



## Using IBM® Rational Team Concert® for complex systems and software development

*Workbook - Using Rational Team Concert® with Rational DOORS® and Rational Rhapsody®*

Version: 2

Last amended: Sept 27, 2011

Author: fraser.chadburn@uk.ibm.com

Status: Ready

IBM Corporation  
Rational software

Using Rational Team Concert with Rational DOORS and Rational Rhapsody for complex systems and software development

Sept 27, 2011

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this documentation in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software  
IBM Corporation  
20 Maguire Road  
Lexington, Massachusetts 02421-3112  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

#### Trademarks and service marks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.html](http://www.ibm.com/legal/copytrade.html)

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Intel trademark information
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both. Microsoft trademark guidelines
- Firefox is a registered trademark of the Mozilla Foundation.
- Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

## Contents

<b>LAB 1</b>	<b>STARTING THE SERVERS.....</b>	<b>8</b>
1.1	STARTING THE IMAGE.....	8
1.2	STARTING THE RTC AND DWA SERVERS.....	9
1.3	VERIFYING THE RTC/DOORS OSLC LOGIN.....	11
<b>LAB 2</b>	<b>DOORS WEB ACCESS.....</b>	<b>14</b>
2.1	LAUNCHING DOORS WEB ACCESS CLIENT.....	14
2.2	EXPLORING TRACEABILITY USING DWA.....	15
2.3	DISCUSSING REQUIREMENTS OVER THE WEB.....	19
<b>LAB 3</b>	<b>CREATING AN IMPLEMENTATION REQUEST FROM DOORS.....</b>	<b>22</b>
3.1	CREATE AN IMPLEMENTATION REQUEST.....	22
<b>LAB 4</b>	<b>USING RTC TO PLAN AND ASSIGN WORK.....</b>	<b>25</b>
4.1	ALLOCATING THE UNASSIGNED WORK.....	25
4.2	CUSTOMIZING THE DASHBOARD BY ADDING A WIDGET.....	29
4.3	EXPLORING THE PLAN VIEWS.....	32
<b>LAB 5</b>	<b>USING RATIONAL RHAPSODY TO PERFORM THE WORK.....</b>	<b>35</b>
5.1	SUSAN PERFORMS SOME WORK AGAINST THE TASK.....	35
5.2	MAKE A CHANGE TO THE RHAPSODY MODEL.....	38
5.3	ATTEMPT TO DELIVER A CHANGE SET.....	41
5.4	USING THE SUBMIT FOR REVIEW FEATURE.....	43
<b>LAB 6</b>	<b>USING RTC TO REVIEW AND APPROVE WORK.....</b>	<b>46</b>
6.1	VIEWING A REVIEW REQUEST.....	46
6.2	REVIEWING CHANGES IN THE CHANGE EXPLORER.....	49
6.3	APPROVING A REVIEW.....	51
<b>LAB 7</b>	<b>DELIVERING THE WORK AND REVIEWING PROJECT STATUS.....</b>	<b>53</b>
7.1	DELIVER THE COMPLETED TASK.....	53
7.2	REVIEW THE FINAL PROJECT STATUS.....	57
<b>APPENDIX A - UPDATING THE ITERATION DATES PRIOR TO LAB 1.....</b>		<b>59</b>
<b>APPENDIX B - CHANGING THE DESKTOP IMAGE.....</b>		<b>65</b>

## Introduction

In this lab we will be exploring the use of Rational Team Concert (RTC) for conducting complex systems and software development. This material can also be used as the basis of a shorten Proof of Technology (PoT) on the use of RTC with Rational DOORS, DOORS Web Access (DWA) and Rational Rhapsody.


The VMware image which accompanies this guide has been built on a Windows 7 64 bit image. This allows the OS to meet the minimum technical specification for the version 3.0 Jazz tooling.

Most laptop and desktop hardware from the past few years is capable of running this image configuration if the configuration steps are followed. This may include changing BIOS settings on your machine. You may also need to call a telephone number to activate the Windows 7 OS.

To run this VMware image you will need:

- VMware Workstation 7.0 ( or higher) or VMware Player 3.1
- 3GB RAM minimum – 4-8 GB preferably.
- To enable your laptop to run 64 Bit OS images.

## Lab 1 Starting the servers



**Lab Scenario**

In this lab we explore the key aspects of Rational Team Concert in the context of directing the work on complex systems and software projects.

You will perform the following tasks:

- Starting the image.
- Starting the RTC and DWA servers.
- Launching DOORS and verifying the connection with RTC.

**Lab Setup**

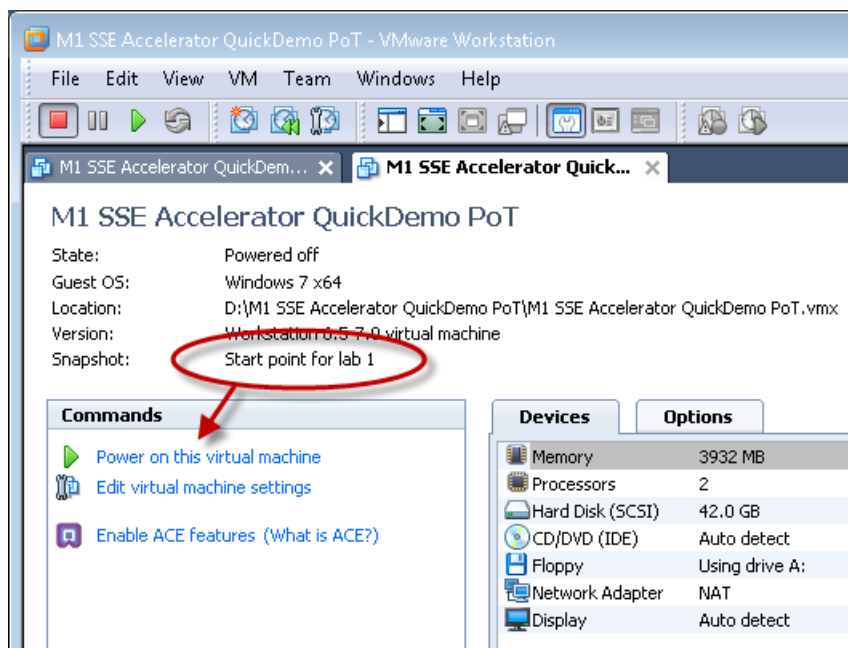
You will need the specially prepared image for the RTC M1 SSE Accelerator ShortDemo PoT.

If you've not updated the iteration dates in the image to reflect the current date then you may need to repeat the steps to do this, otherwise the progress in RTC will not reflect the example screenshots provided in this lab. See Appendix A.

Optionally you can change the Desktop to display an image of the SSE Accelerator tools. See Appendix B.

### 1.1 Starting the image

1. The PoT starts from a shutdown image. Ensure you have the *"Start point for lab 1"* snapshot selected and choose **Power on this virtual machine**.

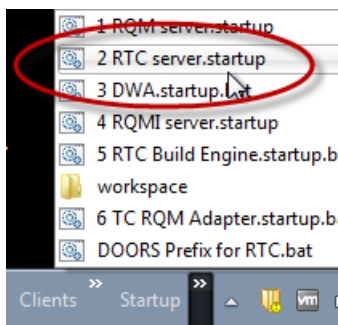


- \_\_2. Login to Windows as **Administrator/passw0rd**.

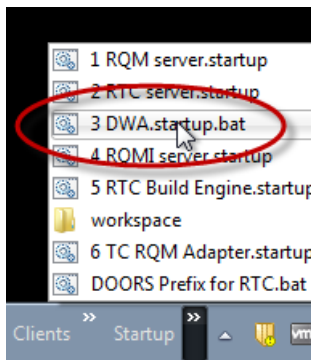


## 1.2 Starting the RTC and DWA servers

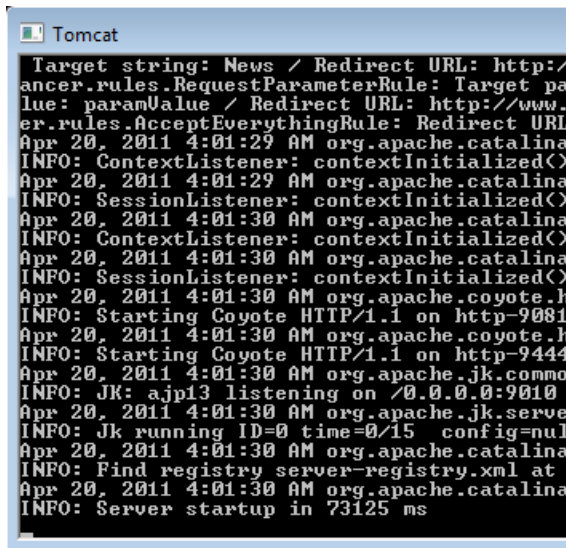
- \_\_1. Start the *RTC server* now by clicking on the **Startup > RTC server.startup** shortcut in the taskbar.



- \_\_2. Also start the *Doors Web Access server* at the same time by clicking on the taskbar **Startup > DWA.startup** shortcut (don't start any of the servers other than DWA and RTC, just yet, though).



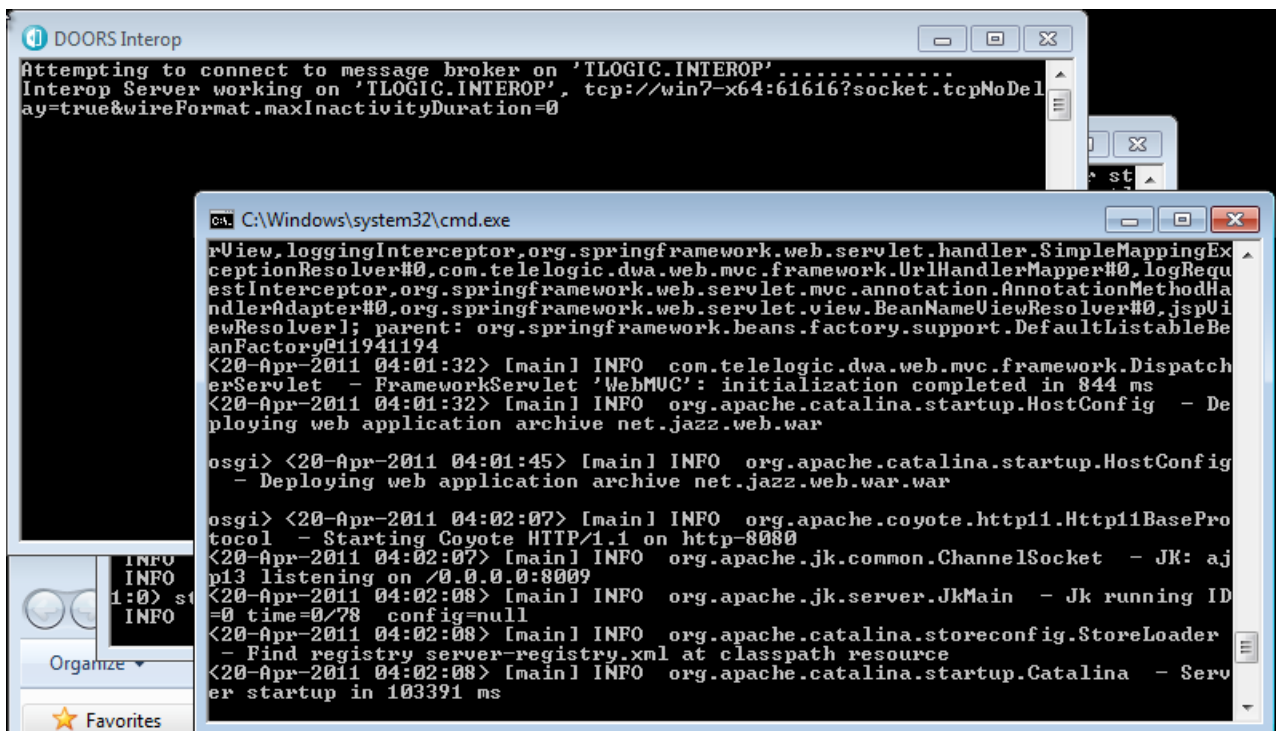
- \_\_\_3. Wait for the RTC server startup message in the Tomcat console window, e.g., "Server startup in 73125 ms".



```

Target string: News / Redirect URL: http://
ancer.rules.RequestParameterRule: Target pa
lue: paramValue / Redirect URL: http://www.
er.rules.AcceptEverythingRule: Redirect URL
Apr 20, 2011 4:01:29 AM org.apache.catalina
INFO: ContextListener: contextInitialized()
Apr 20, 2011 4:01:29 AM org.apache.catalina
INFO: SessionListener: contextInitialized()
Apr 20, 2011 4:01:30 AM org.apache.catalina
INFO: ContextListener: contextInitialized()
Apr 20, 2011 4:01:30 AM org.apache.catalina
INFO: SessionListener: contextInitialized()
Apr 20, 2011 4:01:30 AM org.apache.coyote.h
INFO: Starting Coyote HTTP/1.1 on http-9081
Apr 20, 2011 4:01:30 AM org.apache.coyote.h
INFO: Starting Coyote HTTP/1.1 on http-9444
Apr 20, 2011 4:01:30 AM org.apache.jk.commo
INFO: JK: ajp13 listening on /0.0.0.0:9010
Apr 20, 2011 4:01:30 AM org.apache.jk.serv
INFO: Jk running ID=0 time=0/15 config=nul
Apr 20, 2011 4:01:30 AM org.apache.catalina
INFO: Find registry server-registry.xml at
Apr 20, 2011 4:01:30 AM org.apache.catalina
INFO: Server startup in 73125 ms
  
```

- \_\_\_4. Wait for the *DOORS Web Access* server startup (DWA has x3 console windows).



```

DOORS Interop
Attempting to connect to message broker on 'TLOGIC.INTEROP' .....
Interop Server working on 'TLOGIC.INTEROP', tcp://win7-x64:61616?socket.tcpNoDel
ay=true&wireFormat.maxInactivityDuration=0

C:\Windows\system32\cmd.exe
rView, loggingInterceptor, org.springframework.web.servlet.handler.SimpleMappingEx
ceptionResolver#0, com.telelogic.dwa.web.mvc.framework.UrlHandlerMapper#0, logRequ
estInterceptor, org.springframework.web.servlet.mvc.annotation.AnnotationMethodHa
ndlerAdapter#0, org.springframework.web.servlet.view.BeanNameViewResolver#0, jspVi
ewResolver1; parent: org.springframework.beans.factory.support.DefaultListableBe
anFactory@11941194
<20-Apr-2011 04:01:32> [main] INFO com.telelogic.dwa.web.mvc.framework.Dispatch
erServlet - FrameworkServlet 'WebMVC': initialization completed in 844 ms
<20-Apr-2011 04:01:32> [main] INFO org.apache.catalina.startup.HostConfig - De
ploying web application archive net.jazz.web.war

osgi> <20-Apr-2011 04:01:45> [main] INFO org.apache.catalina.startup.HostConfig
- Deploying web application archive net.jazz.web.war.war

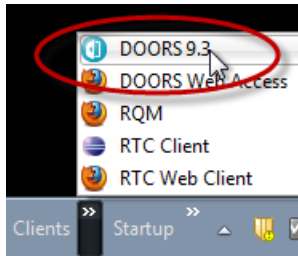
osgi> <20-Apr-2011 04:02:07> [main] INFO org.apache.coyote.http11.Http11BasePro
tocol - Starting Coyote HTTP/1.1 on http-8080
<20-Apr-2011 04:02:07> [main] INFO org.apache.jk.common.ChannelSocket - JK: aj
p13 listening on /0.0.0.0:8009
<20-Apr-2011 04:02:08> [main] INFO org.apache.jk.server.JkMain - Jk running ID
=0 time=0/78 config=null
<20-Apr-2011 04:02:08> [main] INFO org.apache.catalina.storeconfig.StoreLoader
- Find registry server-registry.xml at classpath resource
<20-Apr-2011 04:02:08> [main] INFO org.apache.catalina.startup.Catalina - Serv
er startup in 103391 ms
  
```

- \_\_\_5. Minimize but don't close the x4 console windows.

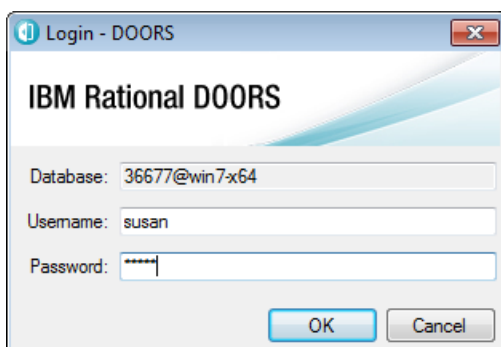


### 1.3 Verifying the RTC/DOORS OSLC login

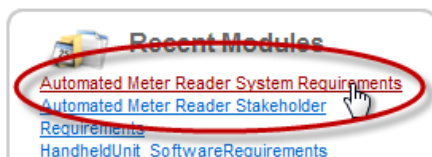
- \_\_1. You should also start the DOORS rich client ahead of the demo by choosing the **Clients > DOORS 9.3** shortcut in the taskbar. This will allow quick navigation from DOORS Web Access to the full client. It will also verify that the DOORS/RTC integration is working.



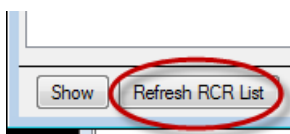
- \_\_2. Login as **susan/susan**, the requirements engineer.



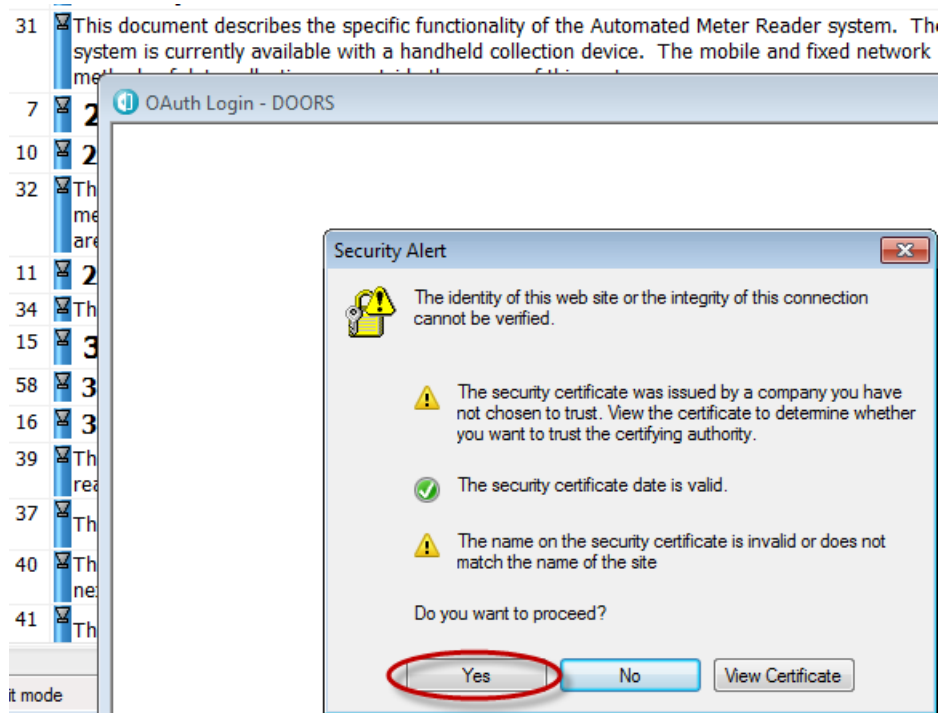
- \_\_3. Use the *Recent Modules* panel in the Welcome screen to navigate to the **Automated Meter Reader System Requirements**.



Because formal change management is enabled for the systems module a dialog is launched to give you the option of choosing how to open the module. At this point you should be presented with a dialog asking you to log into RTC. If not, hit the Refresh RCR List to start the connection.



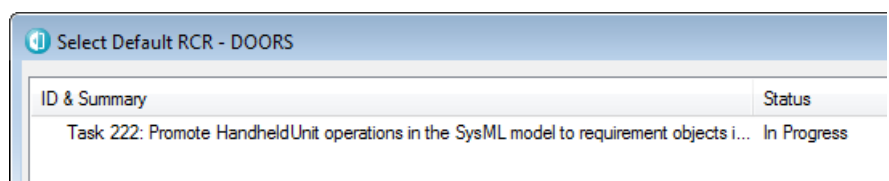
\_\_4. Click **Yes** to proceed.



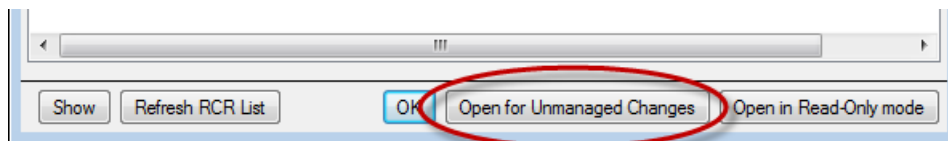
\_\_5. Login to RTC as **susan/susan**.



\_\_6. Verify that Task 222 is shown in the *Select Default RCR - DOORS* dialog.

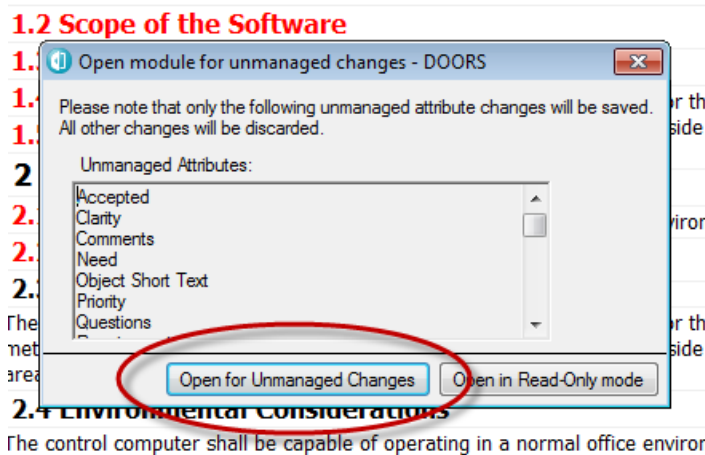


- \_\_\_7. Because the module is under formal change management you now have three options:
- \_\_\_a. To open the module against a Requirement Change Request (RCR).
  - \_\_\_b. Or, to open in read-only mode - This prevents any changes being made.
  - \_\_\_c. Or, to open for unmanaged changes - - This limits changes to only those that don't require a formal change request (it is possible to specify which attributes are formally managed in the change management configuration).
- \_\_\_8. Choose **Open for Unmanaged Changed**, this will enable you to add an implementation request without needing a *Requirement Change Request* (RCR).




Note that if you chose the wrong option you should close and reopen the module for unmanaged changes before proceeding otherwise the *Implementation Request* submenu may not work as expected later on.

- \_\_\_9. Another dialog is launched specifying which attributes you can change. Click **Open for Unmanaged Changes** again to confirm that you want to proceed (!).




- \_\_\_10. You're now ready for the main part (keep DOORS client running for now).

## Lab 2 DOORS Web Access

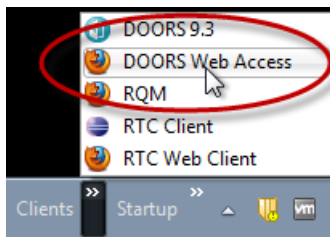
	<p><b>Lab Scenario</b> In this lab we explore the DOORS Web Access client.</p> <p>You will perform the following tasks:</p> <ul style="list-style-type: none"> <li>• Exploring requirements using the DOORS Web Access client.</li> <li>• Using the discussion feature to comment on requirements.</li> <li>• Navigating to the DOORS client using requirements URLs.</li> </ul> <p><b>Lab Setup</b> This lab requires completion of Lab 1.</p>
---	---

### 2.1 Launching DOORS Web Access client

- \_\_1. Let's start by looking at *DOORS Web Access*. This highlights IBM strategic shift to use of web technology to reduce deployment costs and to support multi-site teams.

	<p><b>Benefit</b></p> <p>Note that as well as reducing deployment costs, the use of thin clients also means that you increase the range of stakeholders interacting with requirements. Research has shown that active collaboration of stakeholders is one of the critical factors to project success.</p>
---	--

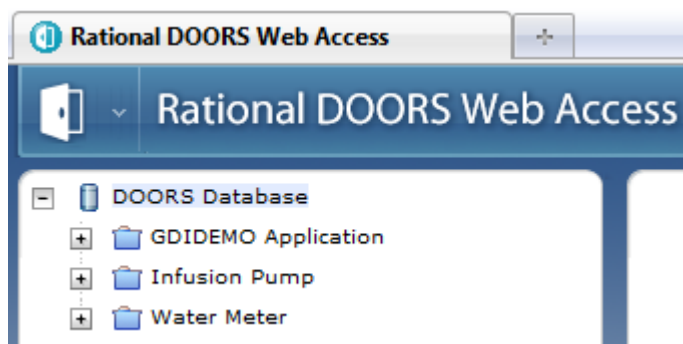
- \_\_2. Launch the Firefox browser by clicking **Clients > DOORS Web Access** in the taskbar.



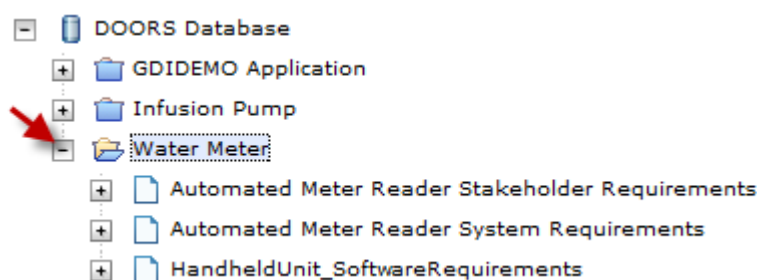
- \_\_3. Login to the DOORS ERS Repository as **susan/susan**, the requirements engineer.



- \_\_4. Note that there are three projects in the database.




- \_\_5. Expand the *Water Meter* project. Note that there are three documents (known as formal modules).

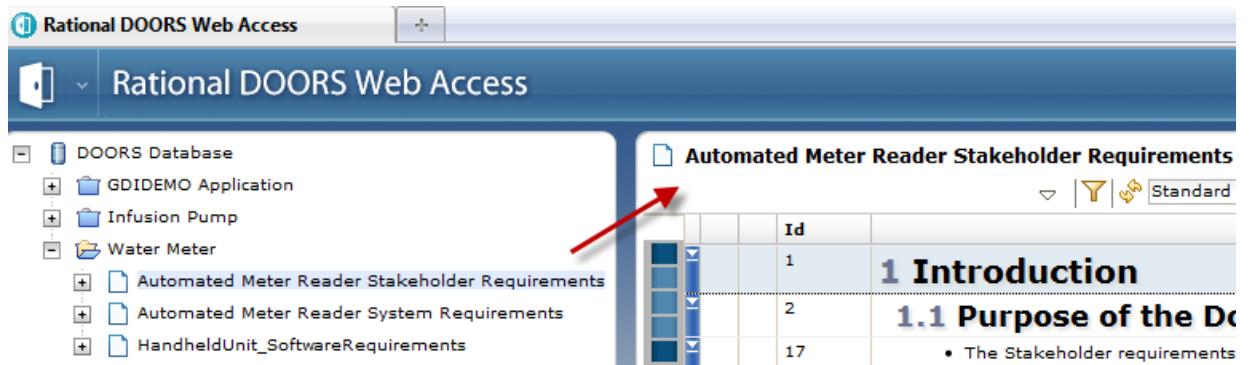


## 2.2 Exploring traceability using DWA

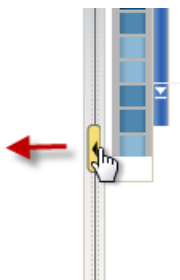
- \_\_1. Essentially this project shows a hierarchy of requirement levels. Stakeholder requirements are elaborated into a systems requirements and some of the system requirements have been elaborated into software requirements for a handheld device.

	<p><b>Benefit</b></p> <p>Layering requirements at different levels of abstraction is one of the best practices of requirements management. Essentially the requirements at different levels are owned by different teams.</p>
---	---

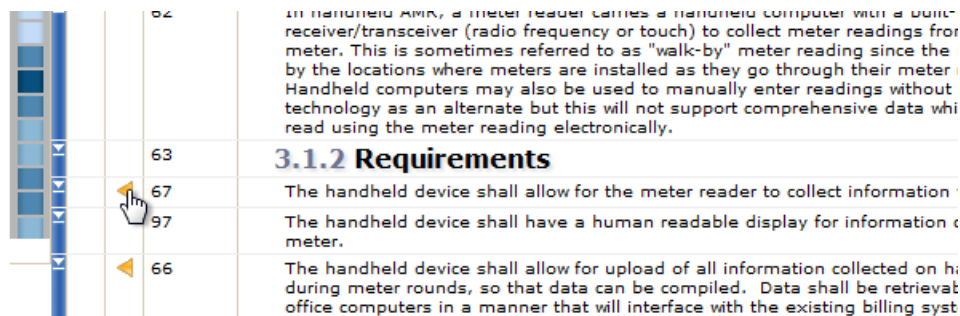
\_\_2. Double-click to open up the *Automated Meter Reader Stakeholder Requirements* module.



\_\_3. Let's explore the traceability a little. Click on the separator button to close the browser, so that we can see the requirement text in more detail.



\_\_4. Scroll down to the first traceability link associated with the stakeholder requirements.

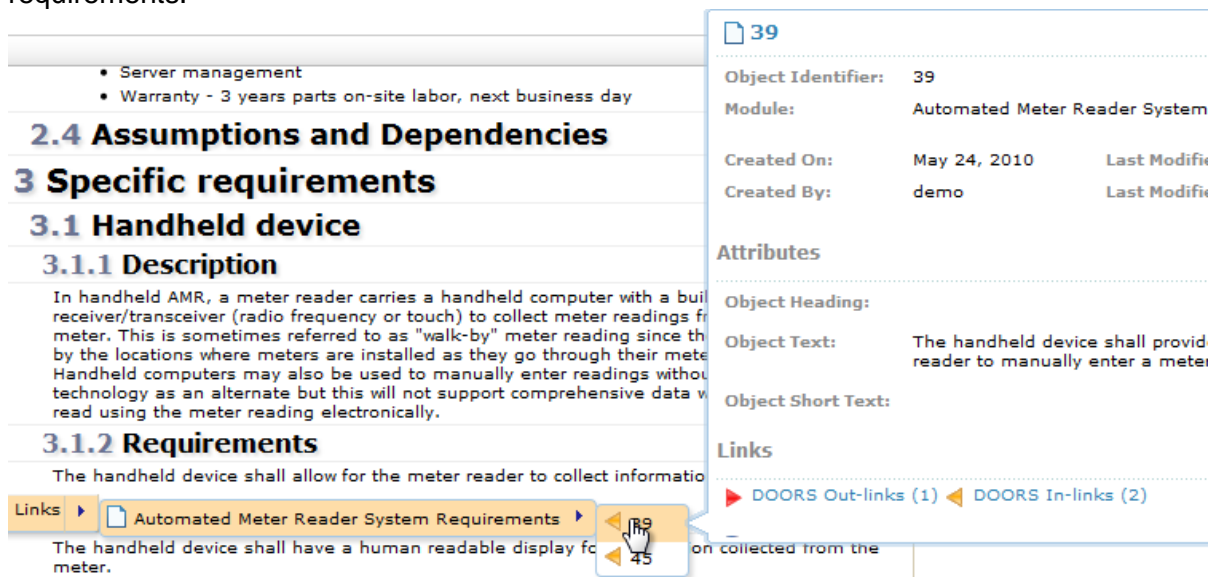


\_\_5. This requirement says that "The handheld device shall allow for the meter reader to collection information from the meter".

\_\_6. Left-click the link icon to expand the DOORS links.



\_\_7. Use hover-over to show how the stakeholder requirement is met by the two system level requirements.



• Server management  
• Warranty - 3 years parts on-site labor, next business day

## 2.4 Assumptions and Dependencies

## 3 Specific requirements

### 3.1 Handheld device

#### 3.1.1 Description

In handheld AMR, a meter reader carries a handheld computer with a built-in receiver/transceiver (radio frequency or touch) to collect meter readings from meters. This is sometimes referred to as "walk-by" meter reading since the meter reader is by the locations where meters are installed as they go through their meter readings. Handheld computers may also be used to manually enter readings without the use of technology as an alternate but this will not support comprehensive data validation. The handheld device shall allow for the meter reader to collect information from the meter.

#### 3.1.2 Requirements


The handheld device shall have a human readable display for the meter reader to read using the meter reading electronically.

Links: Automated Meter Reader System Requirements

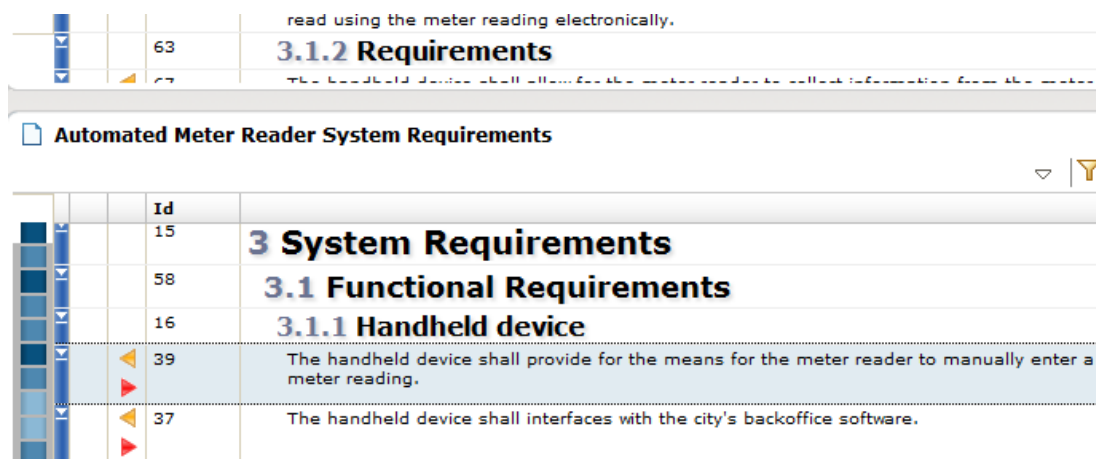
Object Identifier: 39  
Module: Automated Meter Reader System  
Created On: May 24, 2010  
Created By: demo

Attributes  
Object Heading:  
Object Text: The handheld device shall provide for the means for the meter reader to manually enter a meter reading.  
Object Short Text:

Links  
DOORS Out-links (1) DOORS In-links (2)

	<p><b>Benefit</b></p> <p>DOORS web access has some nice usability features, increasing your ability to engage stakeholders from disparate fields. Importantly, you don't have to navigate the link to see what the system requirements are, you can use the concept of rich hovers in the web browser.</p>
---	--

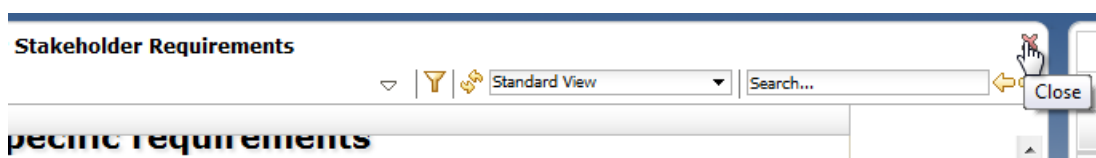
\_\_8. Click to navigate to first system requirement 39. Note that DWA opens up a new window showing the *Automated Meter Reader System Requirements*.



Id	Requirement
15	3 System Requirements
58	3.1 Functional Requirements
16	3.1.1 Handheld device
39	The handheld device shall provide for the means for the meter reader to manually enter a meter reading.
37	The handheld device shall interfaces with the city's backoffice software.

Automated Meter Reader System Requirements

\_\_9. Close the "stakeholder requirements" window to make room for the newly opened "system requirements" module.



Stakeholder Requirements

Standard View Search... Close

\_\_\_10. Using the hover-over feature we can see here that this particular system requirement traces to a number of software requirements in the *HandheldUnit\_SoftwareRequirements* module.

**nated Meter Reader System Requirements**

The screenshot shows a table of requirements with columns for 'Id' and text. Requirement 121 is highlighted, and a tooltip is visible. The tooltip shows details for 'completeMeterRead' including Object Identifier (121), Module (HandheldUnit\_SoftwareRequirements), Created On (December 6, 2010), and Object Text (The Handheld Unit shall have the abi data required for route). It also shows 'DOORS Out-links (3)' and 'DOORS In-links (0)'. A red arrow points to the 'DOORS Links' dropdown menu in the table, which is open and shows 'HandheldUnit\_SoftwareRequirements' selected.

\_\_\_11. We can actually use a view to show this traceability. Select the **Traceability** view in the pull-down list.

The screenshot shows a dropdown menu with 'Traceability' selected. Other options include Engineering Planning, IR Attrs View, RCR Attrs View, RQM Test Results View, RQM Test Trace View, Requirements Engineer, Requirements View, Trace and Impact, and Standard View. A red arrow points to the 'Traceability' option.

\_\_\_12. Notice that this view has a derived column showing which requirements trace downstream.

**Reader System Requirements**

System requirements for the AMR system	Traceability
<b>3 System Requirements</b>	
<b>3.1 Functional Requirements</b>	
<b>3.1.1 Handheld device</b>	
The handheld device shall provide for the means for the meter reader to manually enter a meter reading.	<b>/Water Meter/HandheldUnit_SoftwareReq</b> Object 121. Object Heading: completeMeterRead
The handheld device shall interfaces with the city's backoffice software.	<b>/Water Meter/HandheldUnit_SoftwareReq</b> Object 150. Object Heading: selectOperation
The handheld software device shall allow for programming of a defined route, advancing to the next meter on the route as the meter reader moves through the route.	<b>/Water Meter/HandheldUnit_SoftwareReq</b> Object 114.

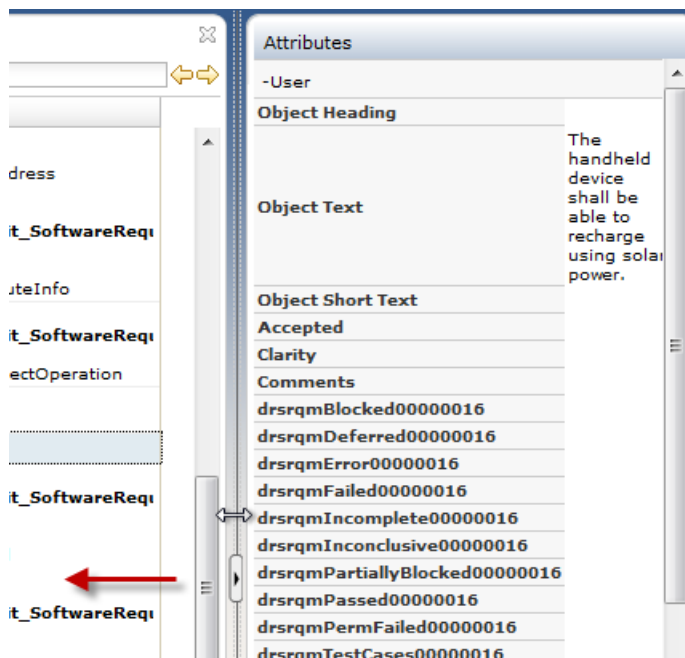


- \_\_\_13. A number of requirements do not have any downstream traceability. Scroll down and look for the "solar power" requirement (number 46) - we can find this more easily by looking for requirements that don't have downstream traceability. Click on the requirement so that it is highlighted.

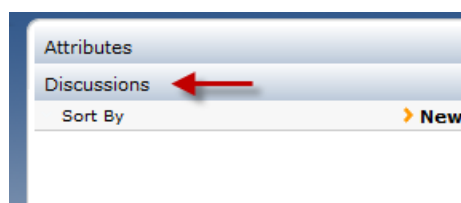
43	The handheld device shall allow the meter reader to enter information about meters relocated on a particular route.	/Water Meter/Handheld Object 150. Object Heading:
44	Information captured via the handheld device shall be downloadable via either cable hookup or wireless signal.	
46	The handheld device shall be able to recharge using solar power.	
77	The handheld device shall provide a means to automatically (electronically) read the meter.	/Water Meter/Handheld Object 121. Object Heading: completeMeterR

## 2.3 Discussing requirements over the web

- \_\_\_1. Expand the right-hand panel a bit. Let's explore this in more detail.

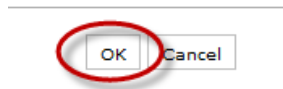


- \_\_\_2. Click on the **Discussions** view to open it up.

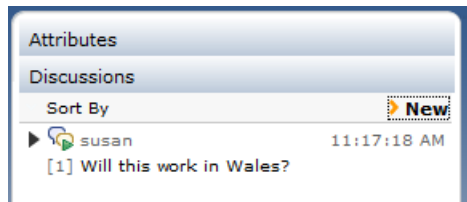



- \_\_3. Click **New** to add a new discussion, e.g., 'Will this work in Wales?', 'Is there a risk that areas with high rainfall will not have enough sunlight?'

- \_\_4. Click **OK** to save it.

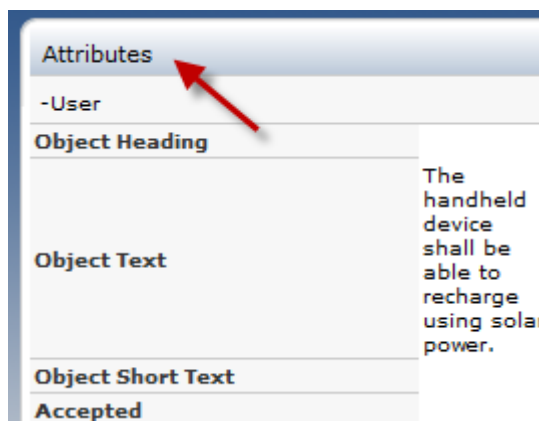


- \_\_5. Note that comment is recorded with the user name and time.

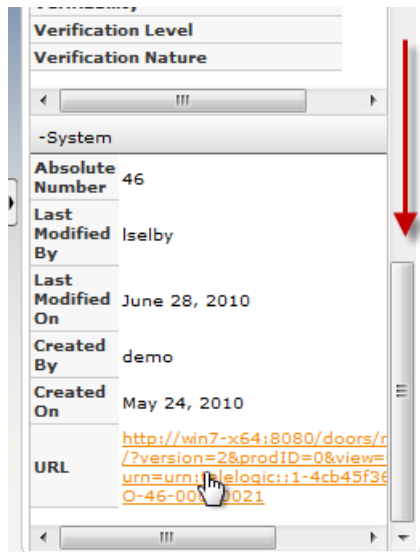



	<p><b>Benefit</b></p> <p>Using this feature you can improve the quality of the requirements through discussion and commenting, reducing the risk of implementing the wrong system or incurring additional cost by implementing features not needed.</p>
---	---

- \_\_6. Click back to the **Attributes** now.

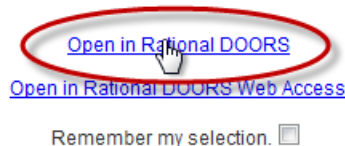


- \_\_7. Scroll down to find the URL. Notice that all requirements have a URL. Click on the URL to open up a navigation window for it.

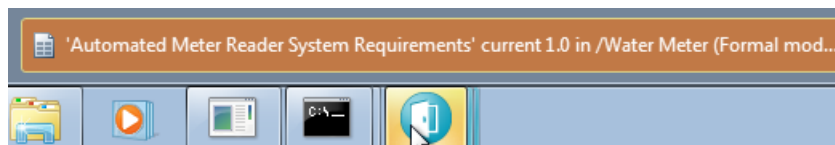


	<b>Benefit</b>
<p>The ability to view data in a web browser and navigate via URLs is an important feature as it means that you can share information between tools, and also send URLs to people to point out information that is relevant.</p>	

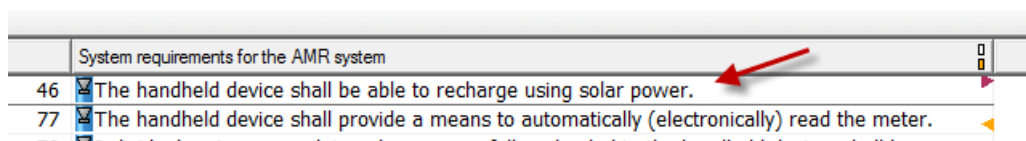
- \_\_8. Choose **Open in Rational DOORS** to navigate to the thick client.



- \_\_9. Systems Requirements module should be highlighted in the task bar. Click on the *Automated Meter Reader System Requirements* module to give it the focus.




- \_\_10. Notice that the same requirement is highlighted in DOORS full client (related to solar power).



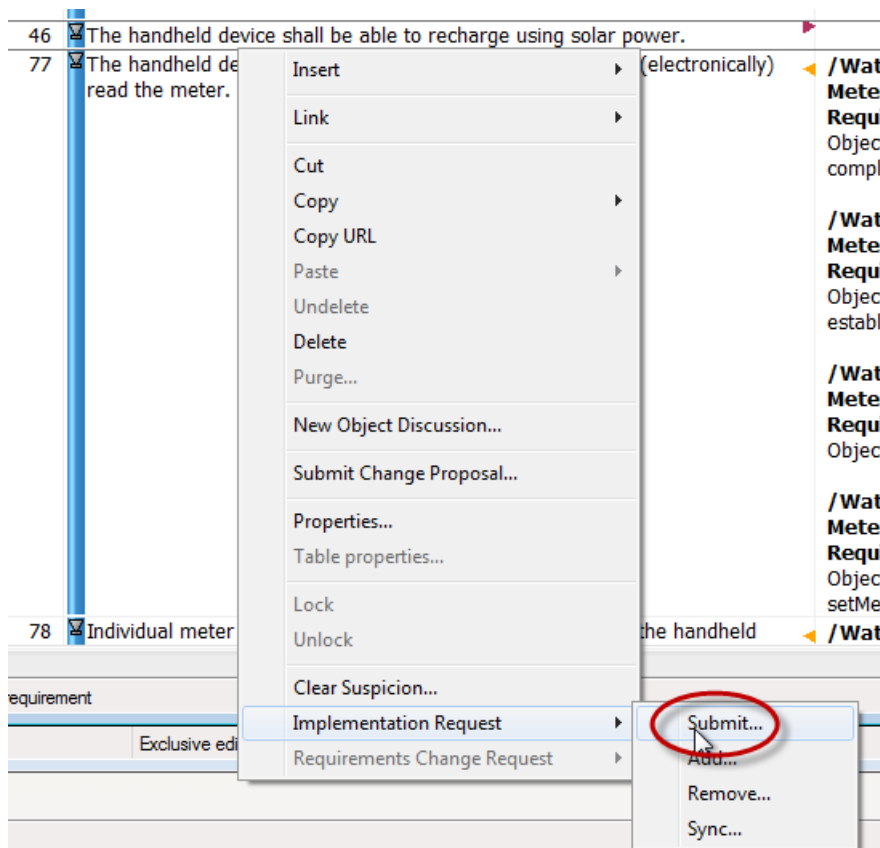
- \_\_11. In the next section we will take this requirement and perform an implementation request to show how requirement can be optionally linked to tasks in RTC.


## Lab 3 Creating an implementation request from DOORS

	<p><b>Lab Scenario</b></p> <p>In this lab we explore the use of the OSLC interface to create an implementation request (task) in RTC from DOORS.</p> <p>You will perform the following tasks:</p> <ul style="list-style-type: none"> <li>• Creating an implementation request for a DOORS requirement.</li> </ul> <p><b>Lab Setup</b></p> <p>This lab requires the completion of Labs 1-2.</p>
---	--

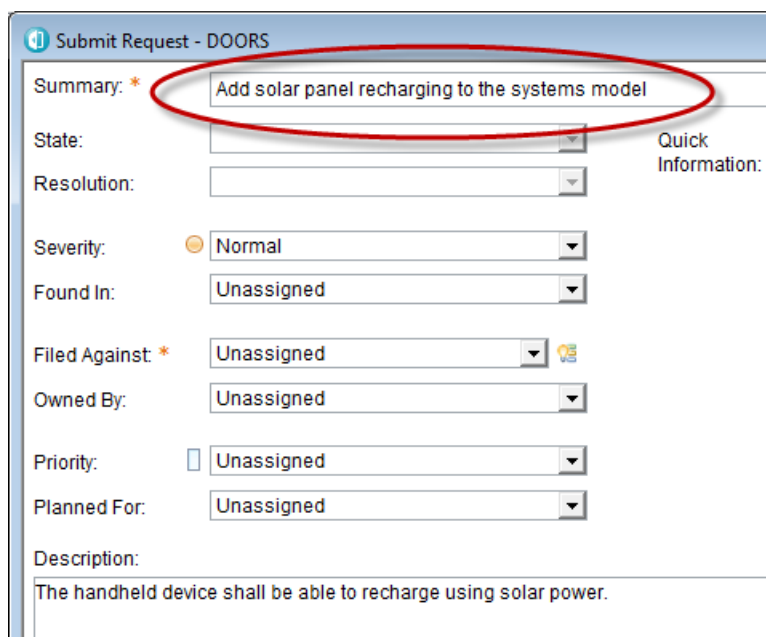
### 3.1 Create an implementation request

1. With requirement 46, highlighted "The handheld device shall be able to recharge using solar power", right-click and choose **Implementation Request > Submit...**



	<p><b>Benefit</b></p> <p>At this point we are using the RTC/DOORS integration. This integration enables the change management of DOORS data to be managed using Work Items in RTC. It also enables Tasks to be created to perform work associated with a particular requirement. This latter capability is known as an <b>Implementation Request (IR)</b>, whereas the former is known as a Requirement Change Request (RCR).</p>
---	---

- \_\_2. We are now going to create a task in RTC to perform some architectural design work on this requirement. At this point we are using the OSLC API with RTC.
- \_\_3. Give the task a name, e.g., Add solar panel recharging to the systems model.



Submit Request - DOORS

Summary: \* Add solar panel recharging to the systems model

State: [Dropdown] Quick Information:

Resolution: [Dropdown]

Severity: [Normal] [Dropdown]

Found In: [Unassigned] [Dropdown]

Filed Against: \* [Unassigned] [Dropdown]

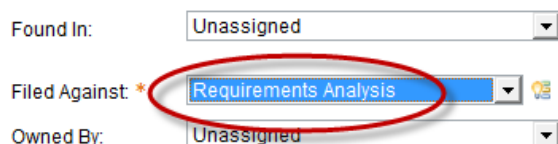
Owned By: [Unassigned] [Dropdown]

Priority: [Unassigned] [Dropdown]

Planned For: [Unassigned] [Dropdown]

Description:  
The handheld device shall be able to recharge using solar power.

- \_\_4. Set *Filed Against* to **Requirements Analysis**.



Found In: [Unassigned] [Dropdown]

Filed Against: \* [Requirements Analysis] [Dropdown]

Owned By: [Unassigned] [Dropdown]

Essentially this is a category in the project that aligns to a particular team. This pull-down can also be configured for a project to provide a list of architectural elements, e.g., GUI or code-generator etc.

- \_\_5. Set *Priority* to **High**.



Owned By: [Unassigned] [Dropdown]

Priority: [High] [Dropdown]

Planned For: [Unassigned] [Dropdown]

\_\_6. Set *Planned For* to the iteration **M2 - Requirements Analysis**.

Summary: \*

State:

Resolution:

Severity:

Found In:

Filed Against: \*

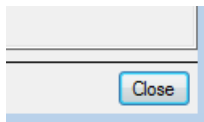
Owned By:

Priority:

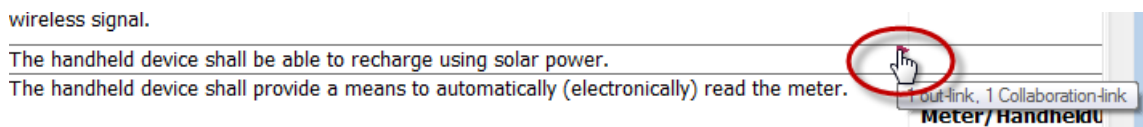
Planned For:

Description:

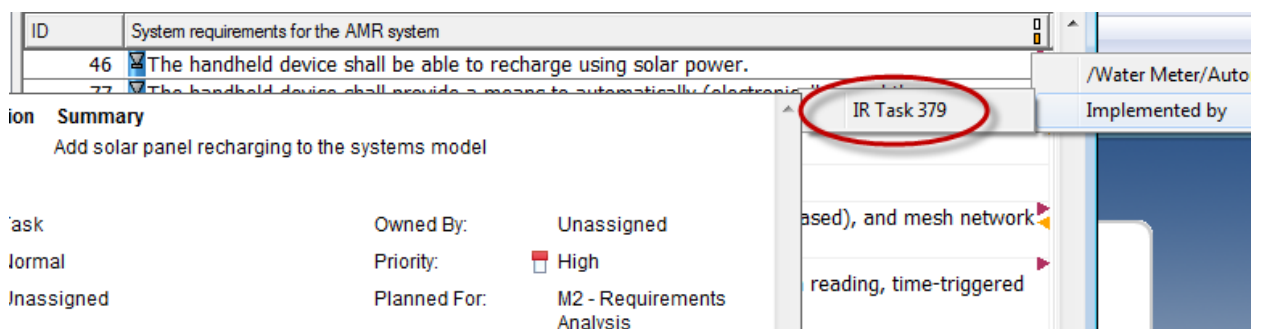
\_\_7. Click **OK** to create the Task. Note that DOORS launches a *Recording Report - DOORS* dialog summarizing the changes made. Click the **Close** button to clear this.



\_\_8. Right-click the link icon to the right of requirement 46. Note that an *Implemented by external link* has now been added. This links to the task in RTC that was created.




\_\_9. Note that if you hover over the IR Task 379 link then a dialog is shown giving more detail about the task in RTC without having to open it.



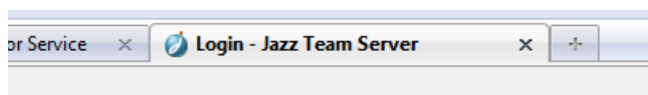
	<p><b>Benefit</b></p> <p>Rich hover over is a usability feature of the OSLC style integration and highlights how it is possible for one server to show data in another server by setting up user authentication between them.</p>
--	---

## Lab 4 Using RTC to plan and assign work

	<p><b>Lab Scenario</b></p> <p>In this lab we explore the key aspects of Rational Team Concert in the context of directing the work on complex systems and software projects. Included in the use of a dashboard to gain visibility of live data associated with the project.</p> <p>You will perform the following tasks:</p> <ul style="list-style-type: none"> <li>• Exploring the project dashboard and customizing it by adding a widget.</li> <li>• Allocating unassigned work.</li> </ul> <p><b>Lab Setup</b></p> <p>This lab requires the completion of Labs 1-3.</p>
---	--

### 4.1 Allocating the unassigned work

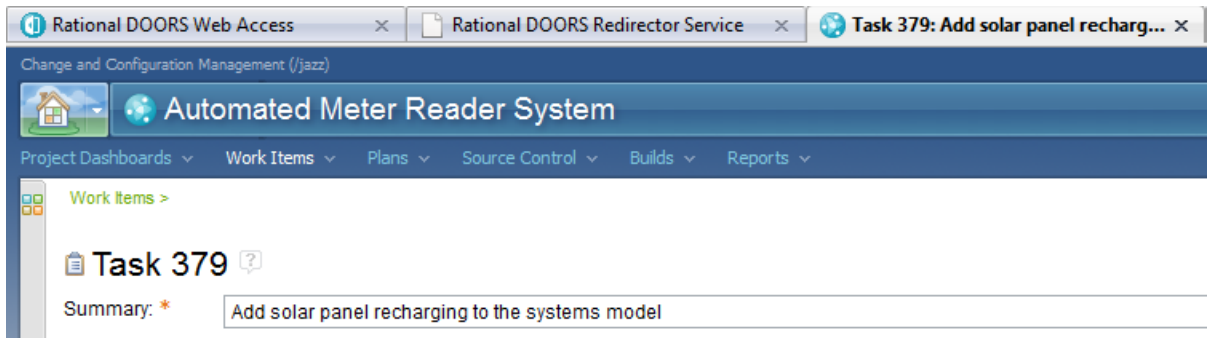
- \_\_\_1. Left click the **Implemented by > IR Task 379** external link open it up in a web browser. DOORS should open up a *Login - Jazz Team Server* window in Firefox.



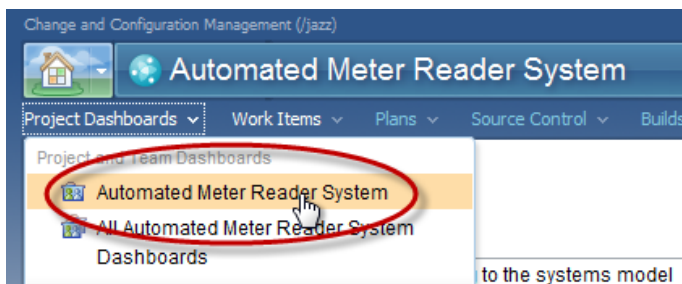
- \_\_\_2. Let's login as the project manager to see his view of the project. Login as **pete/pete** (both lowercase).




\_\_3. We're now viewing the task using the RTC web client.



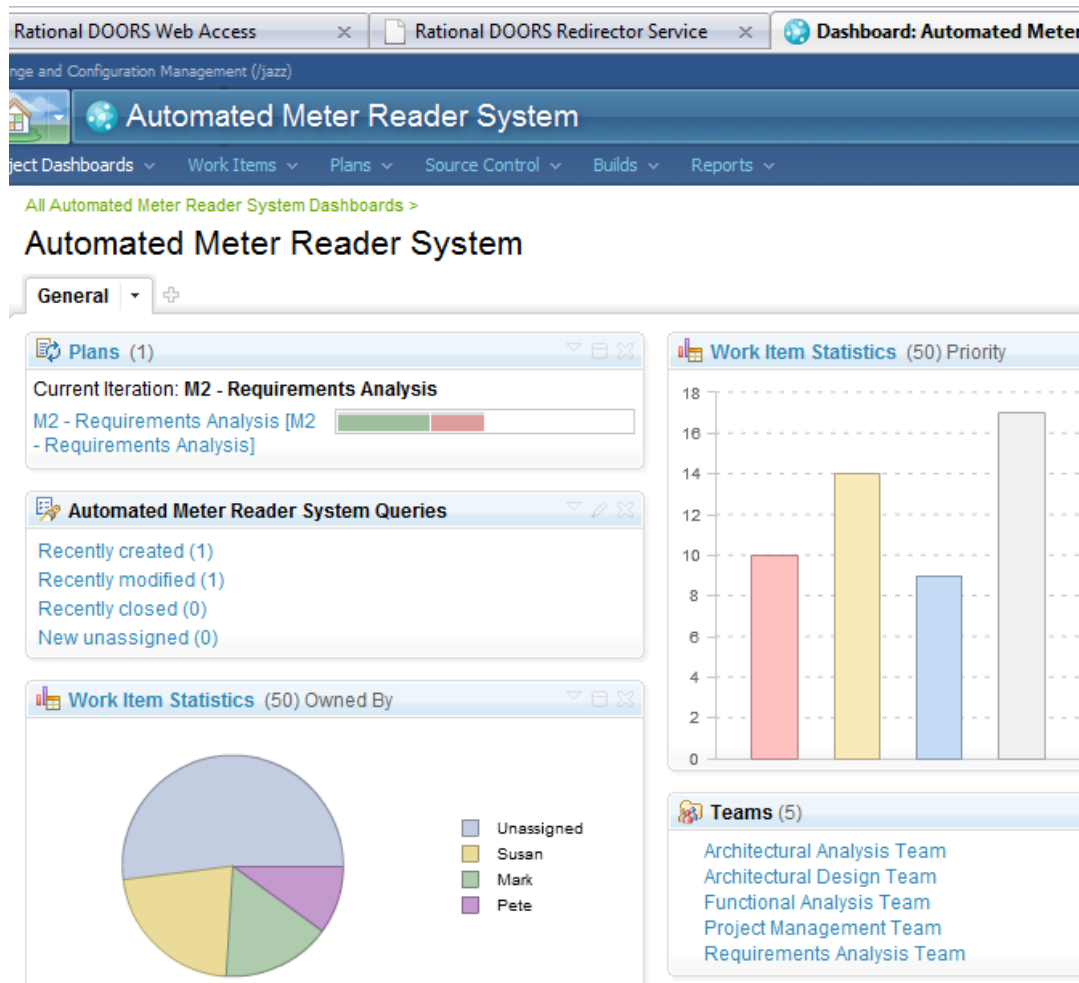
\_\_4. Click **Project Dashboards > Automated Meter Reader System**.




	<p><b>Benefit</b></p> <p>Typically project managers will use the thin client to view data in RTC. One of the most useful views is something called a dashboard. A dashboard is a customizable page that displays information about the project. There is a default dashboard for each project.</p>
---	--



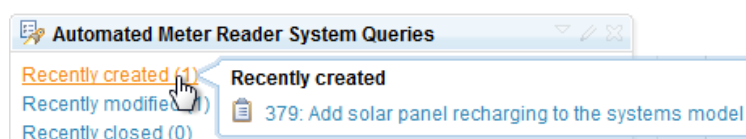
- \_\_\_5. Have a look at the default dashboard for the *Automated Meter Reader System*. Note that this consists of a number of widgets that either act as feeds for events or display the result of queries on the database.




- \_\_\_6. This is the first view that a Project Manager might get when they login. Everything he/she needs on a single page and no need to install the tool.

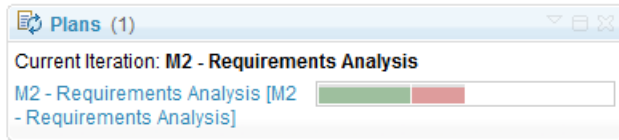
	<p><b>Benefit</b></p> <p>This is what we mean by project transparency. RTC gives project managers data at their fingertips and can automate the presentation of that data into a consumable and readily understandable format. Managers can spend their time managing, rather than compiling spreadsheets manually.</p>
---	---

- \_\_\_7. The project dashboard is also based on real live data.
- \_\_\_8. We also have some useful feeds here. Hover over the "Recently created" list and we can see the task that was just added.

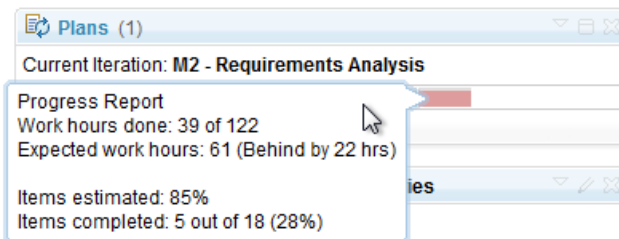


	<p><b>Benefit</b></p> <p>The fact that the data is live removes the time needed to assemble the data, and means that it is always up to date.</p>
---	---

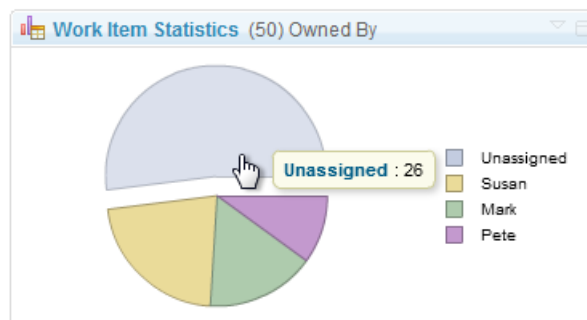
\_\_9. One of the most useful summaries is the progress bar for the Plan. This view gives an indication of the progress being made on the current iteration. The green indicates that some progress has been made but the red indicates that some work is taking longer than estimated and hence the team is falling behind. Fortunately, there is time to catch up, we can tell this from the white space.



\_\_10. Notice that hovering over the graphical icons provides you with a more detailed view without having to navigate.



\_\_11. Click on the **Unassigned** wedge in the Work Item Statistics.



\_\_12. Notice that this gives you a list of Work Items that match the query.





[Work Items > Queries >](#)

### Requirements Analysis ?

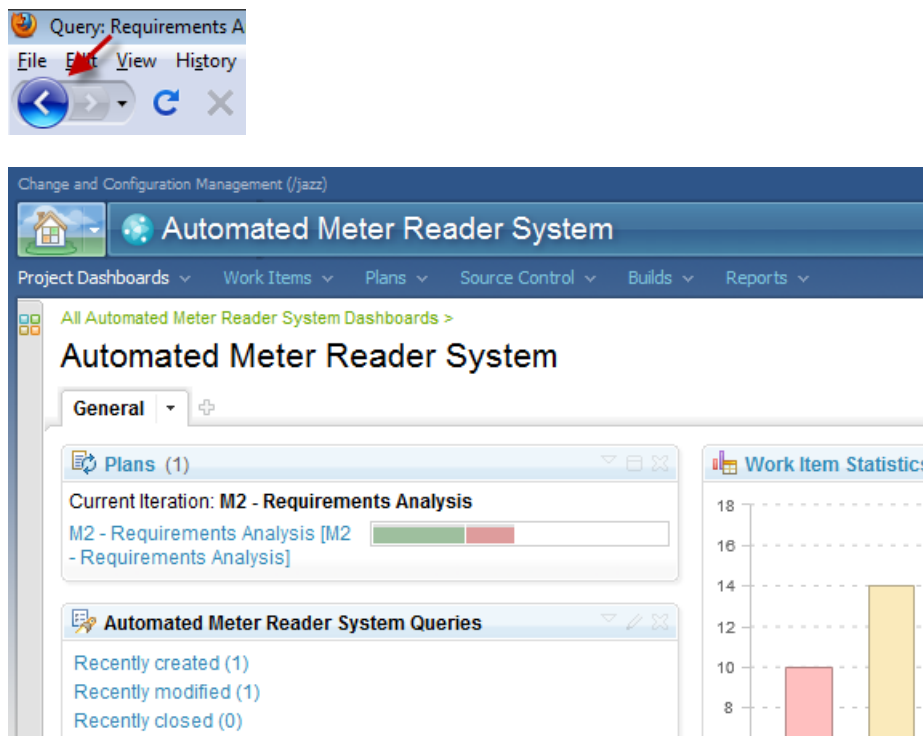
Show All ▼ Items Per Page

Previous | 1 - 26 of 26 | Next



Type	Id	Summary	Owned By
	379	Add solar panel recharging to the systems model	Unassigned
	377	RCR: Update system requirements to include meter leak data	Unassigned
	315	Identify System Use Cases	Unassigned
	242	RCR: Add leak indicator to Handheld	Unassigned

\_\_13. Browse back to the main dashboard view.



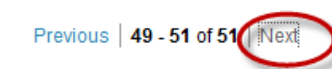
## 4.2 Customizing the dashboard by adding a widget

- \_\_1. Dashboards are fully customizable. In our project we have already placed some governance controls to improve quality. In particular, we've decided that all Work Items need to be reviewed before they can be delivered.
- \_\_2. Let's add a widget to the dashboard to view a list of Work Items currently pending a review by the user by the user who is logged in. Select the arrow icon to the right of the *General* tab, left-click and choose **Add Widget** from the menu.

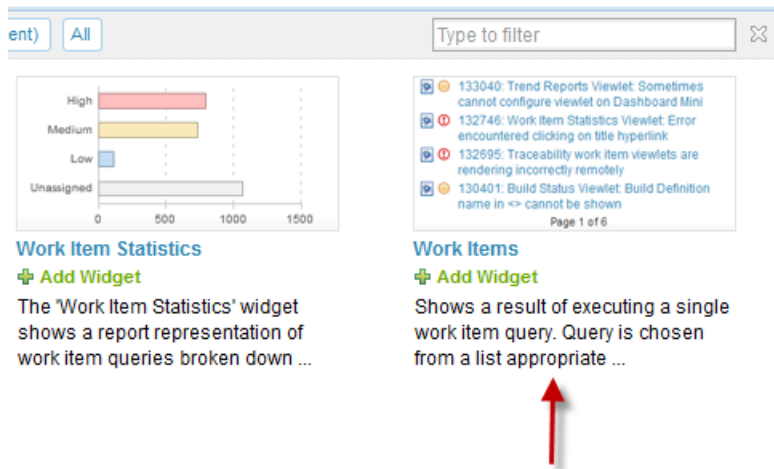
### Automated Meter Reader



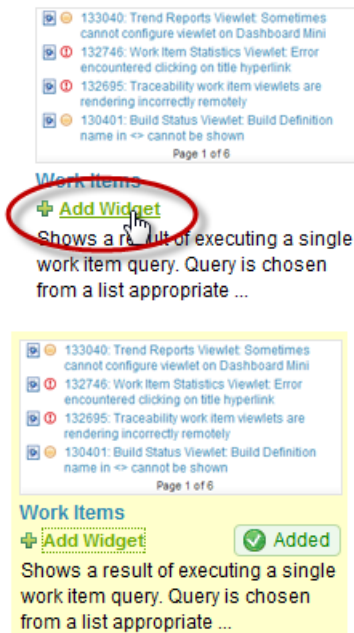
- \_\_3. A panel showing Widgets that can be selected is presented. The one we want is actually the last one on the last page. Click **Next** until you get to the last page.



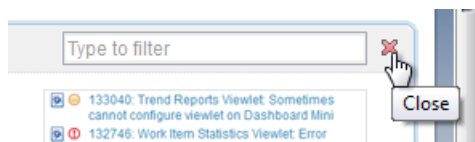
4. Find the last widget called **Work Items**. This shows a list of Work Items that result from executing a single query. This shows a result of executing a single work item query.



5. Click **Add Widget** to add the *Work Item* widget.

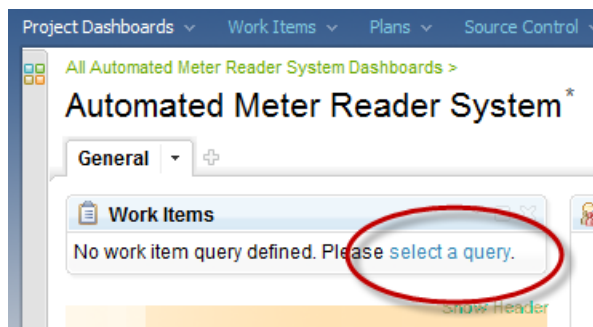


6. Once you've done this click **Close** icon to close the *Add Widget* panel.

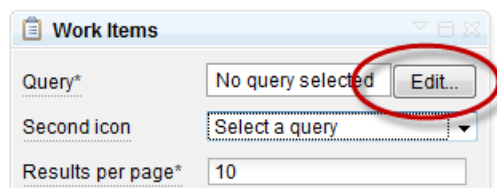


7. Note that the project dashboard now has a new *Work Items* widget. However, the widget says that "No work item query was defined."

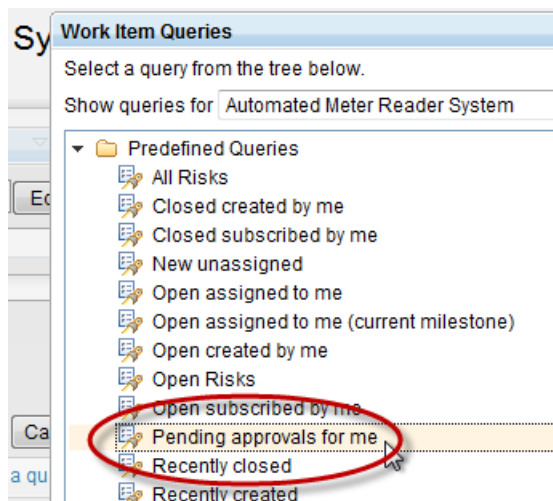
\_\_8. Let's add a query. Click **select a query**.



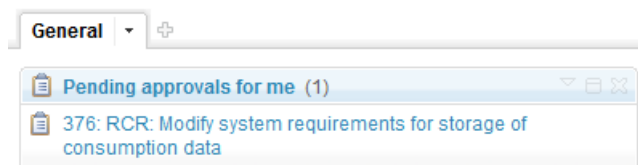
\_\_9. Click **Edit...** to select a query.



\_\_10. Choose **Pending approvals for me** from the list of *Predefined Queries*.

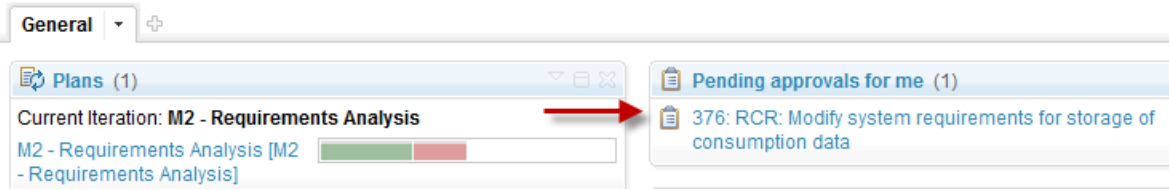


\_\_11. Hit **OK** and **Save**. You should now have a widget on the dashboard that shows the "Pending approvals for me" with an existing RCR in it.

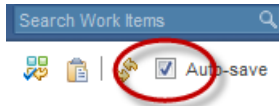


\_\_12. Drag it to the right of the plan widget.

### Automated Meter Reader System

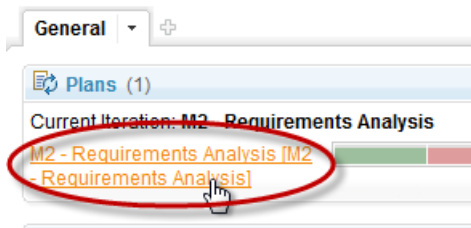


\_\_13. Check the **Auto-save** box in the top-right corner to save the change before proceeding.

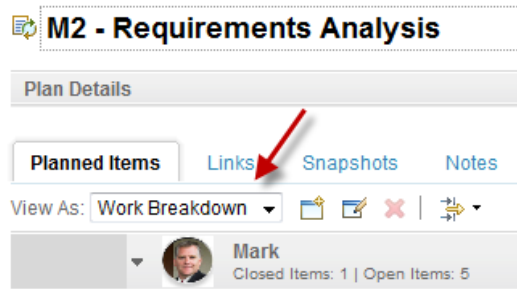


## 4.3 Exploring the plan views

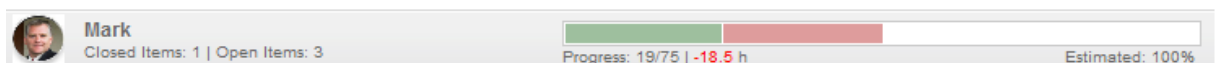
\_\_1. Let's explore the plan a little further. Navigate to the plan for the current iteration by clicking on the link to the left of the progress bar.



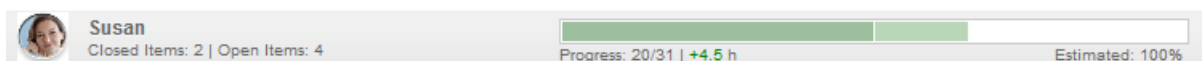
\_\_2. A plan is really a filtered view of the database that just shows the work items that are applicable to part of the project. The plan can be filtered with a number of views. Note that initially we are looking at the *Work Breakdown* view.



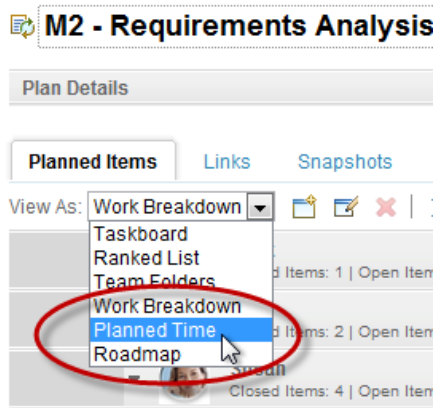
\_\_3. In this view we can see the members of the team and what *Work Items* they are currently tasked with. Again we can gain a lot of information visually. For example, we can see that in terms of progress *Mark* is currently behind on his tasks (from the red)...



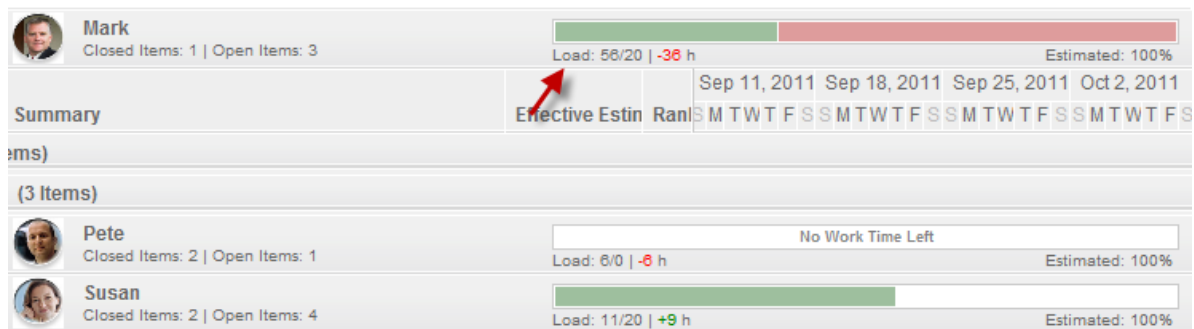
\_\_4. ... However, *Susan* is ahead in her tasks (from the light green).



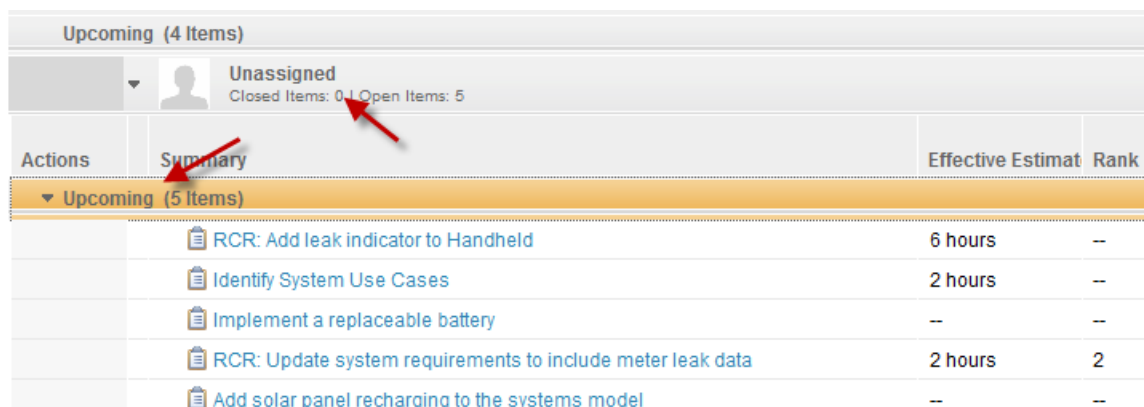
- \_\_\_5. The Progress bar is showing us what progress they've made on the tasks allocated to them. Green indicates the work they've done, red indicates that work is taking longer than estimated, but white indicates that they have time to catch up.
- \_\_\_6. Note that we has a number of views available in the *View As* pull-down menu. Switch to the **Planned Time** view.



- \_\_\_7. Here the bars are set to show *Load* by default rather than *Progress* (we can toggle the bar to show progress if we like by left-clicking on it).

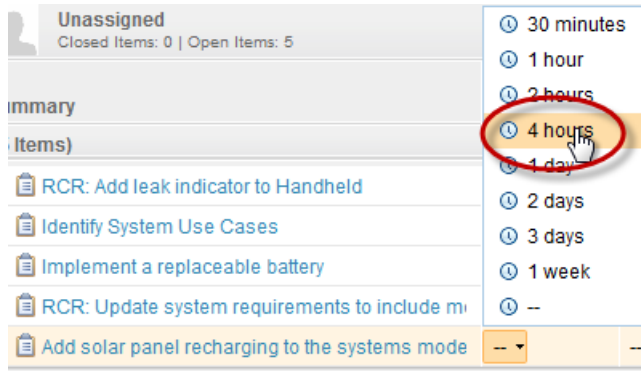


- \_\_\_8. From the red we can see that *Mark* is overloaded compared with *Susan*. Note that *Pete* has no time allocated to the project.
- \_\_\_9. Click on **Unassigned** bar and then upcoming to find the "solar panel" task we added.

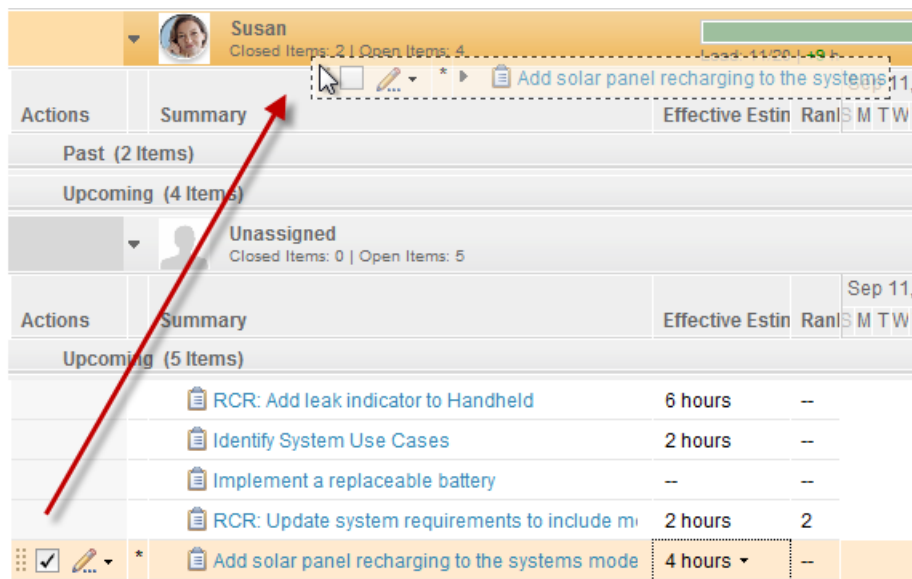


Actions	Summary	Effective Estimat	Rank
▼ Upcoming (5 Items)			
	RCR: Add leak indicator to Handheld	6 hours	--
	Identify System Use Cases	2 hours	--
	Implement a replaceable battery	--	--
	RCR: Update system requirements to include meter leak data	2 hours	2
	Add solar panel recharging to the systems model	--	--

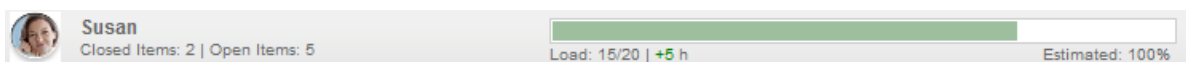
\_\_10. Change the estimate to **4 hours**.




\_\_11. Left-click on the dot symbol to the left and drag the task on to *Susan*.



\_\_12. Verify that Susan's load went up when you did this.



	<p><b>Benefit</b></p> <p>Project transparency helps keep projects healthy. In theory an overloaded situation, like that shown with Mark, should not happen or, at the very least, it should not persist for long. This is because live planning means that resourcing issues are visible not hidden in a spreadsheet.</p>
---	---


\_\_13. Check the **Auto-Save** box in top right before proceeding.



\_\_14. Keep the web client open for pete. We will come back to this. In the next section we're going to switch to Susan's view of the project in the Eclipse client.

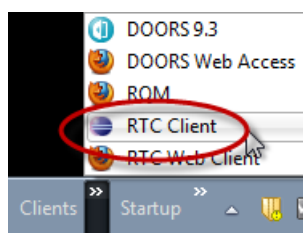


## Lab 5 Using Rational Rhapsody to perform the work

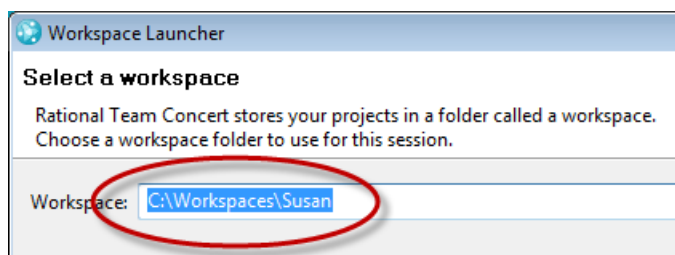
	<p><b>Lab Scenario</b></p> <p>In this lab we explore the key aspects of Rational Team Concert in the context of performing systems engineering work using Rational Rhapsody in Eclipse.</p> <p>You will perform the following tasks:</p> <ul style="list-style-type: none"> <li>• Performing work in Rhapsody Eclipse client for an assigned task.</li> <li>• Submitting pending changes for review.</li> </ul> <p><b>Lab Setup</b></p> <p>This lab requires the completion of Labs 1-4.</p>
---	--


### 5.1 Susan performs some work against the task

- \_\_1. Note that RTC has both an Eclipse client and a web client. Let's log in to the Eclipse client now as *Susan* now in order to do the work.
- \_\_2. Open the Eclipse client using the **Clients > RTC Client** shortcut in the taskbar.

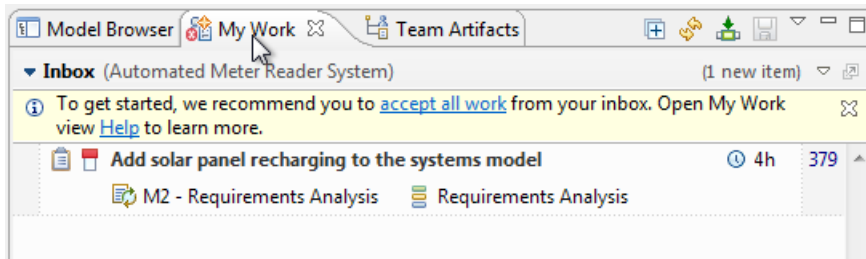


- \_\_3. Select Susan's workspace from the pull-down list.

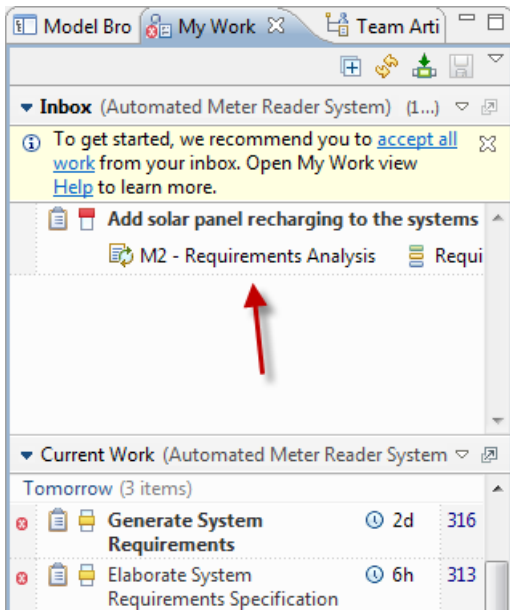


	<p><b>Benefit</b></p> <p>The web client is useful for high-level management, who just want to view dashboards or for users just doing document management. The Eclipse client has integrations into the wider tool portfolio and hence is usually the tool needed by developers. One further benefit of the thick client is the ability to work off-line, if needed. For example, if you want to work remotely.</p>
---	---

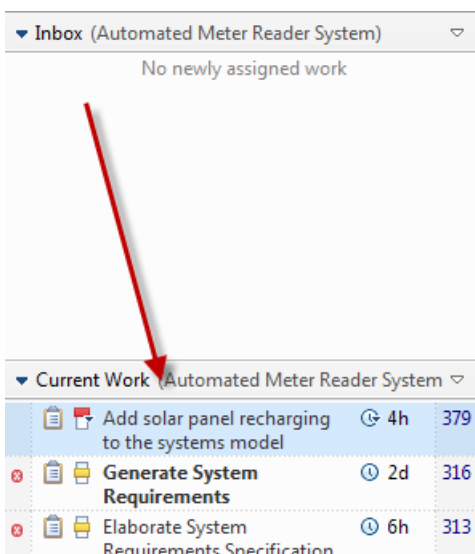
\_\_4. Ensure that the *My Work* view is shown.



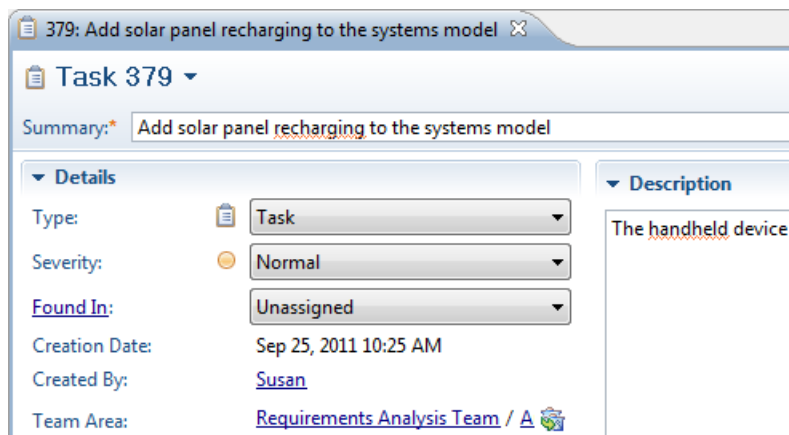
\_\_5. Note that the new task is currently in Susan's inbox.



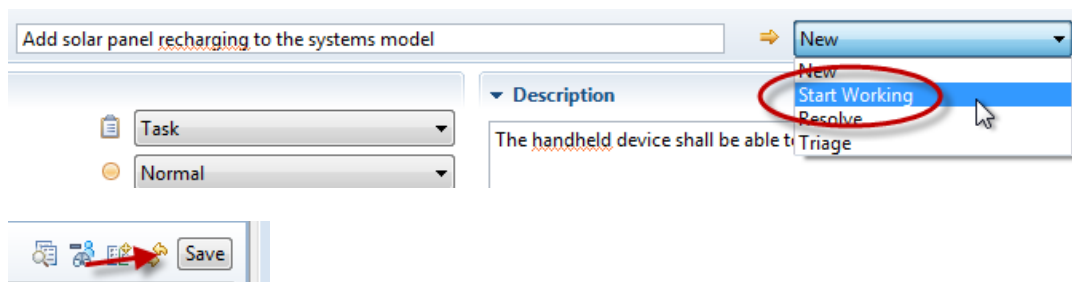
\_\_6. Since Susan can see that the task is high priority, she's going to accept it. Drag the *Add solar panel...* task to the top of the *Current Work* list for today.



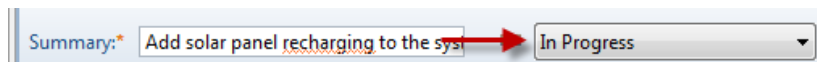
\_\_7. Double-click the *Add solar panel...* task to open it.



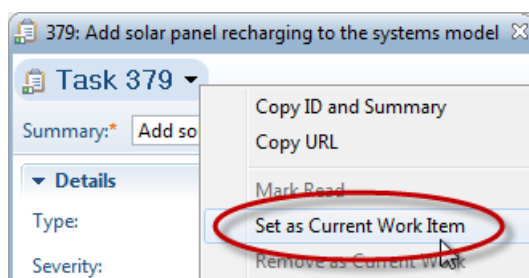
\_\_8. Let's do some work for this task. Change the task from New to **Start Working** and press **Save**.



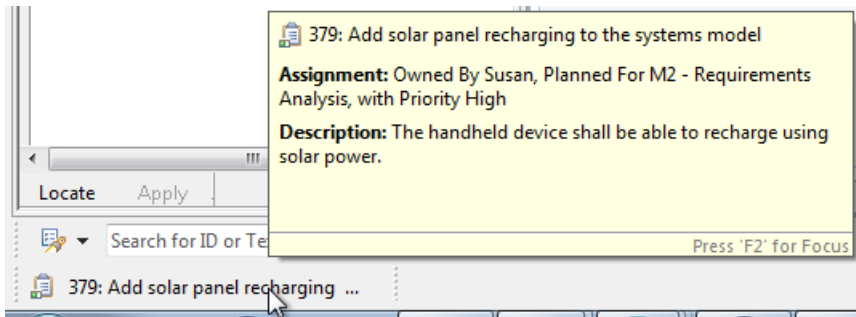
\_\_9. Note that the task is now marked as *In Progress*.



\_\_10. Let's also set the task as the current work item. Click on the down arrow next to the task number and choose **Set as Current Work Item**.



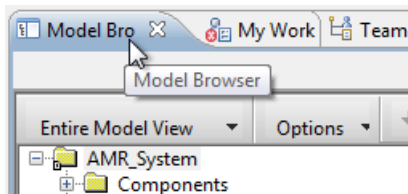
\_\_11. Note that the task is now shown at the bottom left of the Eclipse window.



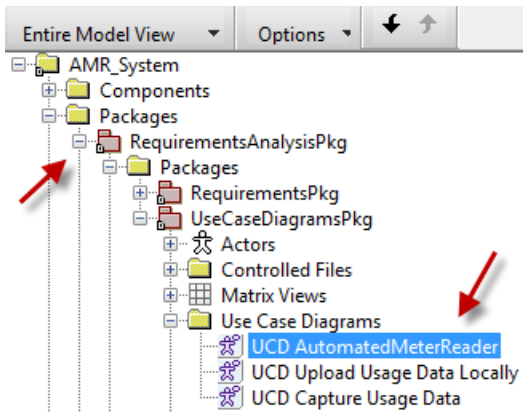
## 5.2 Make a change to the Rhapsody model

\_\_1. Let's make a change to the Rhapsody model and see how the change is tracked by Rational Team Concert.

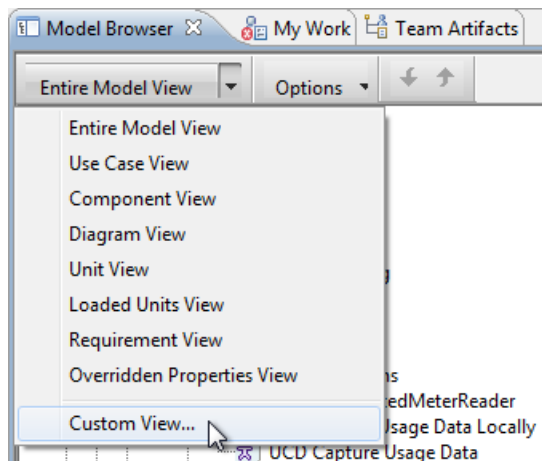
\_\_2. Click from the *My Work* view to the *Model Browser* view.



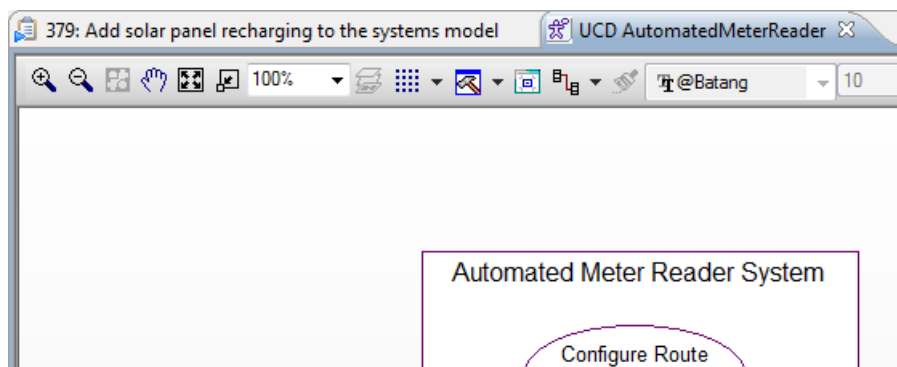
\_\_3. Click on the [+] icons to expand the browser until you find the *UCD AutomatedMeterReader* diagram.



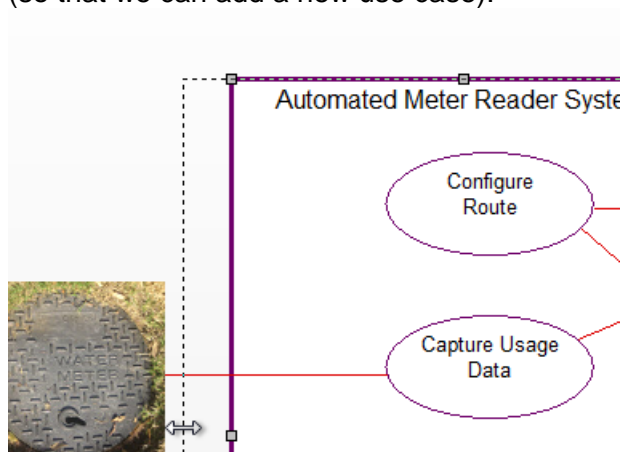
Note: An alternative way to find the same diagram is to use the Custom View filter to simplify the browser to just show UseCaseDiagrams.



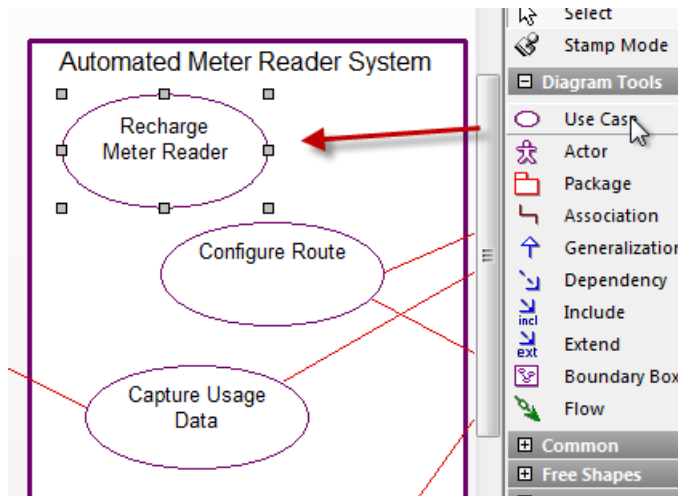
- \_\_4. Double click the *UCD AutomatedMeterReader* use case diagram to open it up.



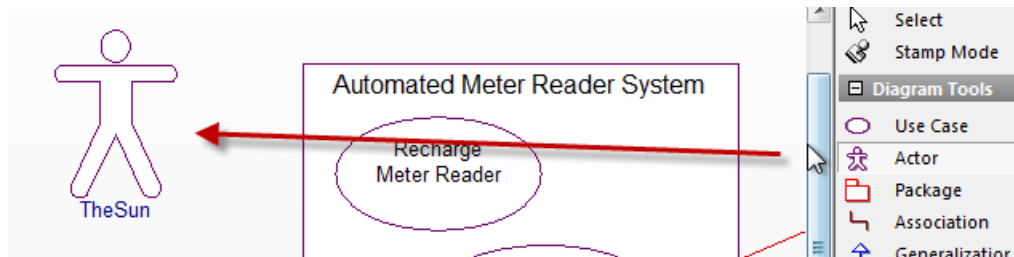
- \_\_5. Hold down the alt key and drag to make the boundary box wider without re-sizing its inners (so that we can add a new use case).



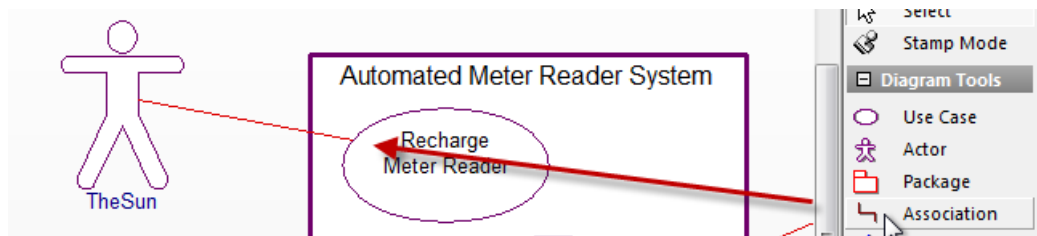
- \_\_6. Make some space and use the toolbar to add a new **use case** to the diagram with the name *Recharge Meter Reader*.



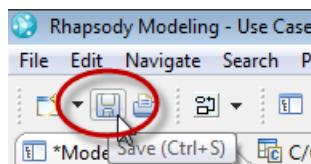
- \_\_7. Add a new **actor** called "TheSun".



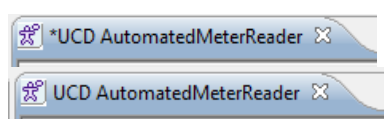
- \_\_8. Draw an **association** between the *TheSun* actor to the *Recharge Meter Reader* use case.



- \_\_9. Click on the **Save** icon in the main toolbar to save the changes to the diagram.

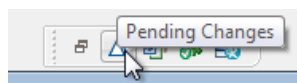


- \_\_10. Verify that the diagram is now saved, i.e. the asterisk (\*) went away when you saved.

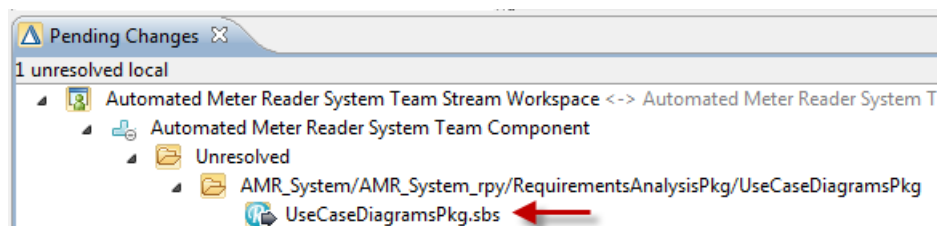


### 5.3 Attempt to deliver a change set

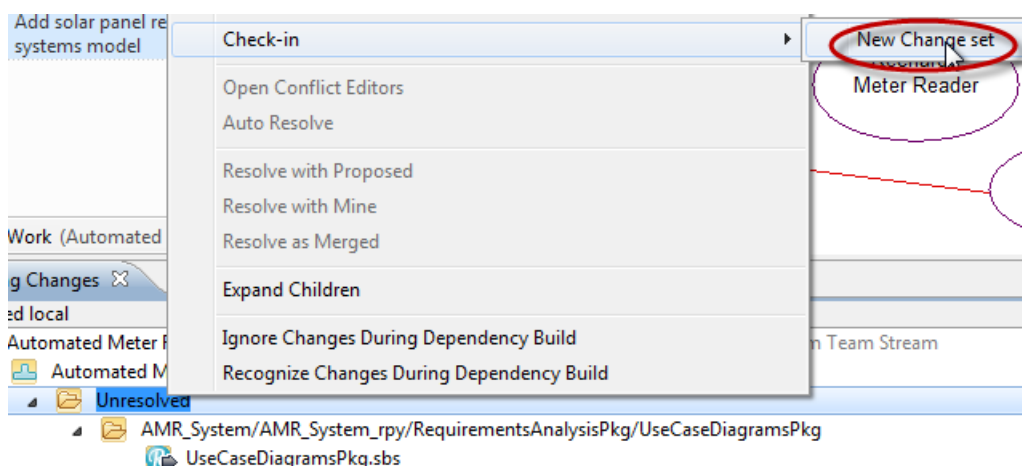
- \_\_\_1. Let's now see how the change was tracked by Rational Team Concert. In the panel at the bottom choose **Pending Changes** to show the *Pending Changes* window.



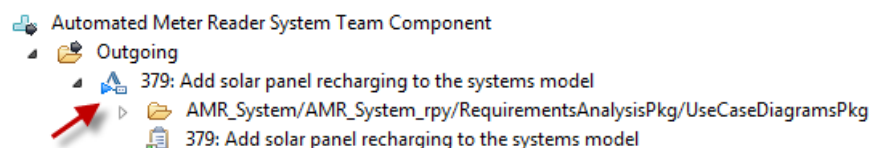
- \_\_\_2. Expand the **Unresolved** folder to see the uncommitted changes. Notice that the *UseCaseDiagramsPkg.sbs* file has been changed.



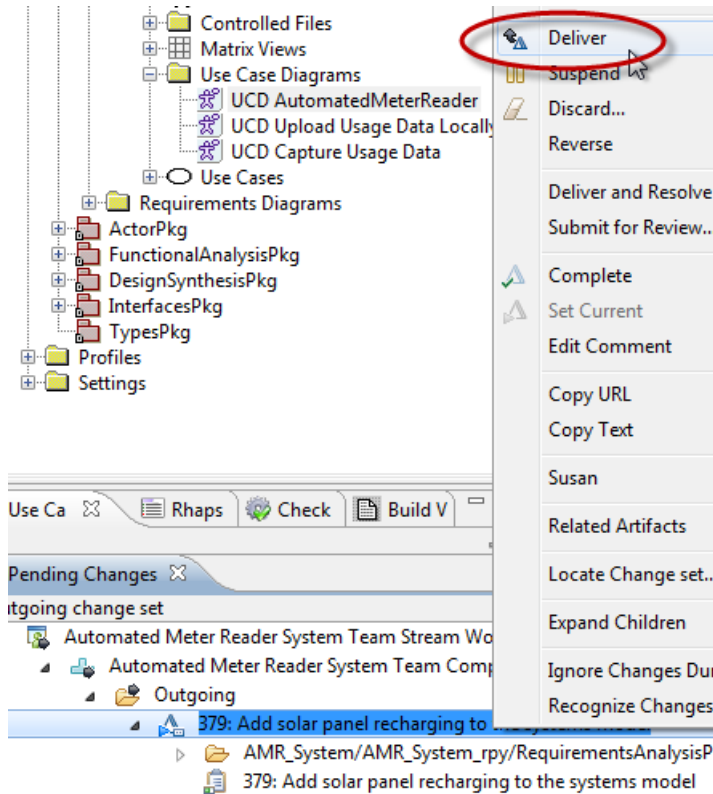
- \_\_\_3. Let's try and deliver the changes to the stream. Right-click the *Unresolved* folder and choose **Check-in > New Change set** from the context menu.



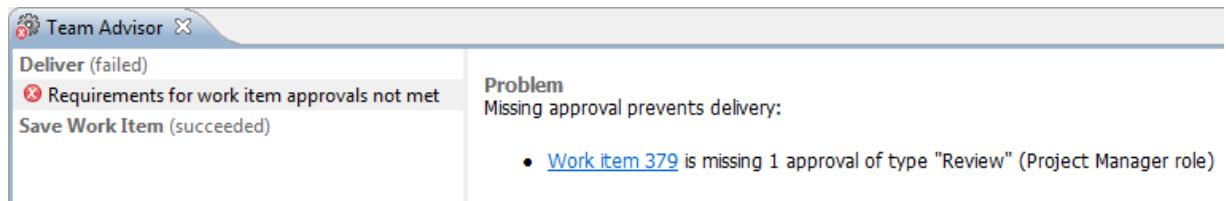
- \_\_\_4. Note that a new "change set" was created. The change set ensures that associated changes are grouped together. Because we have a current work item, RTC has also associated this change set with it.




5. The changes are now persisted in your repository workspace, a private area on the server. However, they have not yet been delivered to the stream. Let's attempt to do this now. Right-click the change set and choose **Deliver** from the context menu.



6. Note that the *Team Advisor* view tells Susan that she cannot deliver her work because the "Requirements for work item approvals have not been met".



7. Essentially we have configured our process to enforce that pre-condition of the deliver operation is that the Change Set must be reviewed by a Project Manager.

	<p><b>Benefit</b></p> <p>The process in Team Concert is very configurable and can be shared using a process template. This enables us to automate aspects of the process, and enforce governance or quality control gates, if needed. This can save lots of time and effort trying to enforce a process manually.</p>
---	---

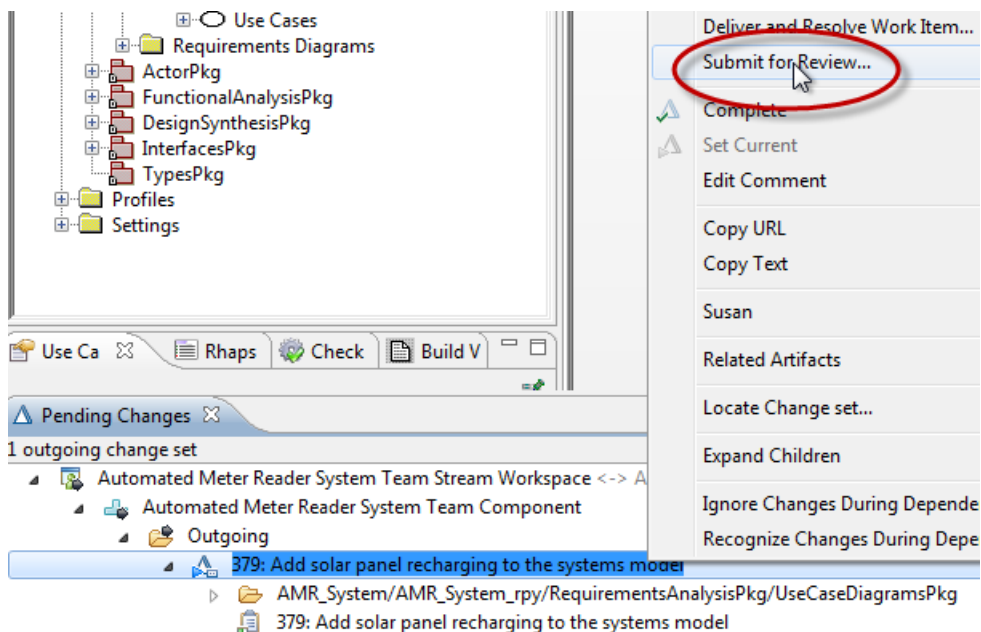


## 5.4 Using the Submit for Review feature

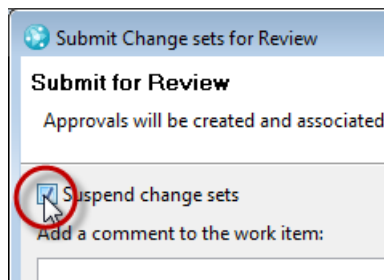
- \_\_1. Click on the icon to switch back to the *Pending Changes* view.




- \_\_2. This time choose **Submit for Review...**

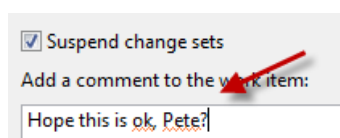


- \_\_3. Check the box to **Suspend change sets**.

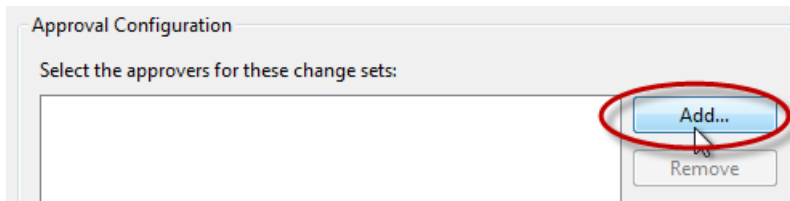


	<p><b>Benefit</b></p> <p>The suspend feature in RTC means that users can continue with other work in their "Repo workspace" while the review is pending.</p>
---	--

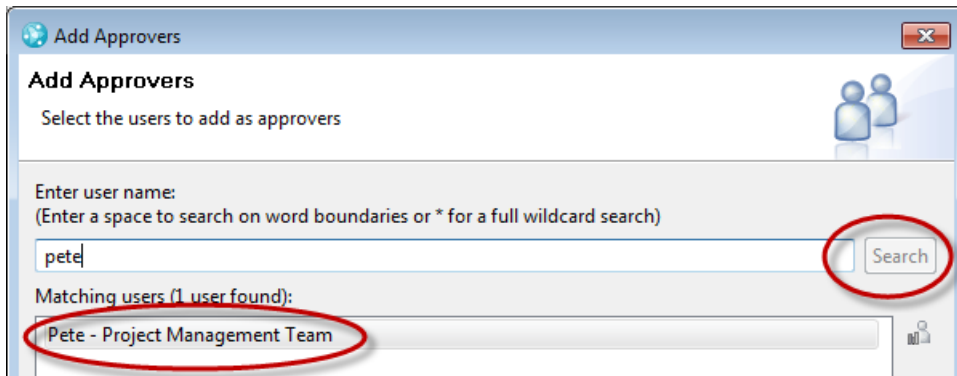
- \_\_4. Add a comment, e.g., "Hope this is ok, Pete?".



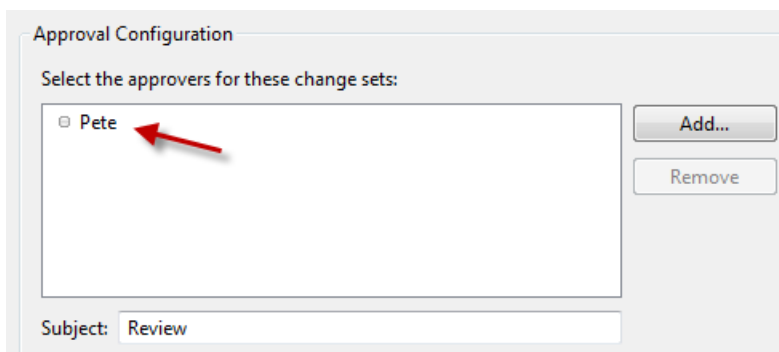
\_\_5. Click the **Add...** button to add an Approver.



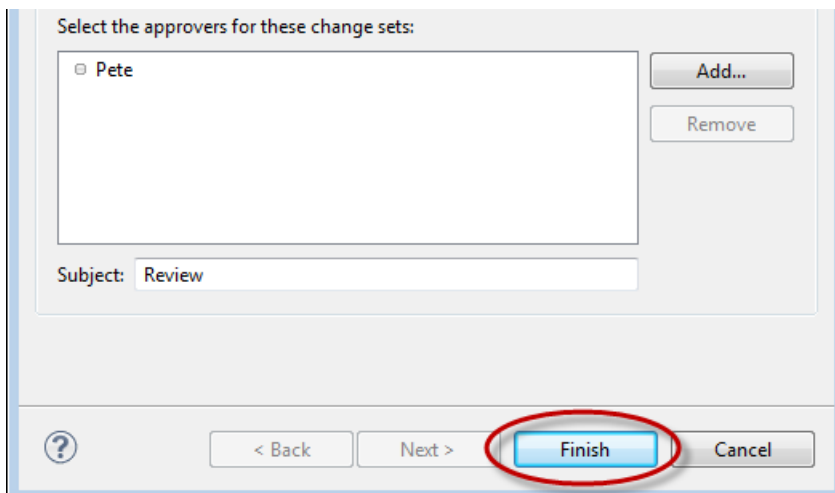
\_\_6. Perform a search for "pete". Click **Pete - Project Management Team** to add him as an approver for the review.



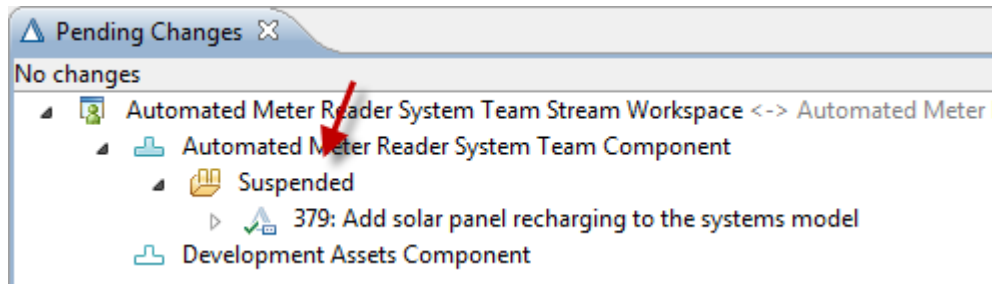
\_\_7. Verify that *Pete* was added as a review approver.




\_\_8. Click **Finish**.



\_\_9. Note that change set is now shown as suspended.

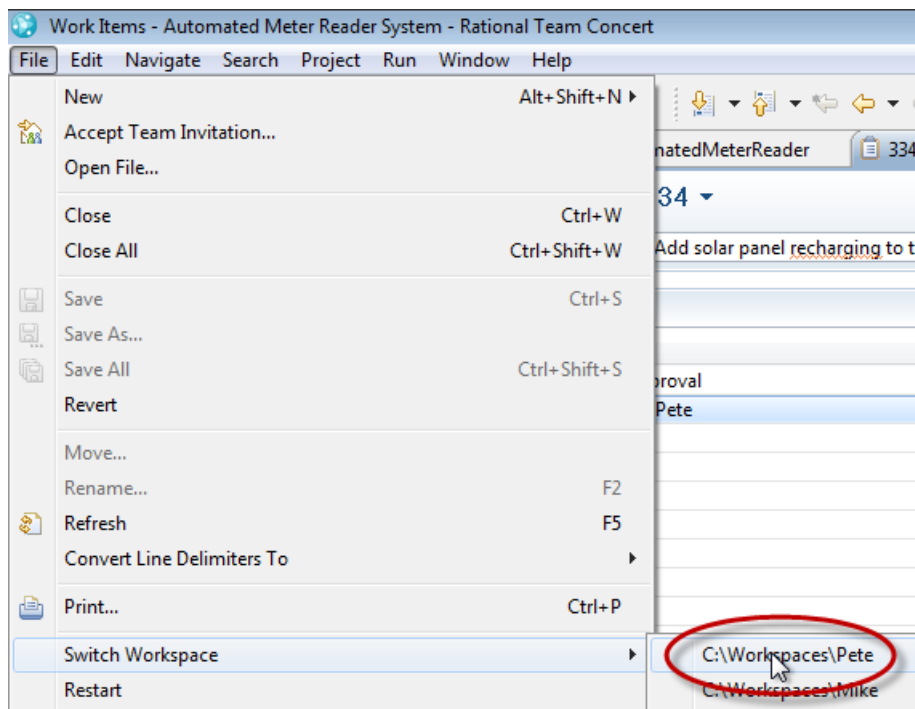


## Lab 6 Using RTC to review and approve work

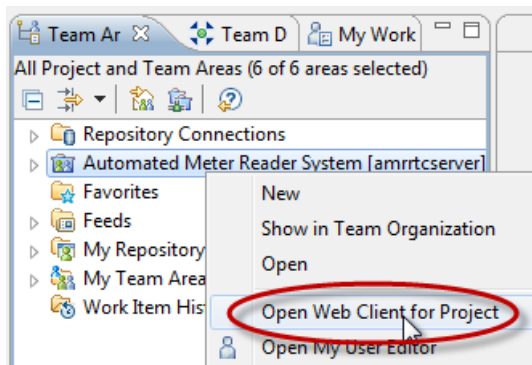
	<p><b>Lab Scenario</b></p> <p>In this lab we explore the key aspects of Rational Team Concert in the context of reviewing submitted by other users.</p> <p>You will perform the following tasks:</p> <ul style="list-style-type: none"> <li>• Reviewing work submitted by another user.</li> <li>• Use of Rhapsody diff/merge to see graphical differences.</li> </ul> <p><b>Lab Setup</b></p> <p>This lab requires the completion of Labs 1-5.</p>
---	---

### 6.1 Viewing a review request

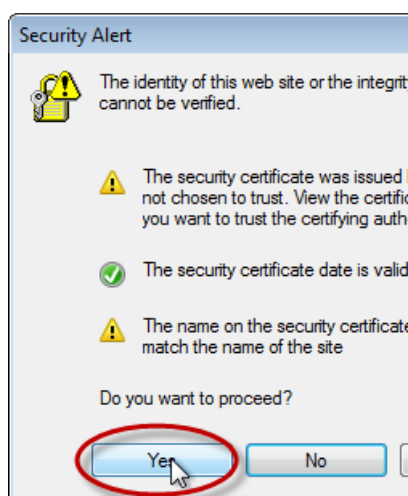
- \_\_1. To do the approval Pete is going to use the Eclipse client. This is because he wants to view some of the changes using Rhapsody. Choose **Switch Workspace > C:\Workspaces\Pete** to switch workspaces.



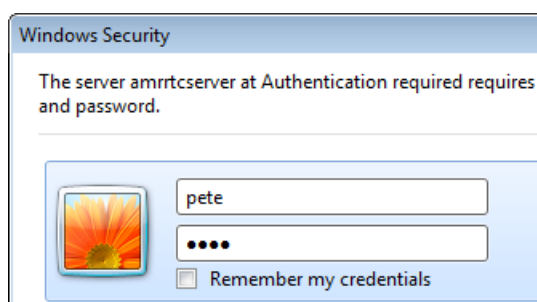
\_\_2. Right-click the *Automated Meter Reader System* and choose **Open Web Client for Project**.



\_\_3. Click **Yes** to proceed.

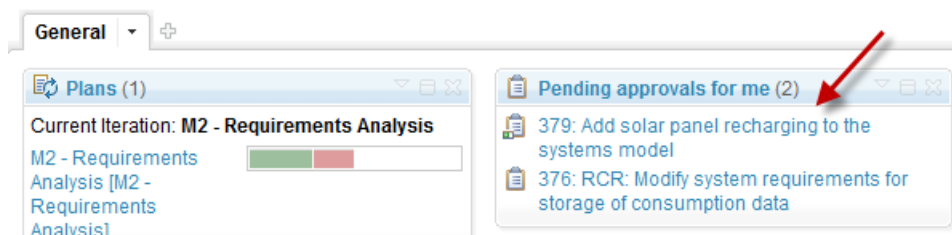


\_\_4. Login as **pete/pete**.

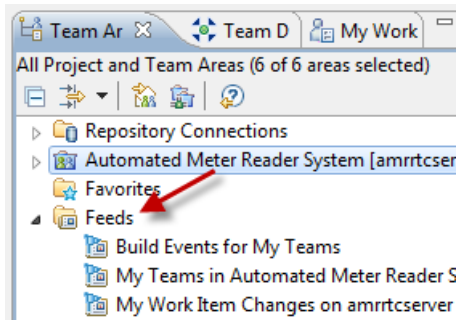


\_\_5. Note that Pete now has a work item pending approval.

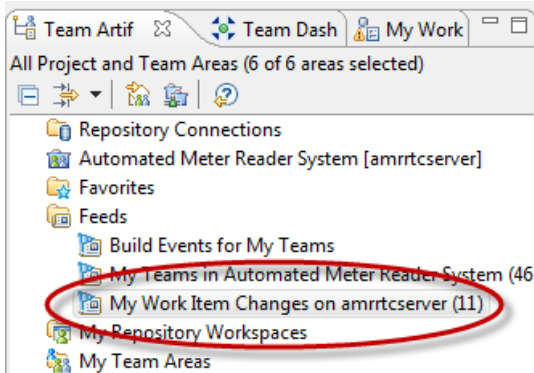
### Automated Meter Reader System



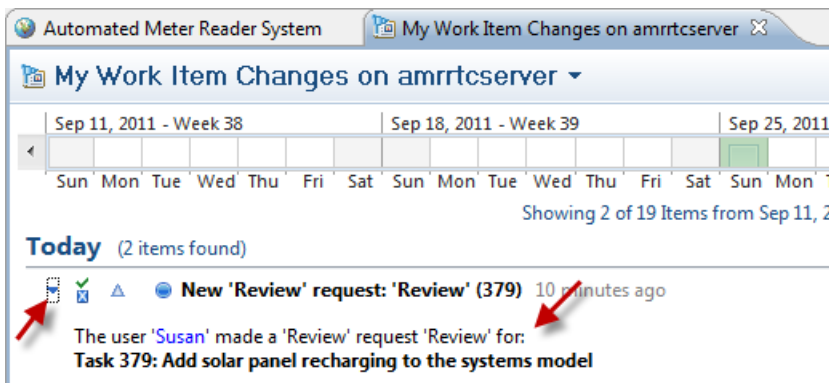
\_\_6. In the Eclipse browser, expand the *Feeds* folder in the *Team Artifacts* view.



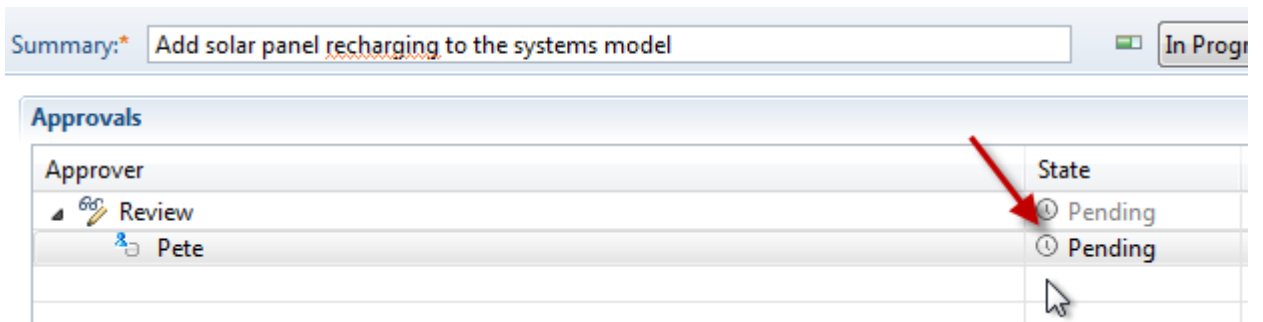
\_\_7. Double click to open the **My Work Item Changes** feed.



\_\_8. Expand the change. Note that *Susan* made a 'Review' request for approval for the task.

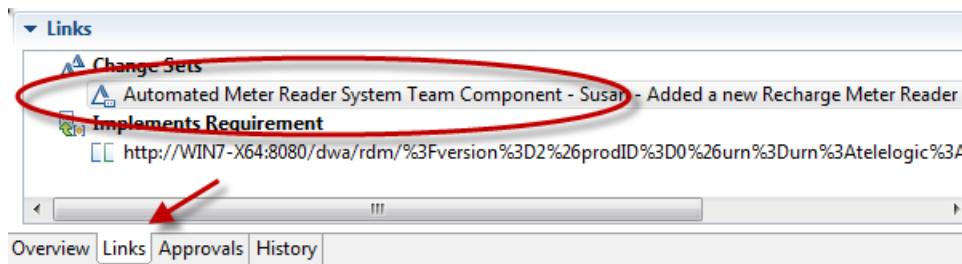


\_\_9. Click the request to open the task. Note that approval is "Pending".

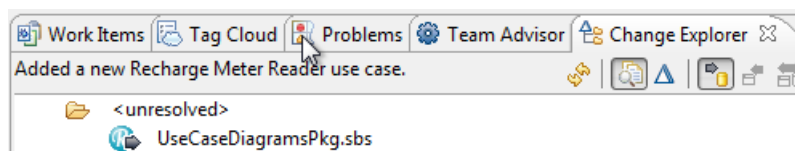


## 6.2 Reviewing changes in the change explorer

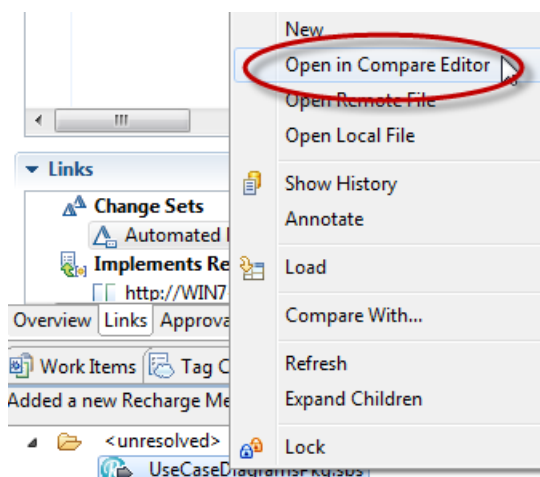
- \_\_\_1. Before we approve let's have a quick look at the *Change Set*. Switch to the **Links** tab and double-click on the *Change Set* to open it up in the *Change Explorer*.



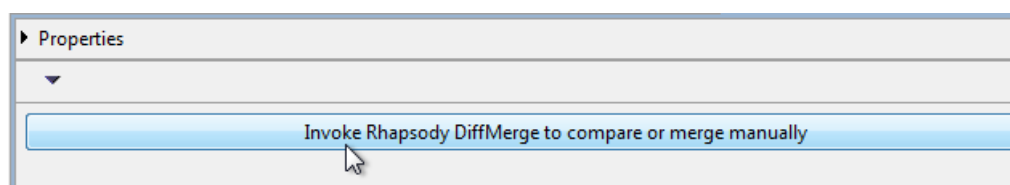
- \_\_\_2. Verify that a *Change Explorer* view opens.



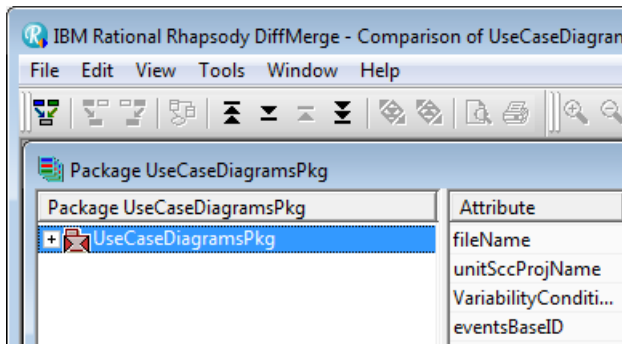
- \_\_\_3. Right-click *UseCaseDiagramPkg.sbs* file and choose **Open in Compare Editor**.



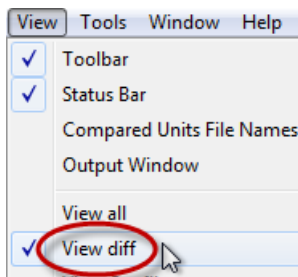
- \_\_\_4. Since this is a Rhapsody file, RTC will prompt to open using the Rhapsody Diff/Merge. Left click **Invoke Rhapsody DiffMerge to compare or merge manually**.



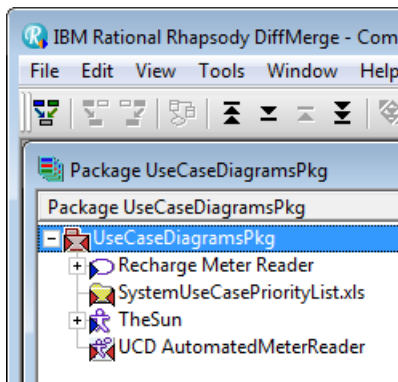
\_\_5. The *Rhapsody DiffMerge* should launch.



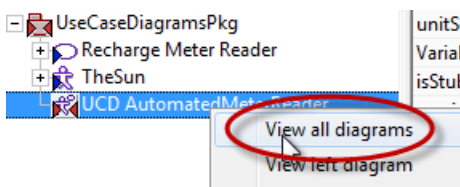
\_\_6. Select **View > View diff** from the main menu in the Rhapsody Diff/Merge.



\_\_7. The browser is now filtered to just show differences.

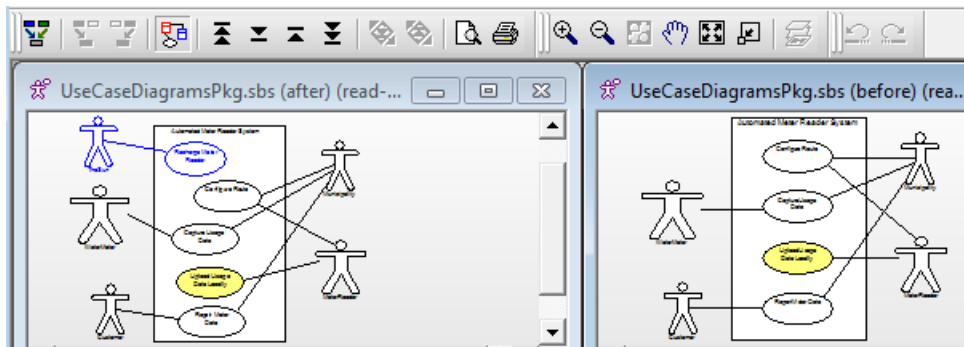


\_\_8. Note that the Rhapsody Diff/Merge can also show graphical differences. Right-click on the *UCD AutomatedMeterReader* diagram and choose **View all diagrams**.

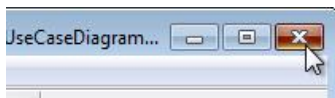




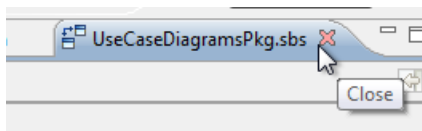
\_\_9. We can see that the changes that *Susan* made to the diagram.



\_\_10. Let's assume that *Pete* is happy with the changes. Close the Rhapsody Diff/Merge.

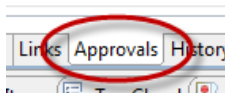


\_\_11. Also close the Diff/Merge launch page by pressing the cross in the top right.



### 6.3 Approving a review

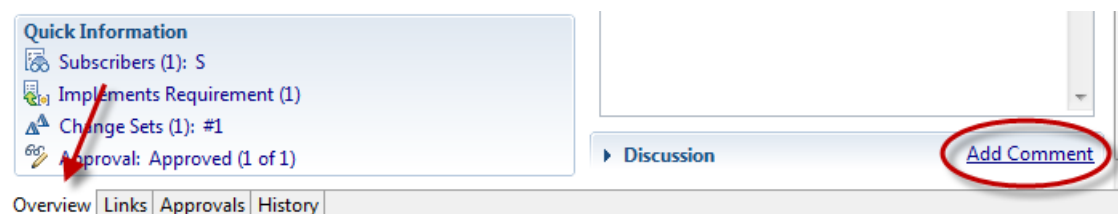
\_\_1. Return to the task and switch back to the **Approvals** tab.



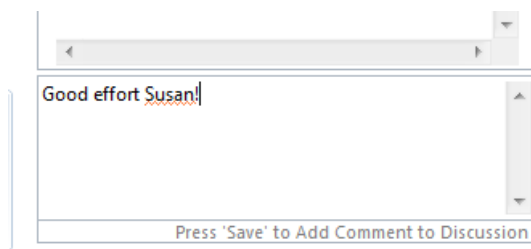
\_\_2. Change the state to **Approved**.

Approver	State
Approval	✓ Approved
Pete	✓ Approved Approved Rejected Pending

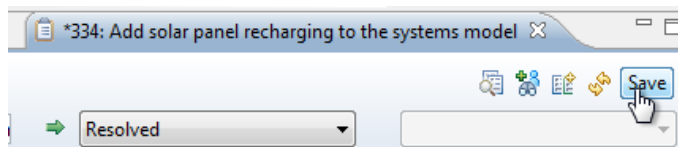
\_\_3. Finally lets add a quick comment. Switch to the *Overview* tab, scroll down and choose **Add Comment**.



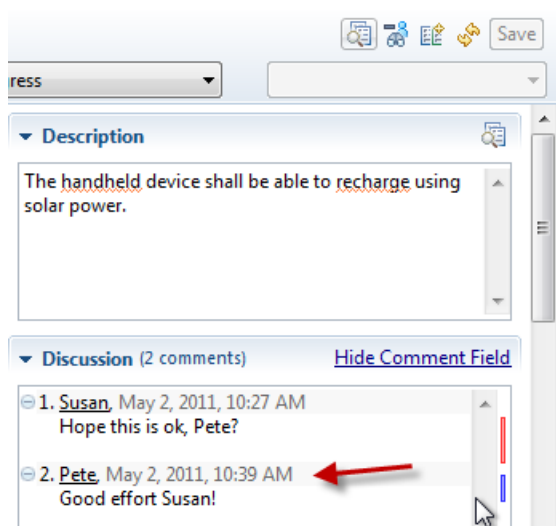
\_\_4. Add a comment, e.g., "Good effort Susan!"




\_\_5. Click the **Save** button.



\_\_6. Verify the comment was added to the discussion.

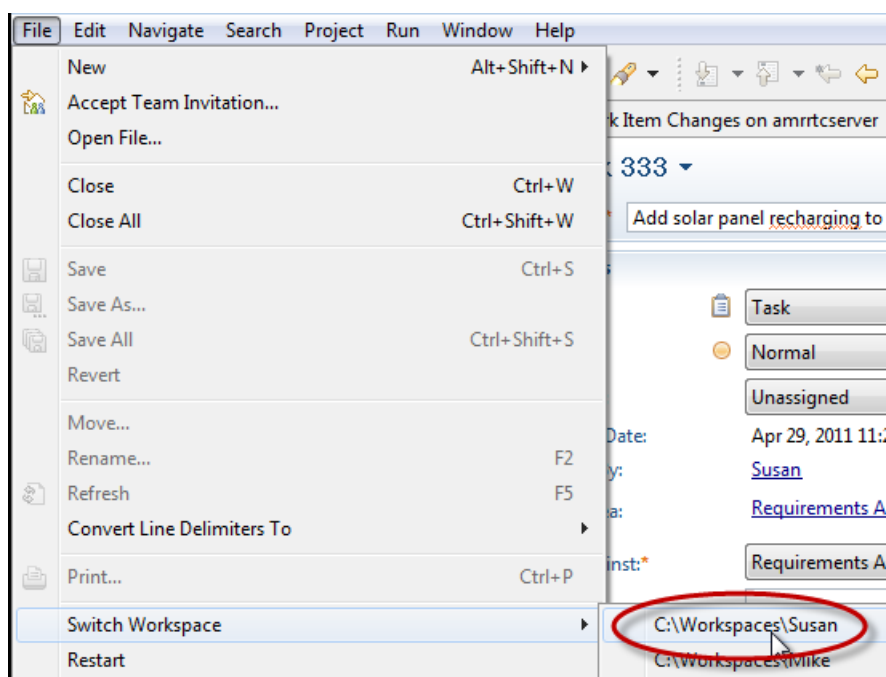


## Lab 7 Delivering the work and reviewing project status

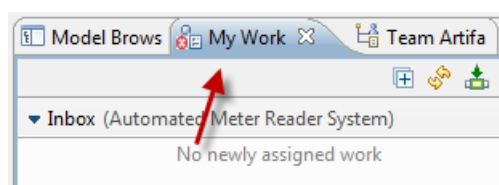
	<p><b>Lab Scenario</b></p> <p>In this lab we explore the key aspects of Rational Team Concert in the context of submitting work to a stream.</p> <p>You will perform the following tasks:</p> <ul style="list-style-type: none"> <li>• Delivering a change set to the stream.</li> <li>• Reviewing the project status</li> </ul> <p><b>Lab Setup</b></p> <p>This lab requires the completion of Labs 1-6.</p>
---	---

### 7.1 Deliver the completed task

- \_\_1. Let's return to Susan's workspace. Choose **File > Switch Workspace > C:\Workspaces\Susan**.

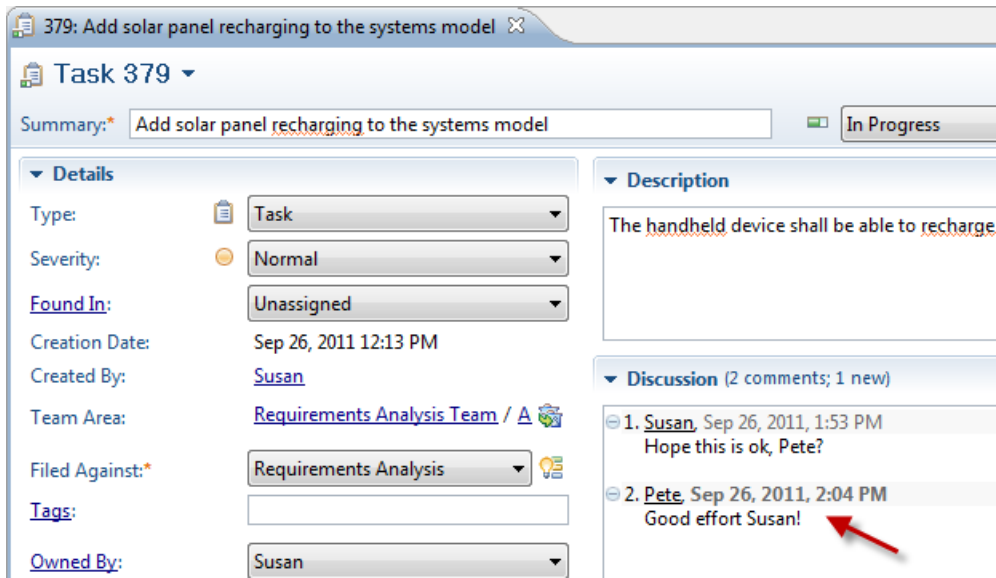


- \_\_2. Switch to the My Work view.

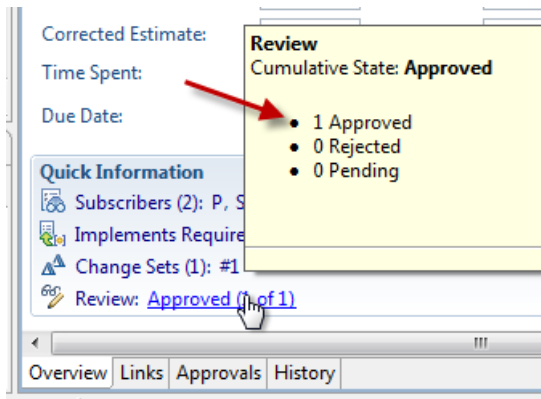


- \_\_3. Find the *Add solar panel recharging...* task in the *My Work* view and double-click to open it up.

\_\_4. Verify that the comment *Pete* added can be seen here.



\_\_5. Look at the bottom and note that the review has been approved.



\_\_6. Click on the *History* tab to see what happened.



**History**

Apr 24, 2011 - Week 18

Sun Mon Tue Wed Thu Fri Sat

Showing 6 of 6 Item

**Today** (6 items found)

- Pete** 5 minutes ago
 


Approvals	Review "Review" state: Pending → Approved Pete: Pending → Approved
-----------	---
- Pete** 10 minutes ago
 

Comments [#2]	Good effort Susan!
---------------	--------------------
- Susan** 20 minutes ago
 

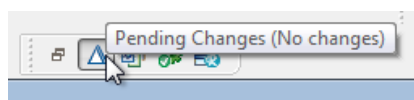
Comments [#1]	Hope this is ok, Pete?
Approvals	added: Review "Review" Pete
Subscribed By	added: Pete
- Susan** 1 hour ago
 

Approvals	added: Review "Review" Pete
-----------	-----------------------------

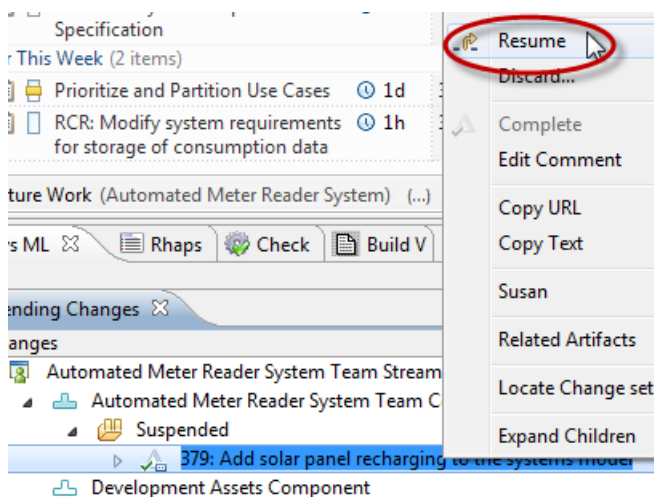
Overview Links Approvals **History**

	<p><b>Benefit</b></p> <p>RTC automatically captures an audit trail. This can be useful for informal as well as for formal quality management purposes.</p>
---	--

\_\_7. Let's try delivering the change again. Click the icon to reopen the *Pending Changes* view.



\_\_8. Choose to **Resume** the change set.



Specification

This Week (2 items)

- Prioritize and Partition Use Cases 1d
- RCR: Modify system requirements for storage of consumption data 1h

ature Work (Automated Meter Reader System) (...)

s ML Rhaps Check Build V

ending Changes

anges

- Automated Meter Reader System Team Stream
  - Automated Meter Reader System Team C
    - Suspended
      - 379: Add solar panel recharging to the systems model

Development Assets Component

Resume

Discard...

Complete

Edit Comment

Copy URL

Copy Text

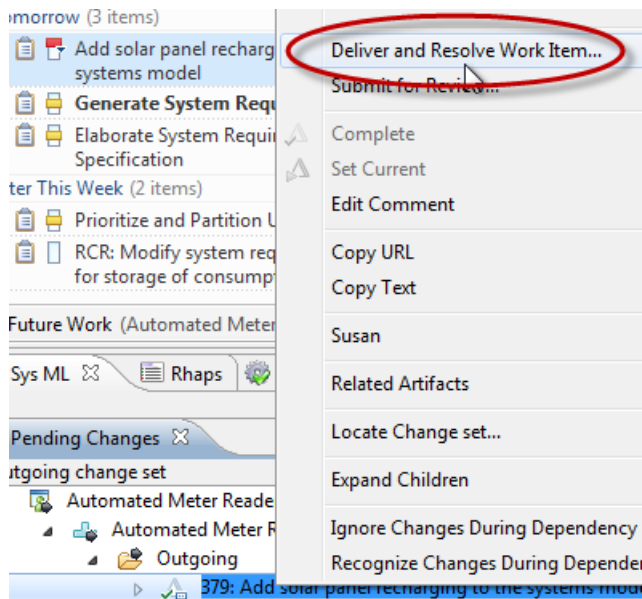
Susan

Related Artifacts

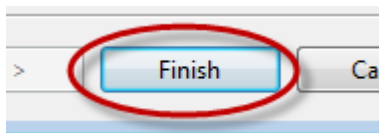
Locate Change set.

Expand Children

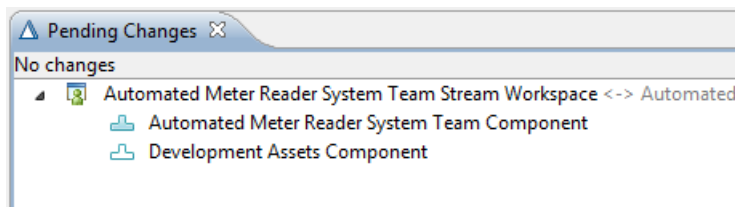
\_\_9. Right-click and choose **Deliver and Resolve Work Item...**



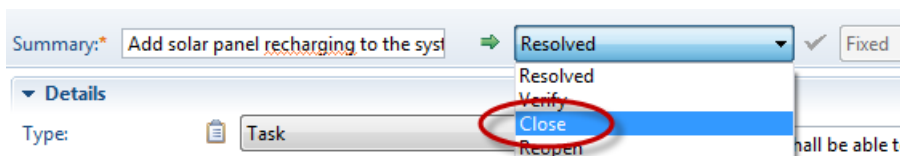
\_\_10. Click **Finish**.



\_\_11. Note in the *Pending Change* view that there are no changes.



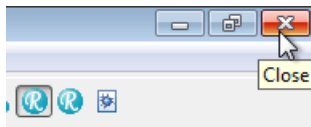
\_\_12. Change the status of the task to Closed (choose **Close** in the pull-down list and then press the **Save** button).



\_\_13. Verify it was set it to a *Closed* state when you saved.

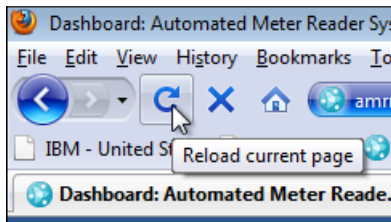


\_\_14. Susan has now finished all the work. You can exit the RTC Eclipse client now.

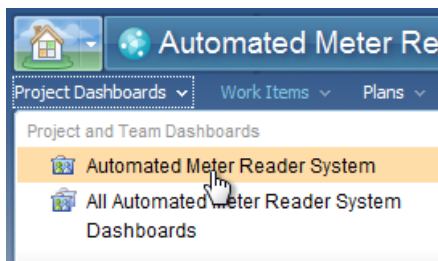


## 7.2 Review the final project status

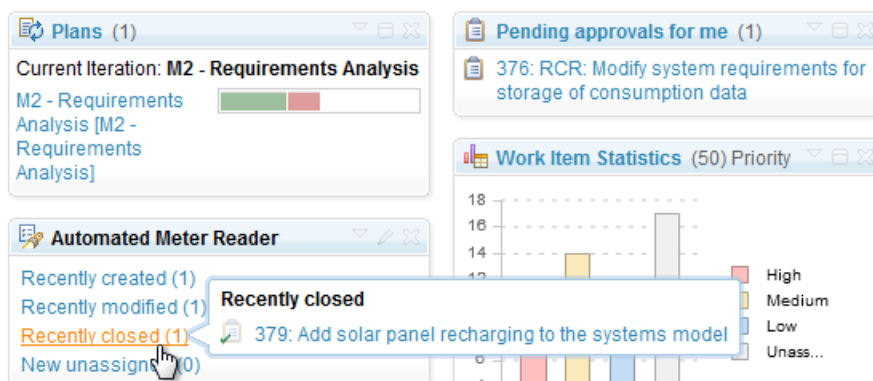
\_\_1. Return to Firefox as Pete and reload the current page.



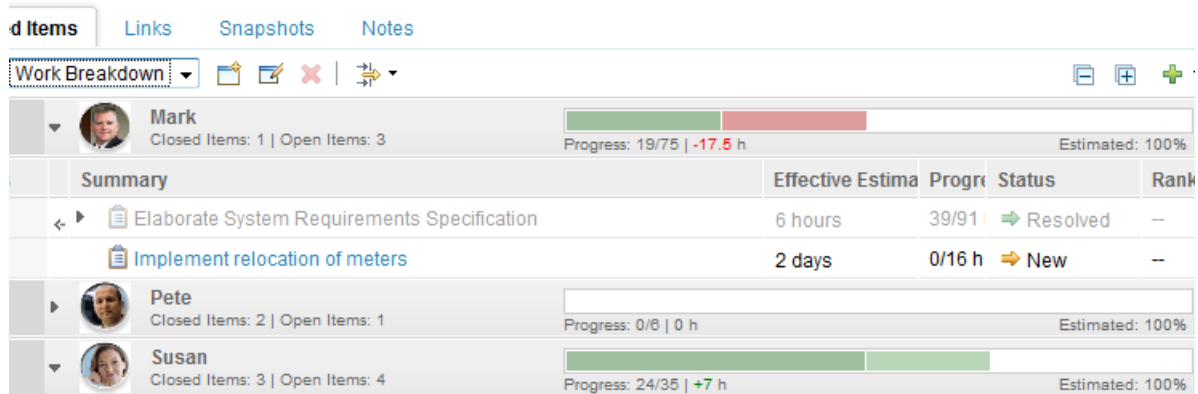
\_\_2. Navigate to the Automated Meter Reader System Project Dashboard.



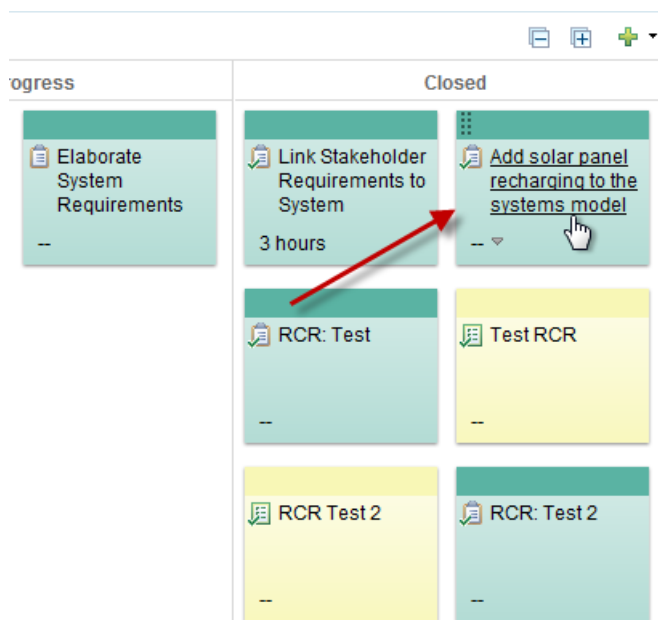
\_\_3. Note that the task is now shown as "Recently closed" in the dashboard and is no longer listed as pending approval.



\_\_4. Navigate to the plan and note that in the Work Breakdown view Susan is still doing pretty well.



\_\_5. Choose **Taskboard** in the View As menu for the plan. Not that we can now see that the task is in a *closed* state.

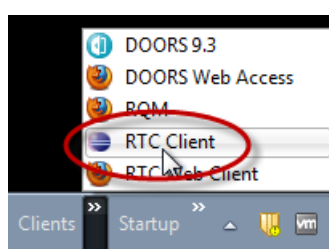


\_\_6. This concludes the formal part of the PoT. Your instructor may be able to show you other aspects of interest using the image, such as the process guidelines and workflows, process templates and how the review process was customized in the process configuration.

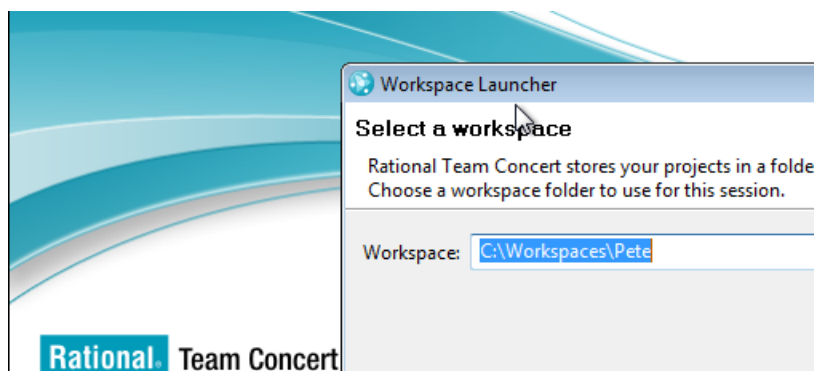


## Appendix A - Updating the iteration dates prior to Lab 1

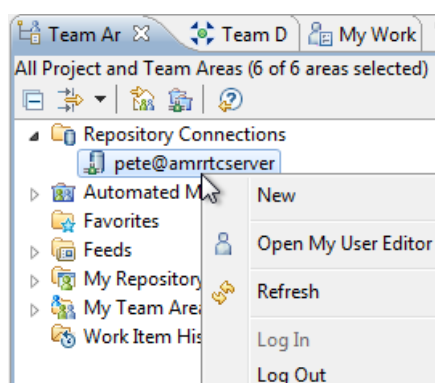
- \_\_1. We need to make sure on that the dates for the iterations are valid. It is important to realize that planning in RTC is using live information and dates with times. Therefore, you may need to repeat this step before doing this PoT or demo in order for the status' of the project to match that described in Lab 1. Essentially this means ensuring that the dates are configured so that you're halfway through a one month iteration.
- \_\_2. The easiest place to update this information is in the RTC Eclipse client. First ensure that the RTC server is running. If not, start the server using the **Startup > RTC server.startup** shortcut in the taskbar.
- \_\_3. Once the server is running, launch the RTC Eclipse client using the **Client > RTC Client** shortcut.



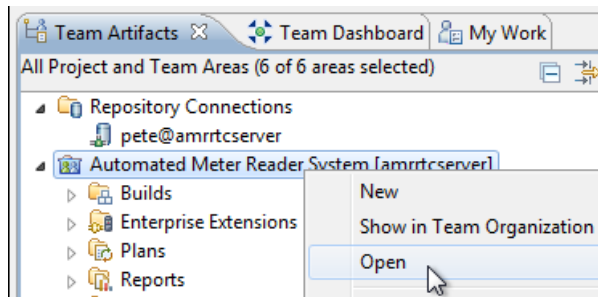
- \_\_4. Ensure that Pete's workspace is selected when prompted. Pete is the project manager.



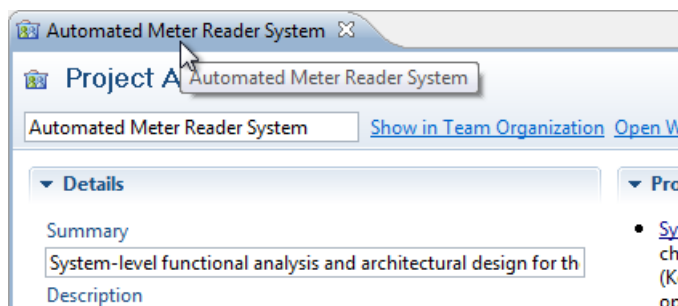
- \_\_5. Note that, if you expand the Repository Connections folder, you can see that Pete is automatically logged in to the *amrrtcserver* Jazz repository.




\_\_6. Right-click on the *Automated Meter Reader System* project area and choose **Open**.

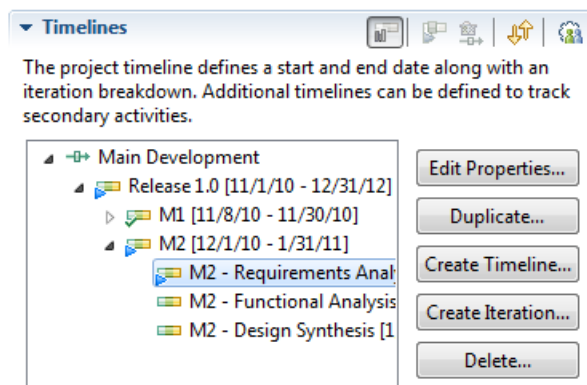


\_\_7. This will open the *Project Area* for the Automated Meter Reader System. Double-click the title bar to expand the Project Area window to full screen.



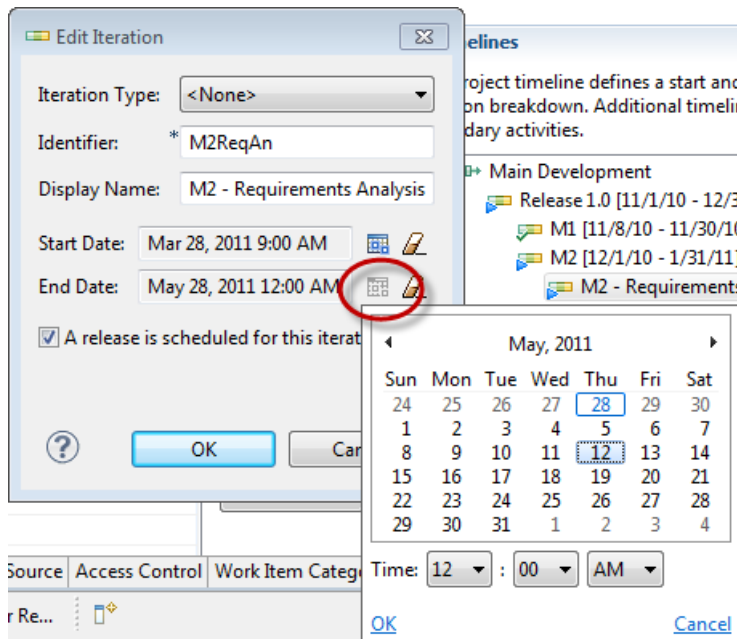
	<p><b>Benefit</b></p> <p>Double-clicking the title bar is a standard mechanism in Eclipse for expanding a window. Double-clicking the title bar again will return you to the docked view.</p>
---	---

\_\_8. Have a look at the dates on the timelines on the *Overview* tab. Notice that they are somewhat in the past.

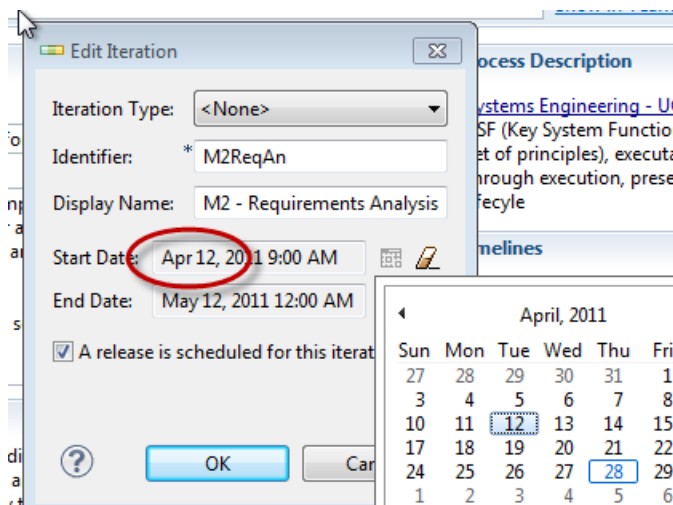


\_\_9. The demo will focus on the *M2 - Requirements Analysis* iteration. Select this and choose **Edit Properties...**

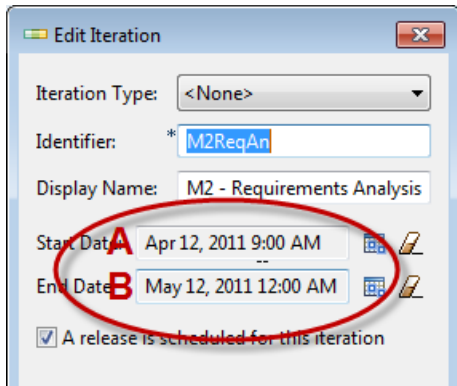
- \_\_10. Set the **End Date** for *M2 - Requirements Analysis* to be two weeks from the current date, i.e. the real date now + 2 weeks (the following screenshot will obviously vary over time).



- \_\_11. Set the **Start Date** for *M2 - Requirements Analysis* to be 1 month before the end date (essentially meaning that we are half way through this iteration).



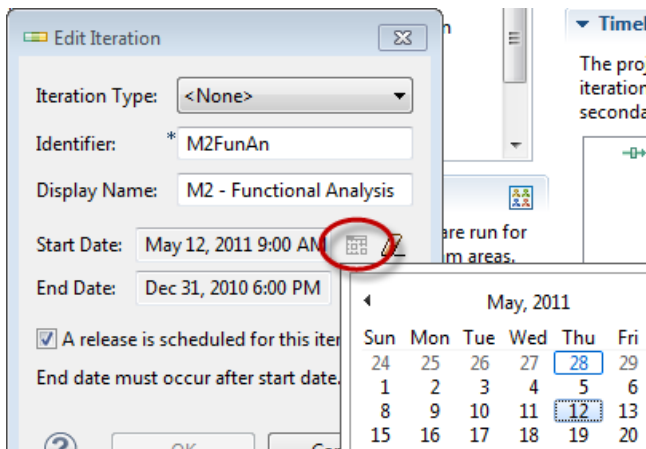
- \_\_\_12. Check the dates for *M2 - Requirements Analysis* are basically set as one month's duration and record the dates below.



A = \_\_\_\_\_

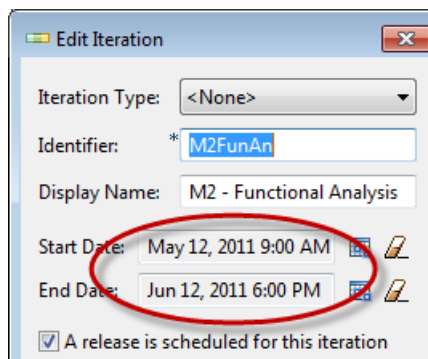
B = \_\_\_\_\_

- \_\_\_13. Set **Start Date** for the *M2 - Functional Analysis* to the end date of the previous iteration B, above and record it below.



C = \_\_\_\_\_

\_\_14. Set the **End Date** for *M2 - Functional analysis* to C + 1 month and record it below.

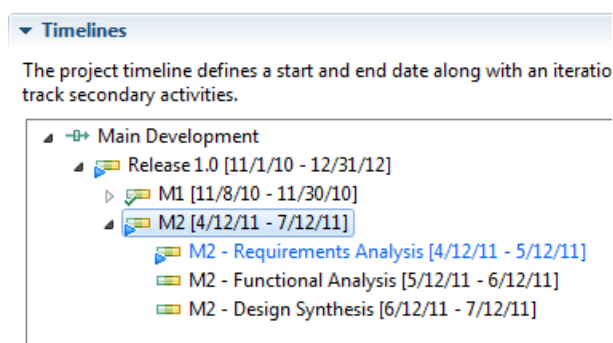


D = \_\_\_\_\_

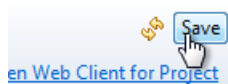
\_\_15. Verify the *M2 - Functional analysis* is one month's duration.

\_\_16. Set the **Start Date** for *M2 - Design Synthesis* to start at date D (the end of M2 - Functional Analysis) and last for one month also.

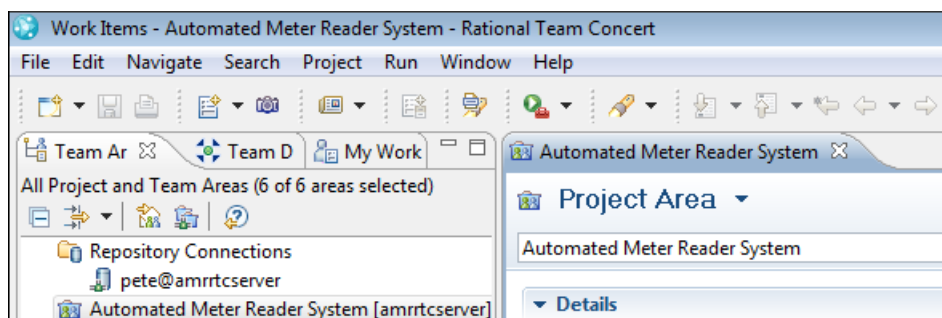
\_\_17. You should now have set up the dates to be more relevant.



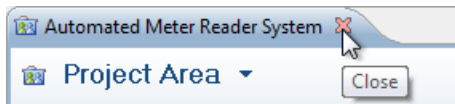
\_\_18. Click **Save** to ensure that the changes are persisted.



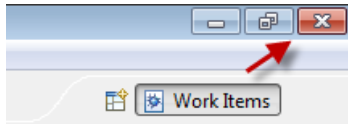
\_\_19. Double-click the title bar to return the window to the docked view.



\_\_20. Close the project area.



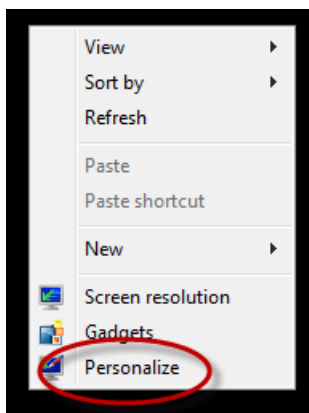
\_\_21. Exit Team Concert.



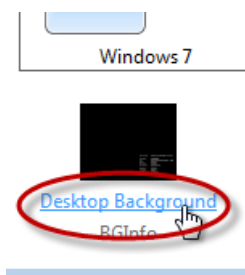
\_\_22. Hit OK to confirm you want to exit. You're now ready to proceed with Lab 1 (you don't need to re-start the RTC server though, hence you can skip the step to do this).

## Appendix B - Changing the Desktop image

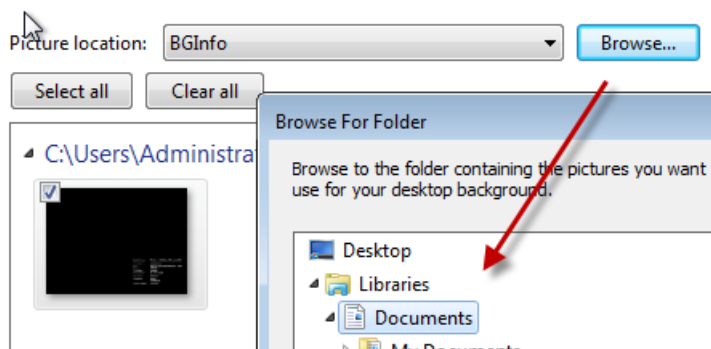
- \_\_1. Optionally, you could display an image of the tools involved in this image on the Desktop. To do this right-click the Desktop and choose **Personalize**.



- \_\_2. Select **Desktop Background** near the bottom.



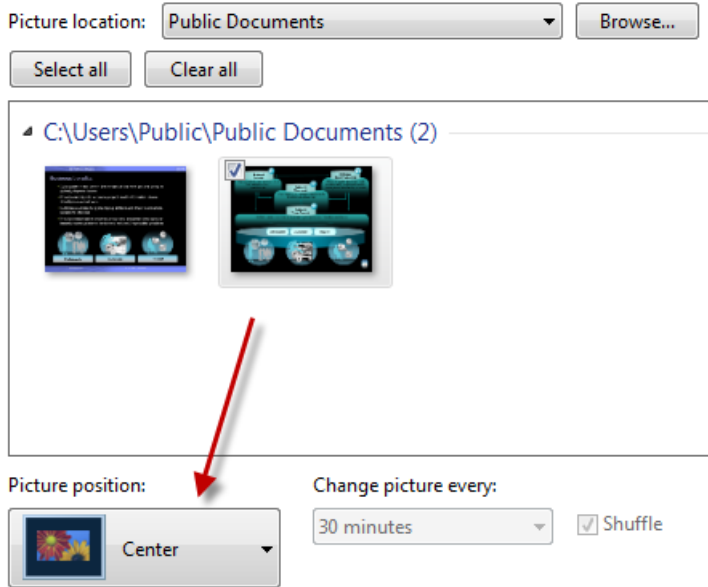
- \_\_3. Click **Browse...** and browse to Libraries > Documents.



\_\_4. Check the second image here and choose **Center**.

### Choose your desktop background

Click a picture to make it your desktop background, or select more than one picture



\_\_5. Click the **Save changes** button.



\_\_6. Close the control panel dialog.

