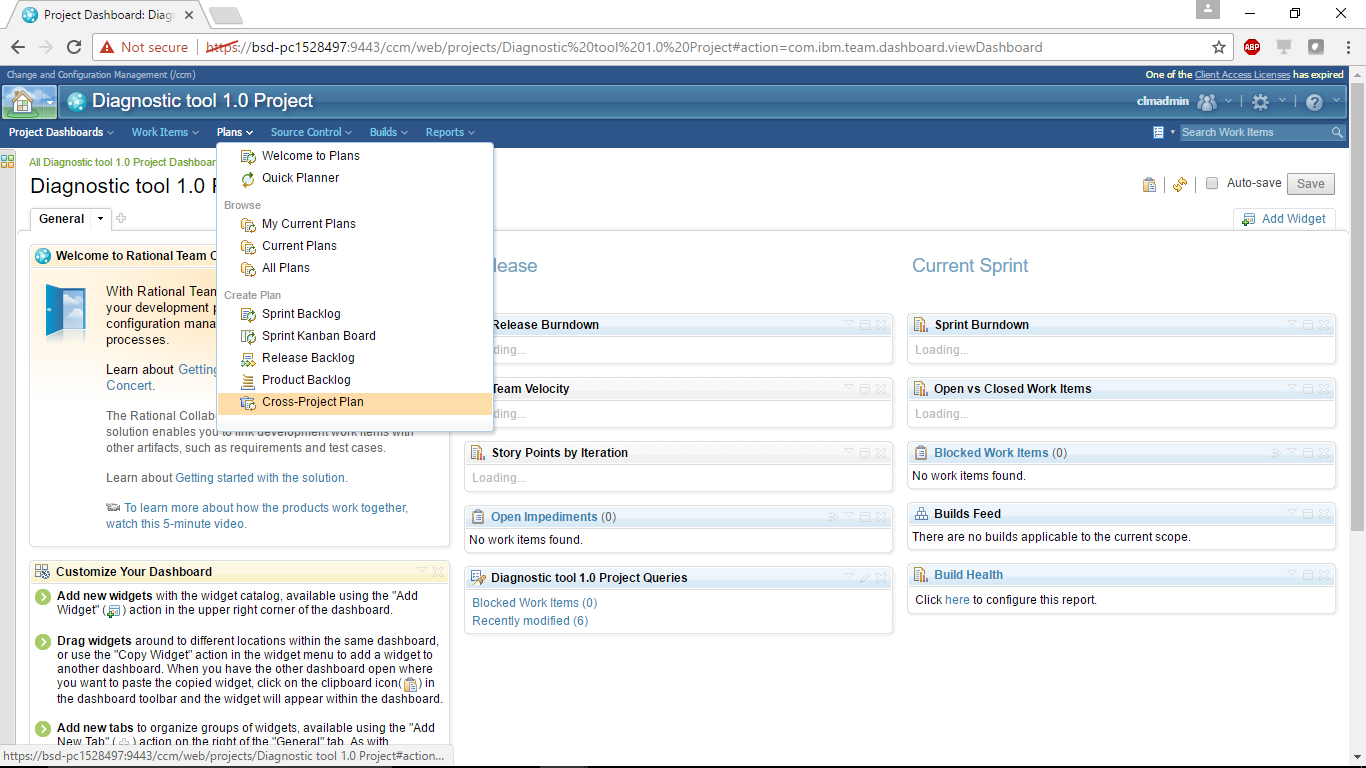
1. Log in to the Jazz server by using an account that has Jazz Administrator privileges.
2. On the Server Administration home page, click **Project Areas** in the Application Administration section.
3. Select **Diagnostic Tool 1.0** project, and scroll to the **Associations**section.
4. Click **Add.**
5. In the *Add Association* window in the Application drop-down list, select a target Jazz application(Here it is CCM of the same server).
6. Entries for the target server project areas are displayed in the **Artifact Containers**field**.** The artifact container corresponds to a project area on the selected server.
7. In the **Association field,** select the association as **Provides – Related Change Requests.**



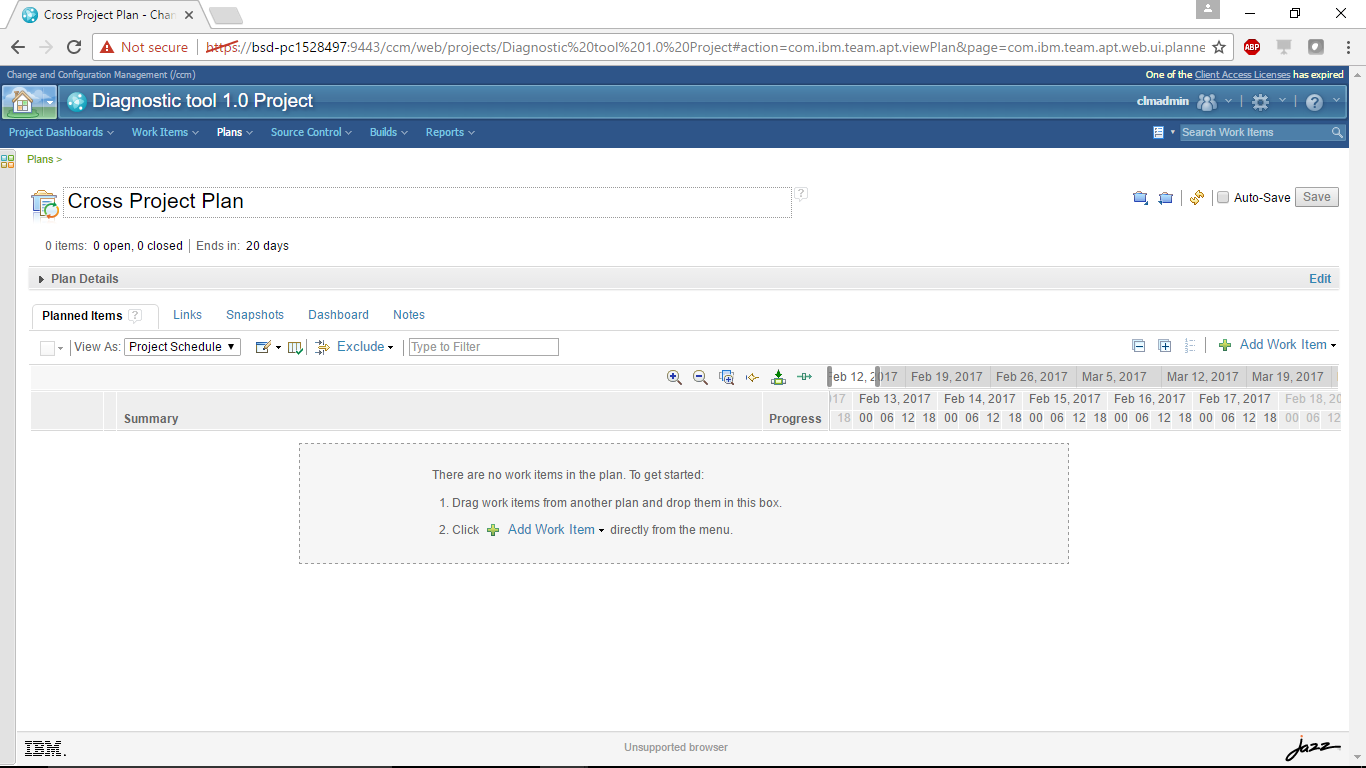
1. In the **Artifact Containers** field, select the **Web UI** project.
2. Click **OK.**
3. Repeat this procedure for the **Repository**and **Hardware** projects. The **Diagnostic Tool 1.0** project’s associations look like the following after the configuration is completed:



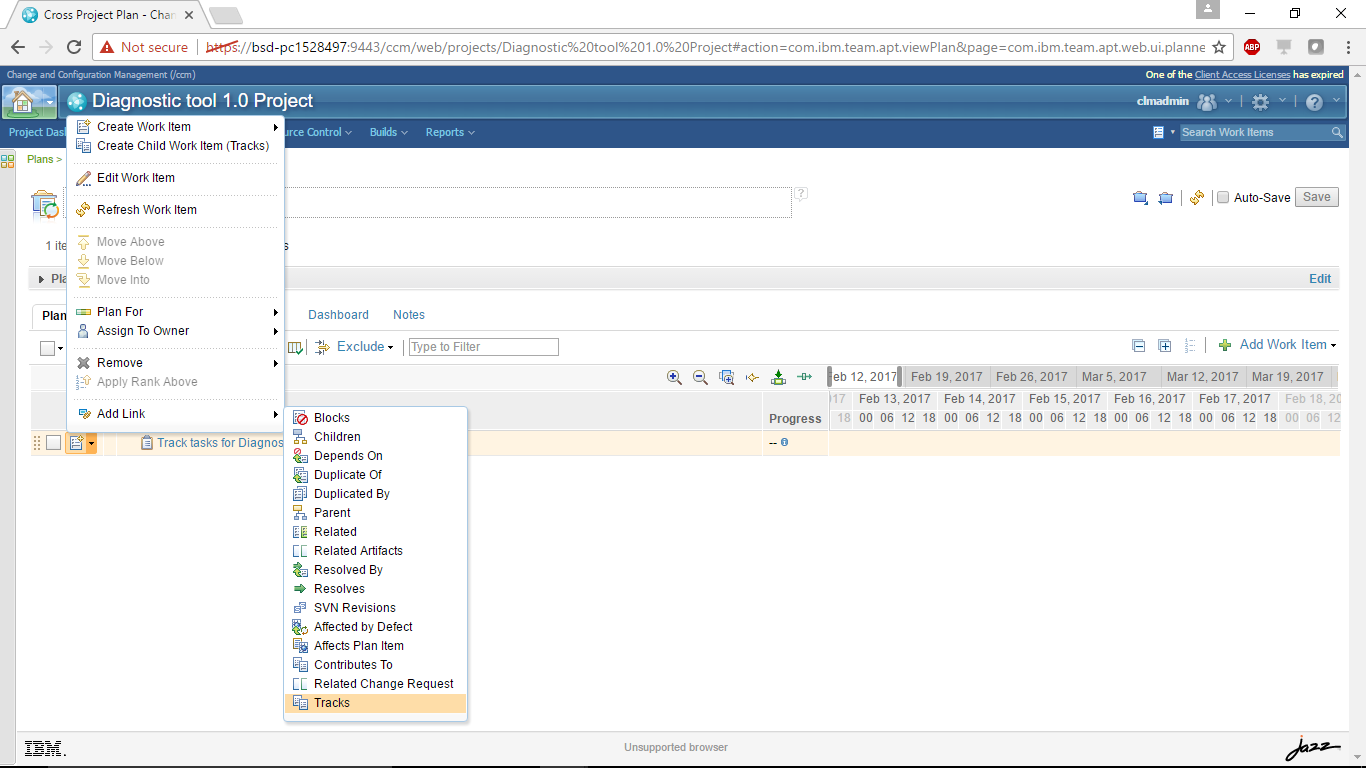
* Open the *Diagnostic tool 1.0* project and create a cross-project plan.



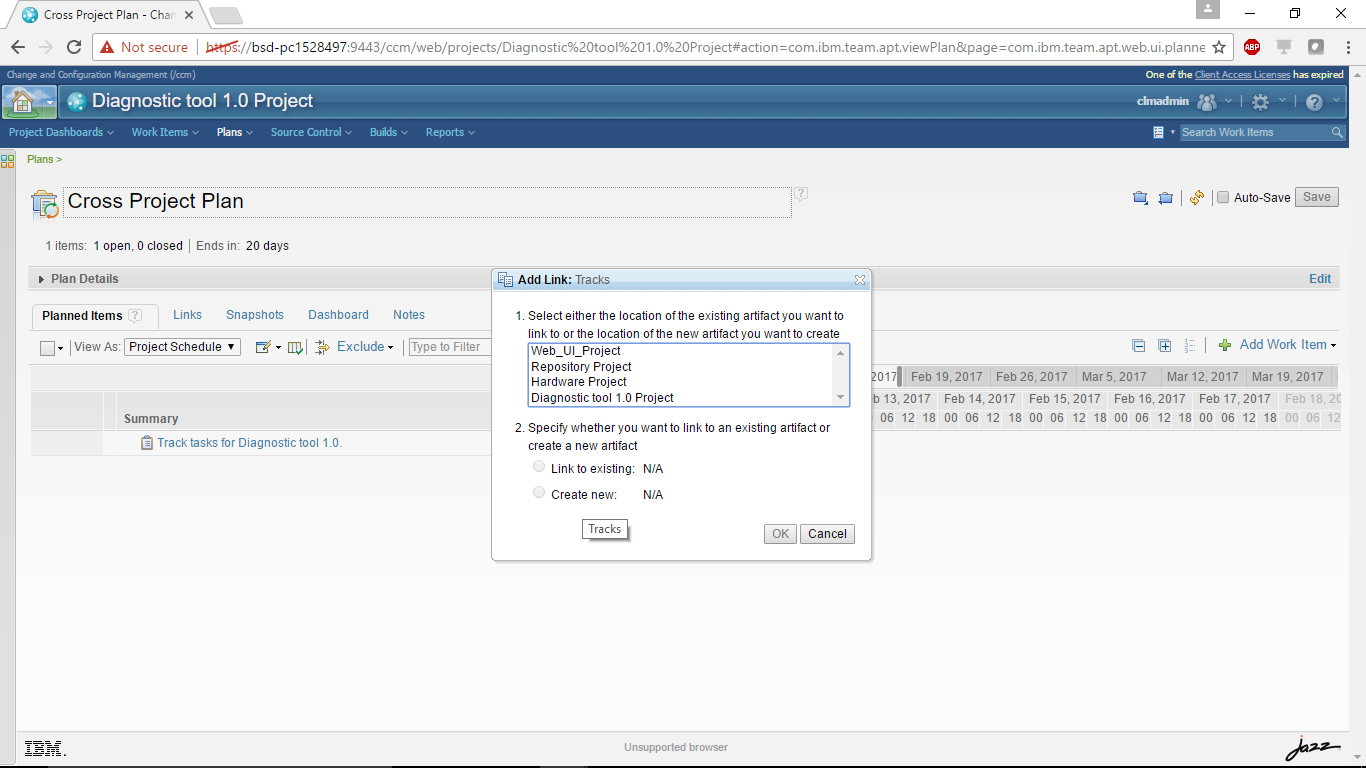
* . The default view for this plan type is the *project schedule*, which includes the *accumulated time* column for viewing a roll-up schedule across items.



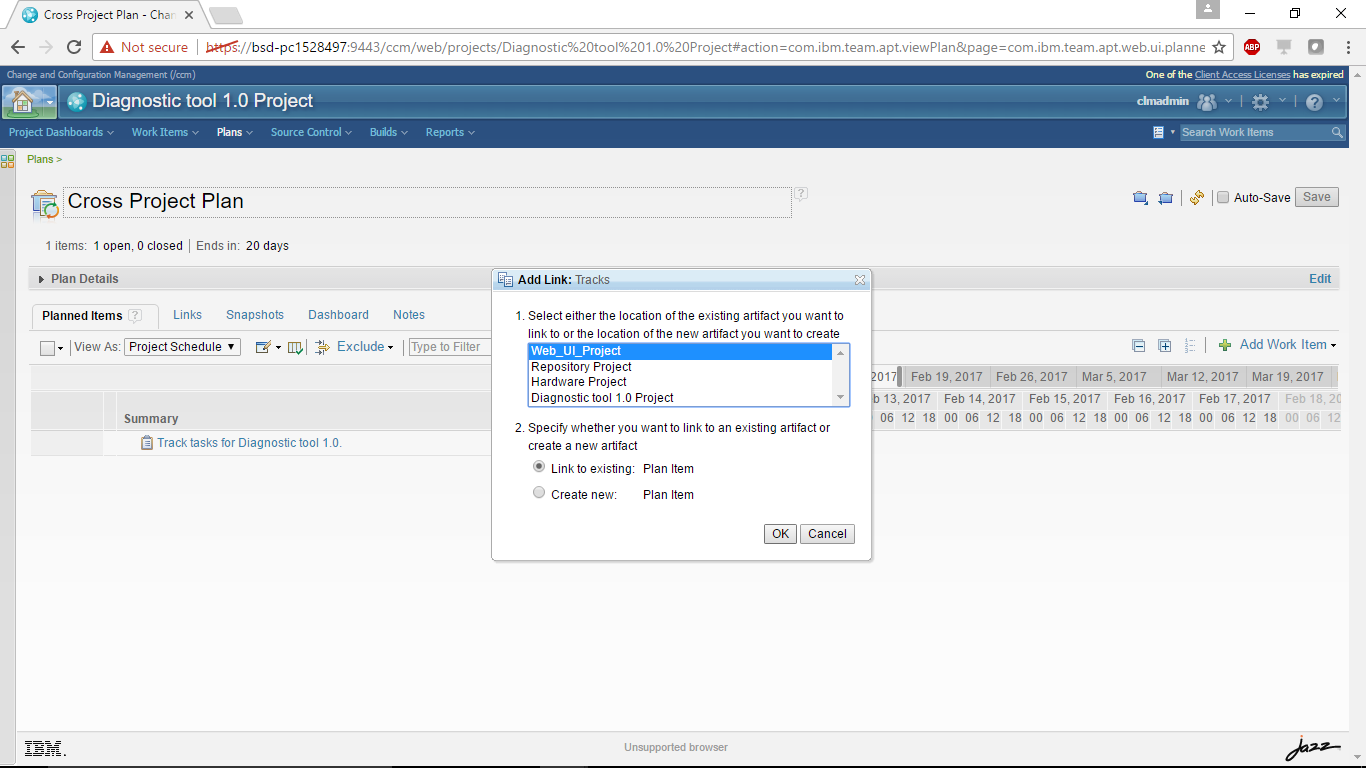
* Create one or more work items to track the work from different plans. For example, create a task work item type with the summary, *Track tasks for Diagnostic tool 1.0*.



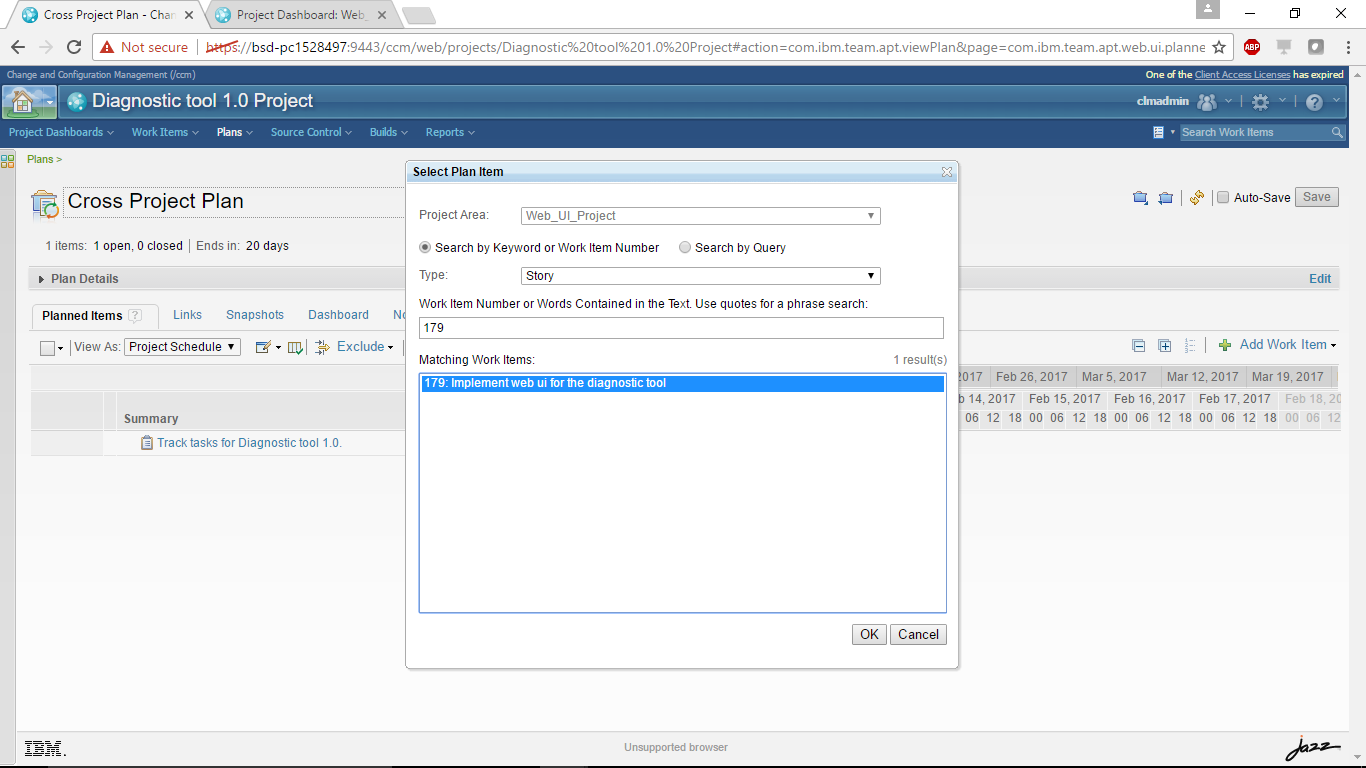
* Add the **Tracks** link to work items in the other projects.



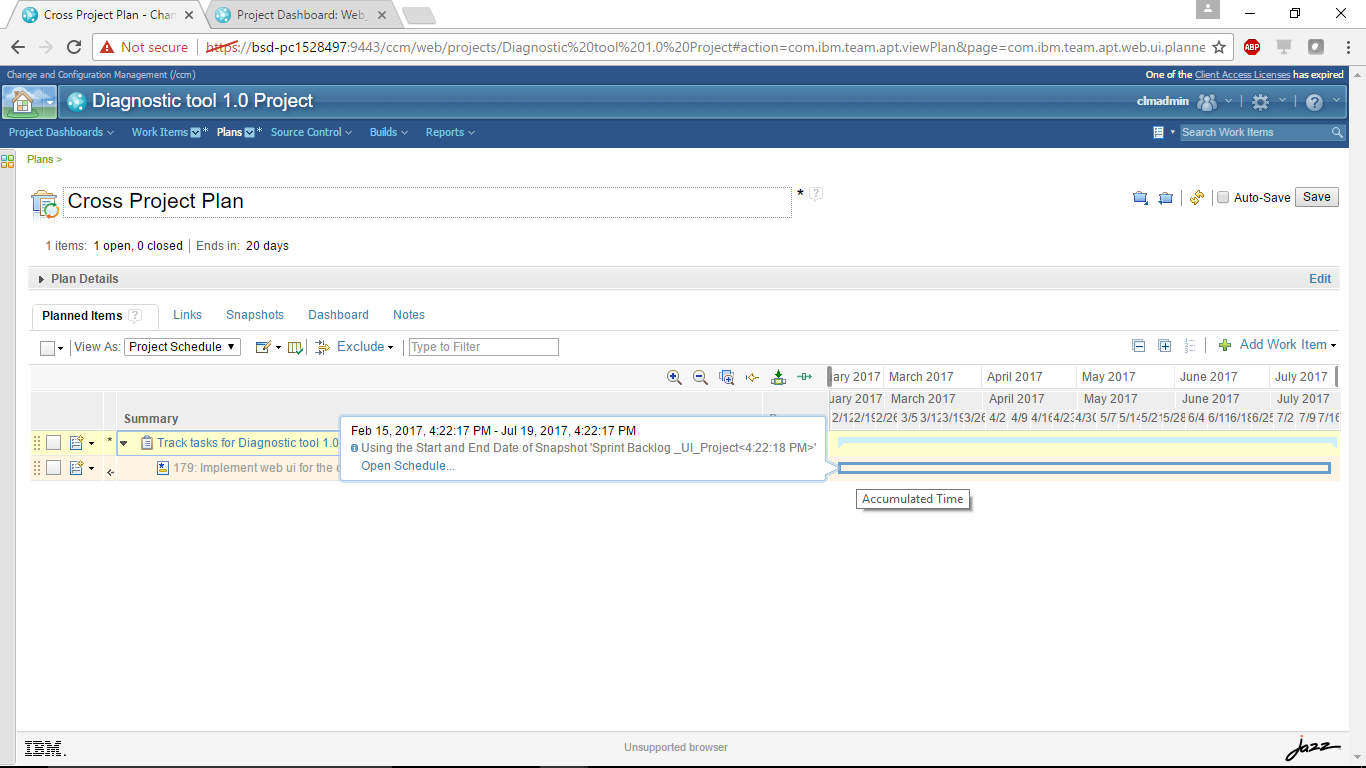
* In the **Add Link: Tracks** window, select the project area of the work item to track, select **Link to existing**, and then click **OK**.



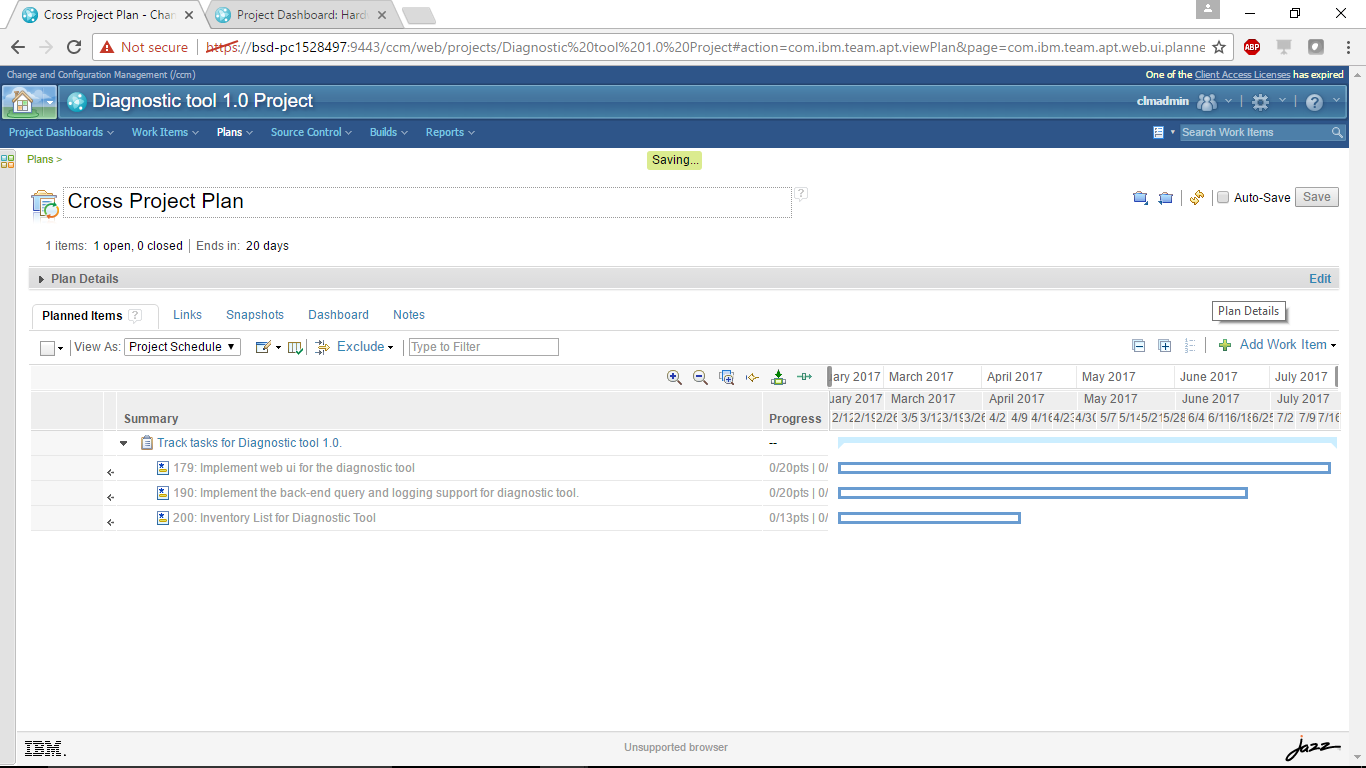
* In the **Select Plan Item** window, select the work items to track.



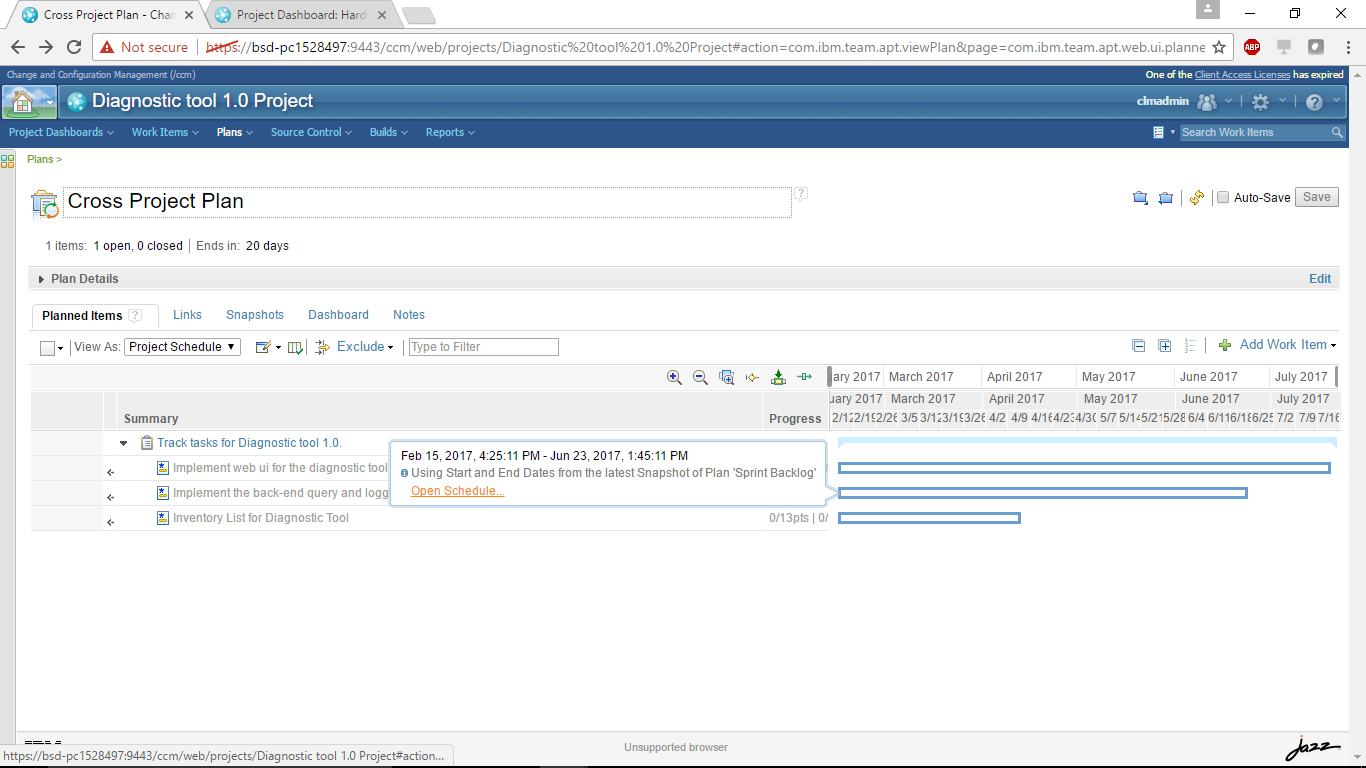
**Caution:** This setup can be cumbersome; therefore, it is better to organize the execution items in the tracked project, so there is a parent plan item that can be used to link here.



* Repeat these steps for **Repository** and **Hardware** projects, and add a link to the plan items to be tracked. This rolled-up view is helpful when you are tracking work that is spread across different projects (on potentially different servers).



* Hover over the bars for the work items in the Gantt Chart. It provides additional details about how the schedule is calculated.

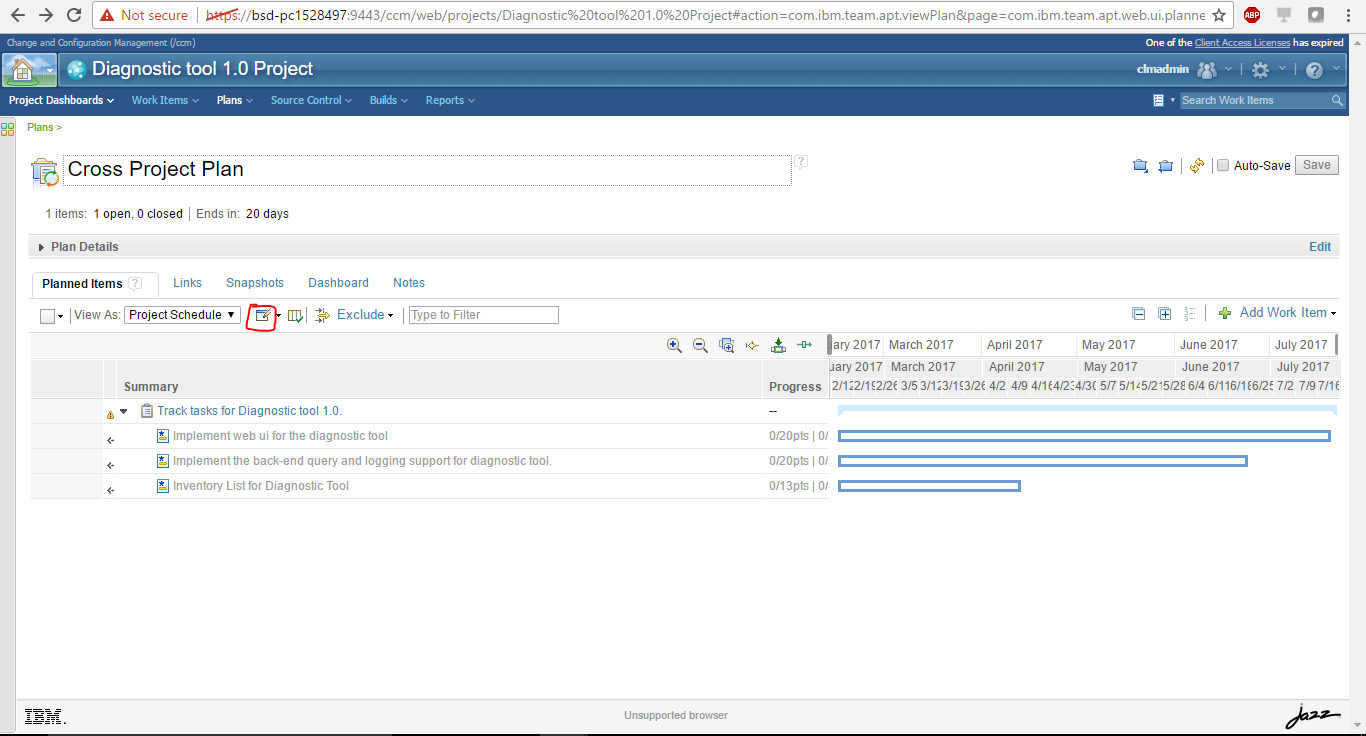


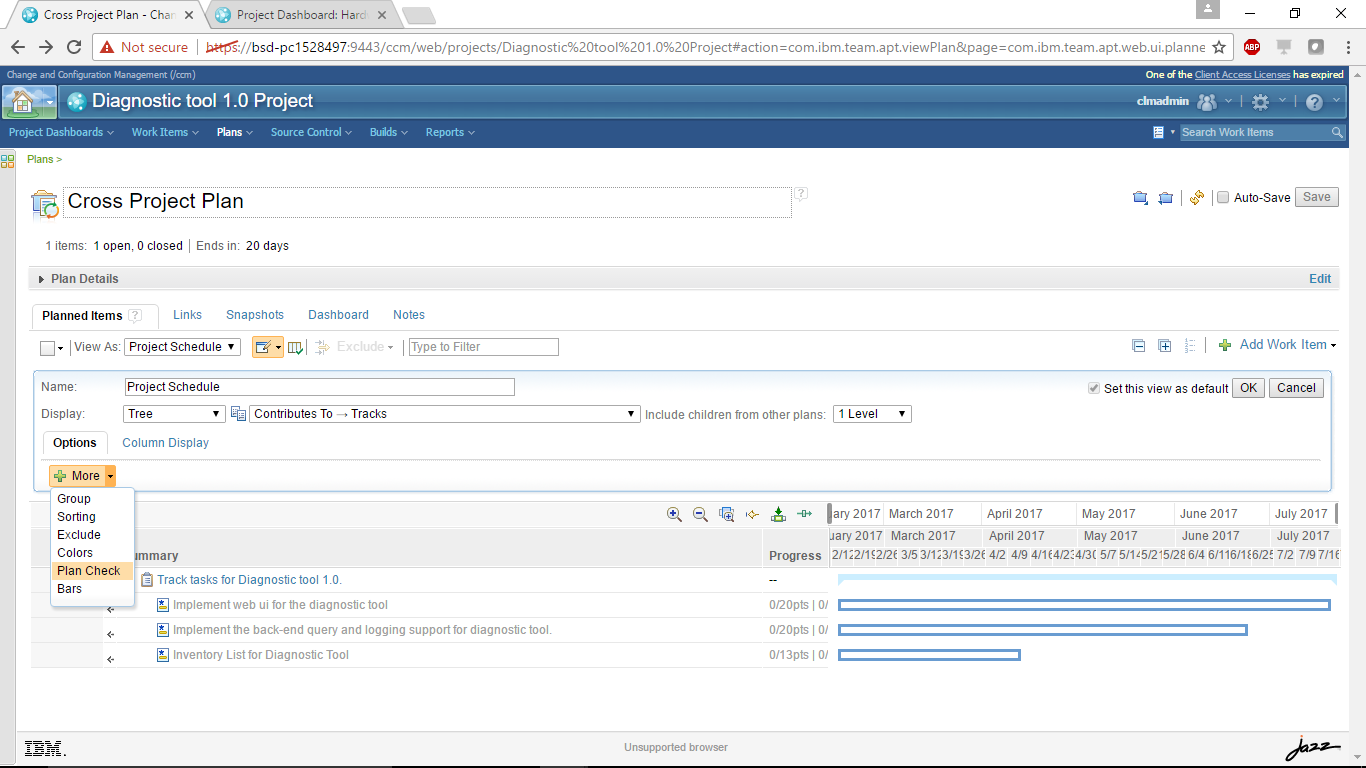
* Click **Open Schedule** to open the plan containing the tracked work item and shows the schedule.   
  **Note:** The schedules are fetched from the planned snapshot of the plans containing these work items. It is important to take these snapshots when the schedule is ready to be rolled up.

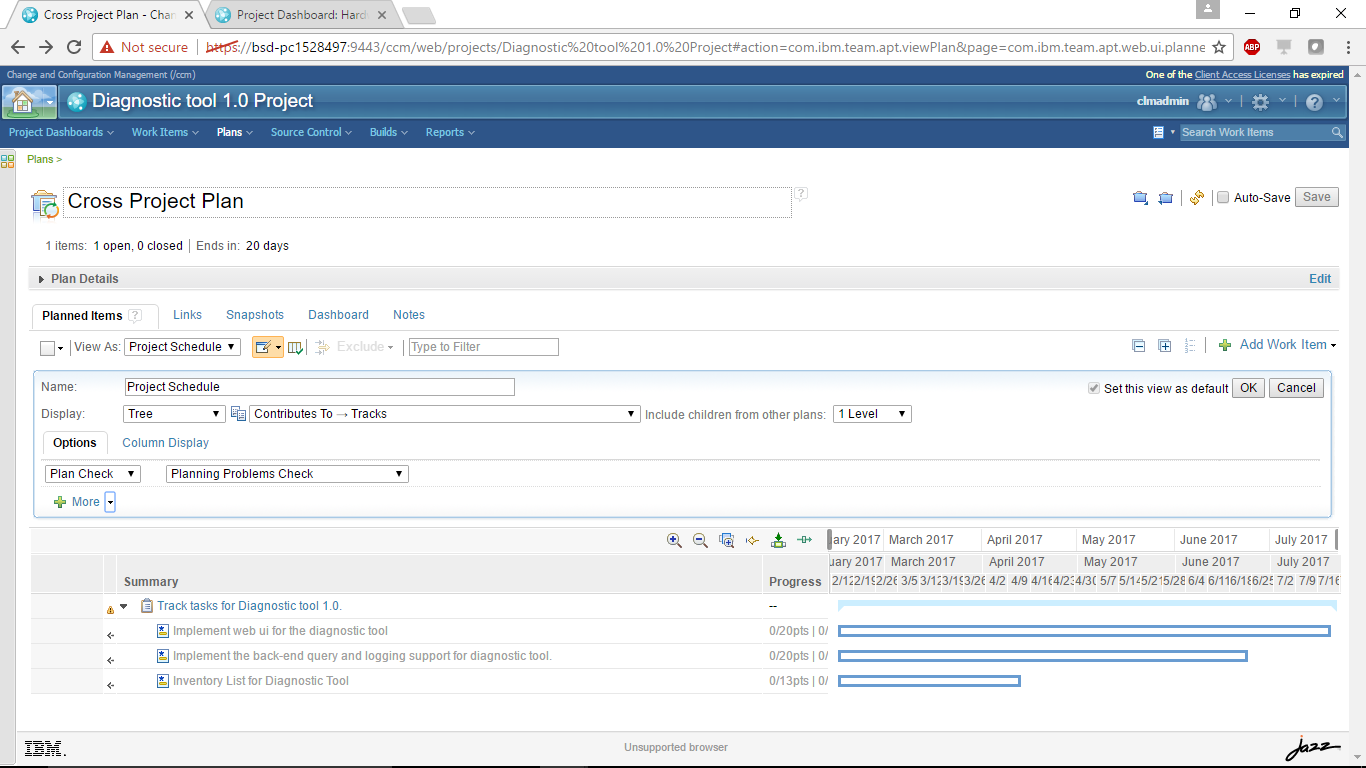
If there are no planned snapshots containing the tracked work items, the iteration start and end dates are used to calculate the schedule by default. In this case, clicking **Open Schedule** opens the *Timelines* tab of the project area editor. The hover, however, does not tell you how old the snapshot is (or when it was made).

### **Assess the health of a plan**

To make sure that the plan is progressing steadily, you can the add the *Cross-Project Planning Problems Check*, which is a new selection in the Plan Check option of a plan view.







It contains three plan checks for cross-project plans that display error messages when any of the following conditions are met:

* 1. The rolled-up schedule for an item on the plan exceeds the end date of the iteration that the plan is associated with.
  2. The “planned for” date of a plan item exceeds the plan’s iteration end date.
  3. The rolled-up schedule for an item on the plan exceeds the due date specified on the item.

### **Date expression support in plans**

Plans now support the ability to colorize or filter items based on a date expression

