

Hands-On Lab

Planning your Projects with Team Foundation Server 2010

Lab version: 1.0.0

Last updated: 12/14/2010



Contents

[Overview 3](#_Toc261342487)

[Exercise 1: Managing User Stories 5](#_Toc261342488)

[Exercise 2: Managing Iteration Work Items and Team Capacity 12](#_Toc261342489)

[Exercise 3: Working with Hierarchical Work Items in Visual Studio and Project 23](#_Toc261342490)

[Exercise 4: Introduction to MSF Agile Process Sample Documents 28](#_Toc261342491)

Overview

* 1. In this lab, you will learn about some of the new agile project management capabilities of Team Foundation Server 2010. You will learn how Team Foundation Server 2010 now supports hierarchical work items, which allow for such rich relationships as parent/child and successor/predecessor. You will see how new Excel workbooks can be used to quickly manage your backlog and plan your iterations. And you will also see how Microsoft Project can be used to create a detailed project plan which can then be fully synchronized with Team Foundation Server 2010.

# System Requirements

* 1. In order to complete this lab you will need the Visual Studio 2010 virtual machine provided by Microsoft. For more information on acquiring and using this virtual machine, please see “Working with the Visual Studio 2010 RTM Virtual Machine”.

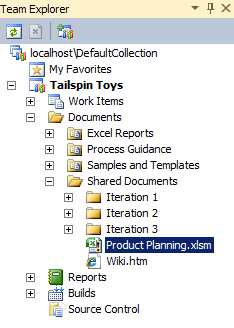
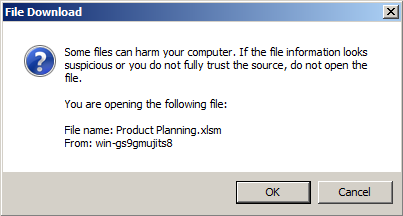
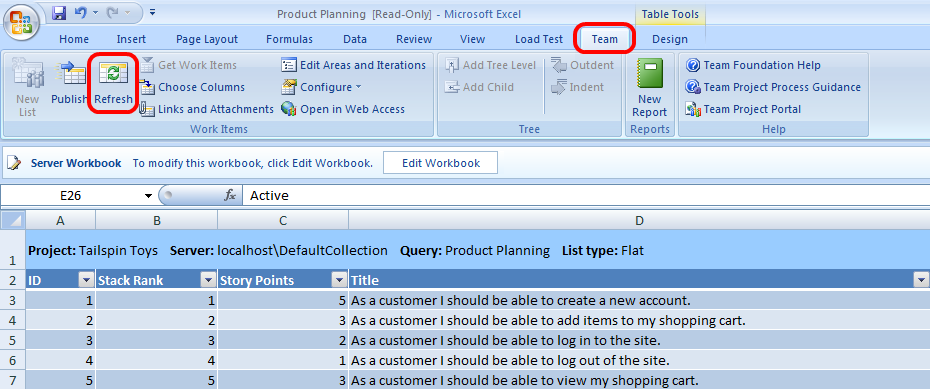
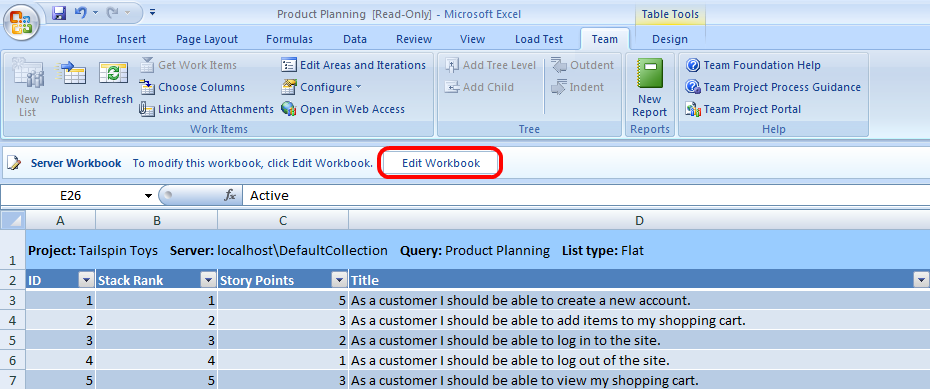
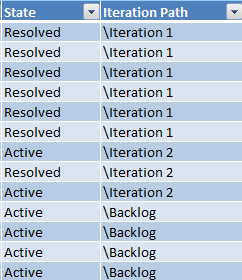
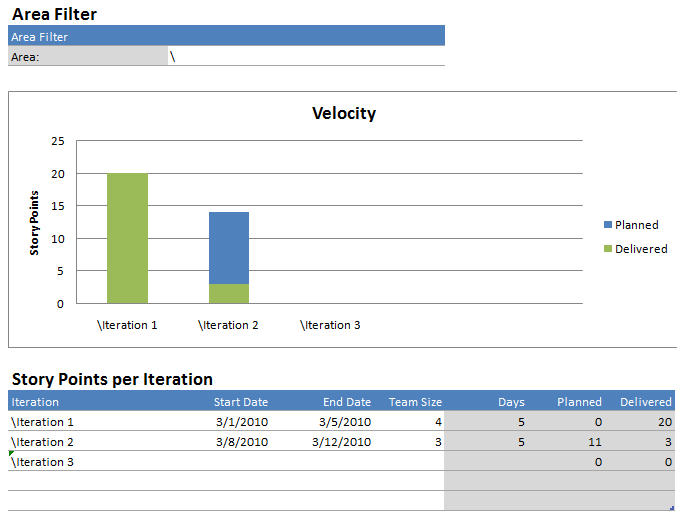
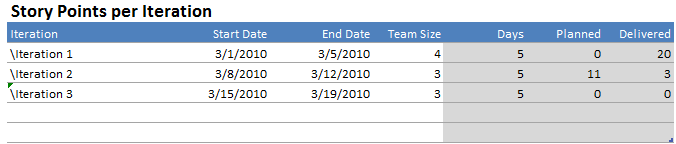
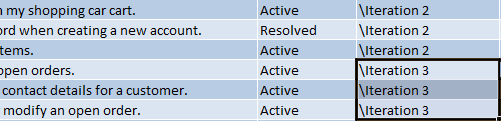
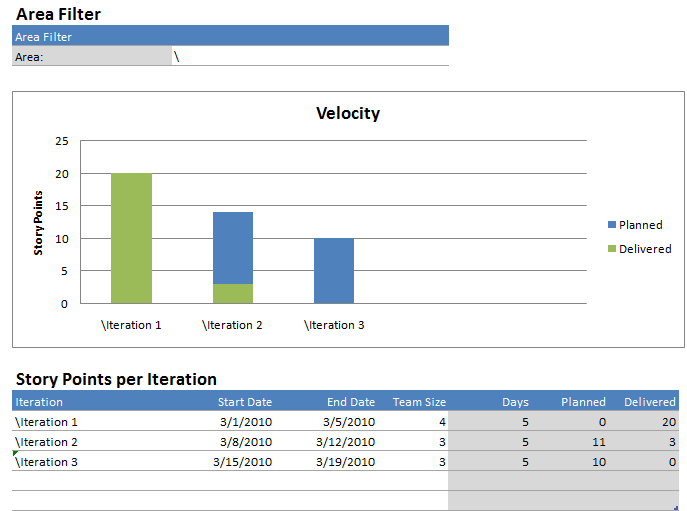
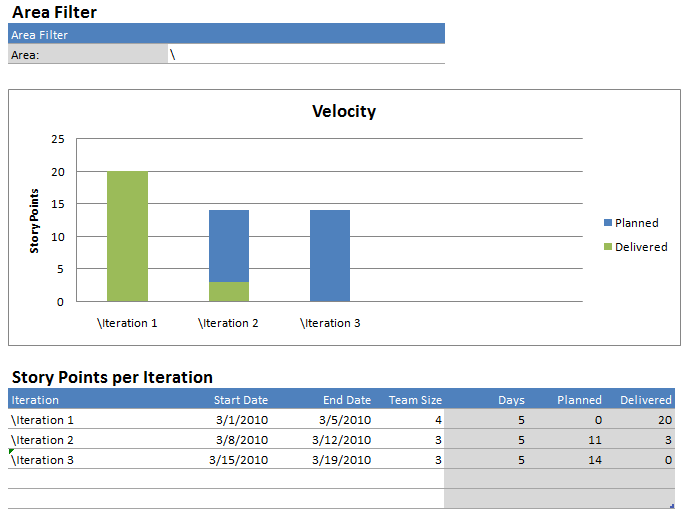
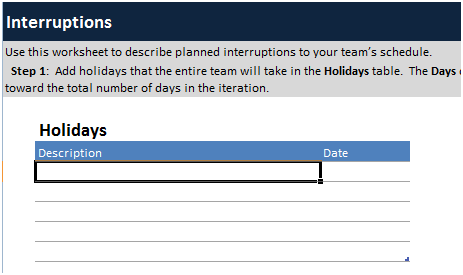
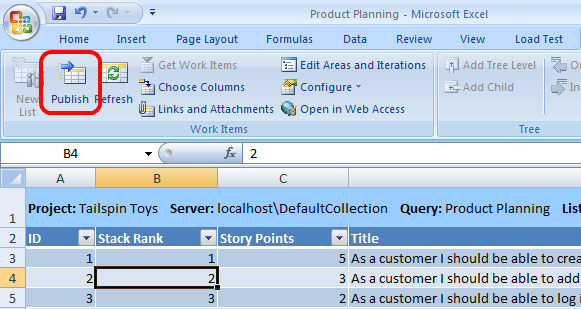
# Exercises

* 1. This Hands-On Lab comprises the following exercises:
  2. Managing User Stories
  3. Iteration Work Items and Team Capacity
  4. Working with Hierarchical Work Items in Visual Studio and Project
  5. Introduction to MSF Agile Process Sample Documents
  6. Estimated time to complete this lab: **60 minutes**.

# Next Step

Exercise 1: Managing User Stories

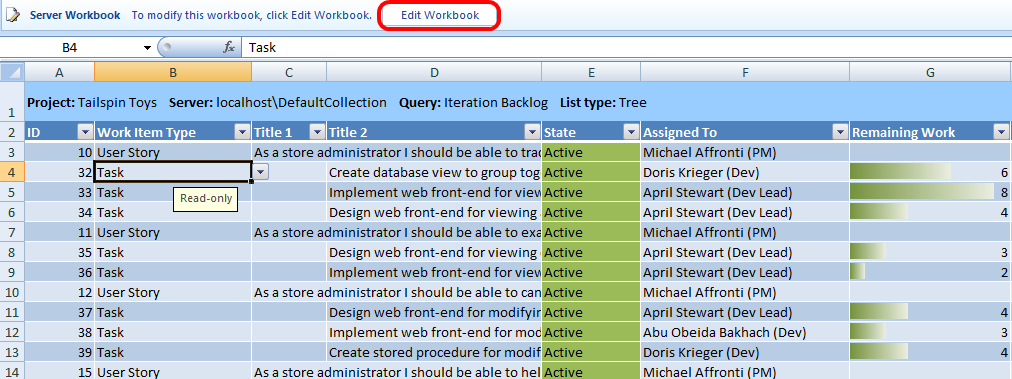
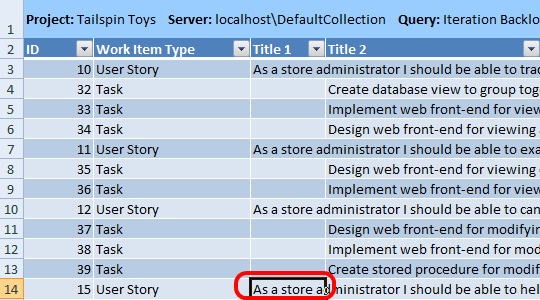
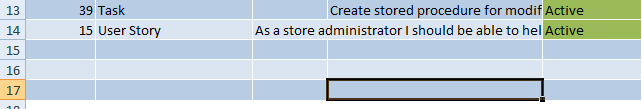
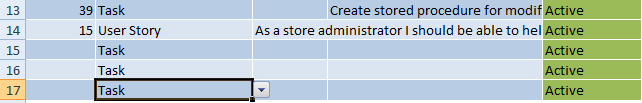
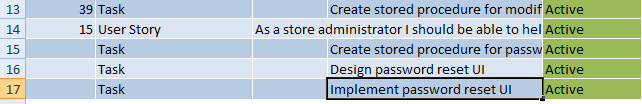
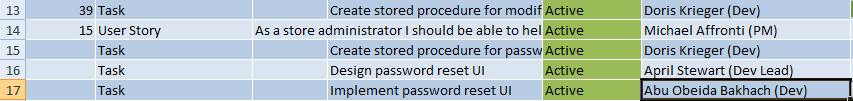
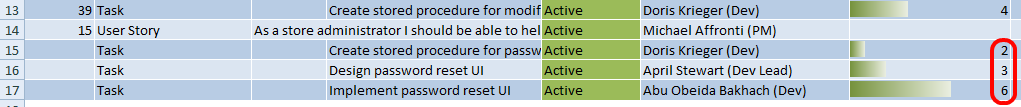
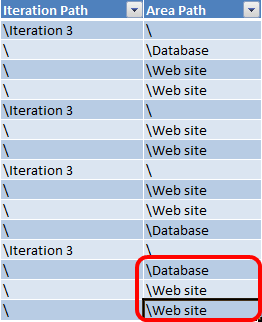
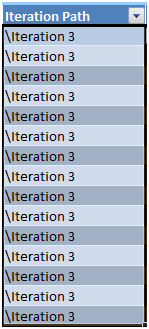
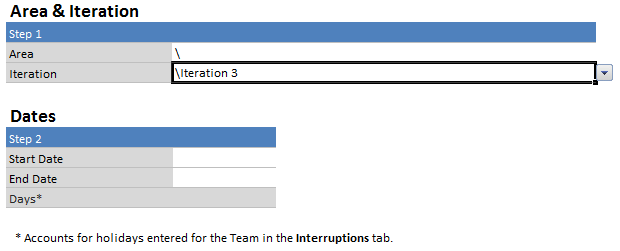
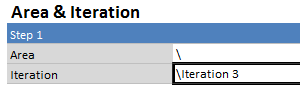
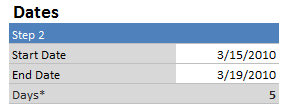
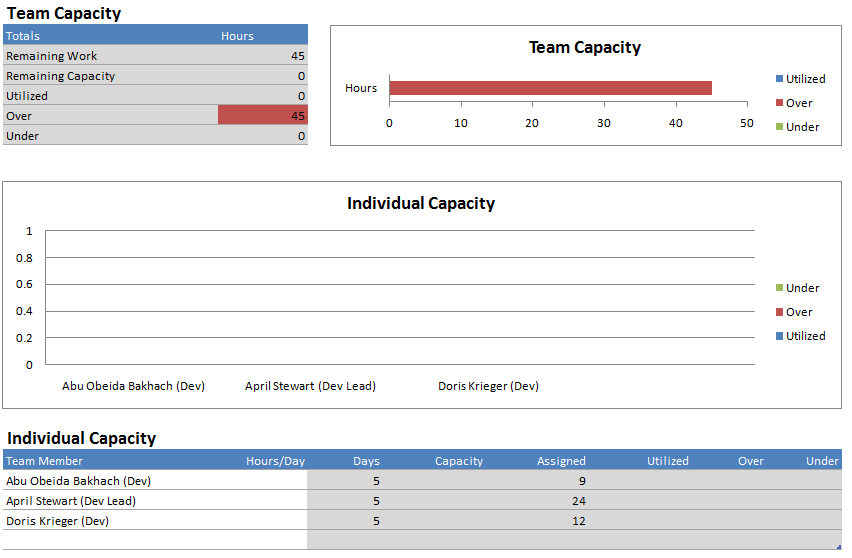
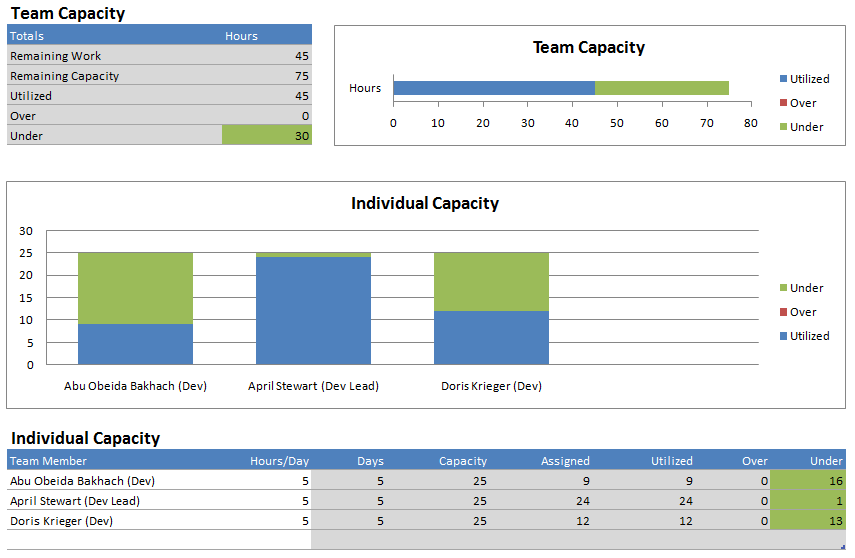
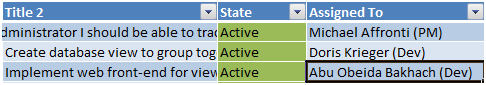
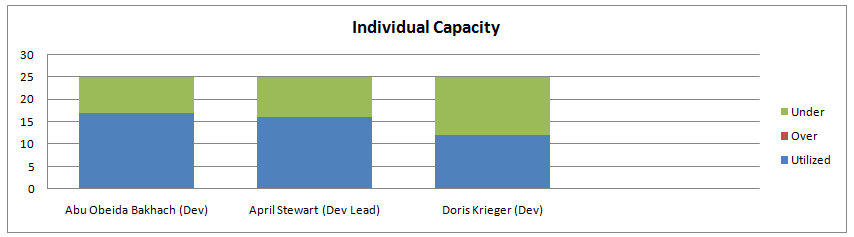
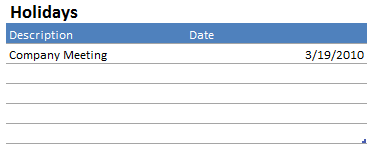
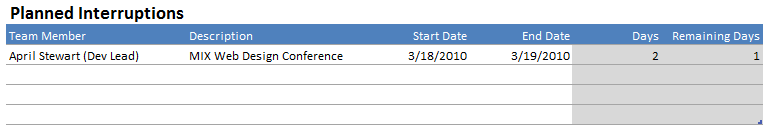
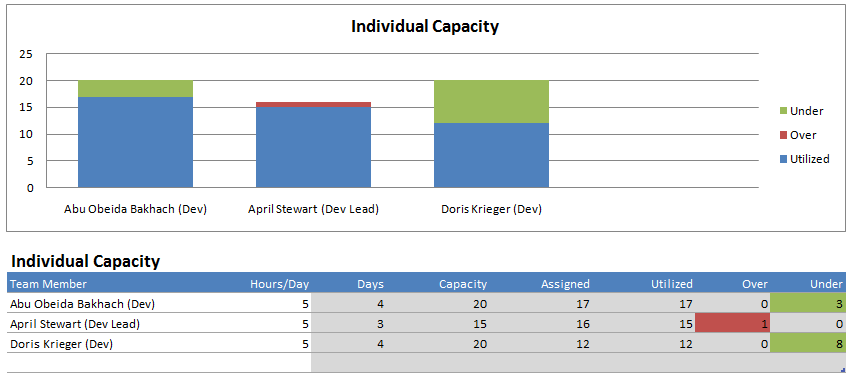
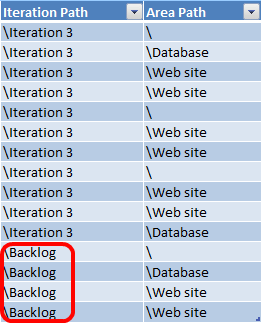
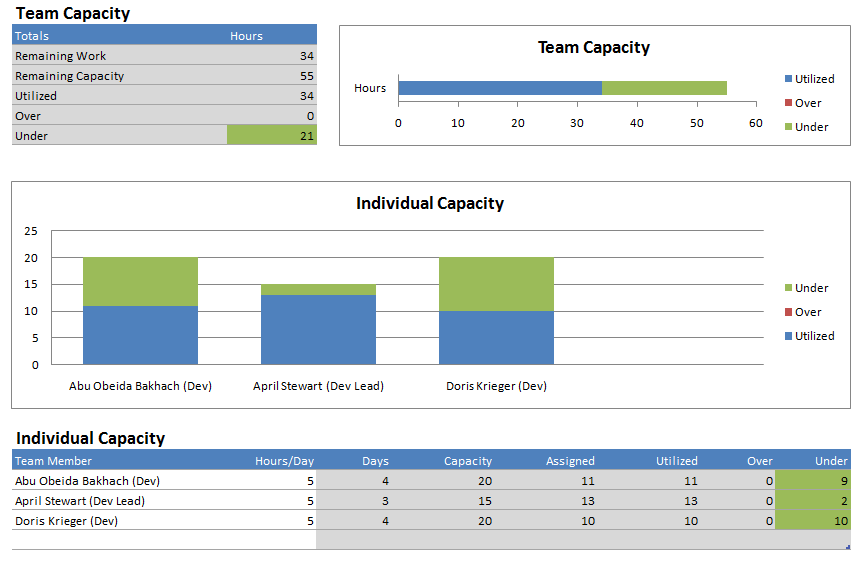
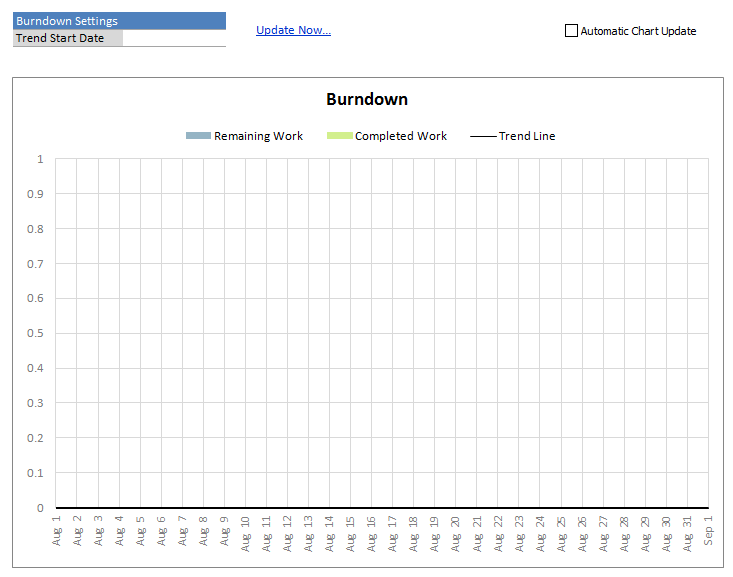
Exercise 1: Managing User Stories

1. In this exercise, you will learn how to use the Product Backlog workbook to work with user story work items from Team Foundation Server. User stories are single sentences that describe actions that the user can take and are typically written when starting a software project.
   1. Log in as **Abu Obeida Bakhach (Dev)** if you have not already done so. The password is P2ssw0rd (capital letter P, the number two, the letter s, the letter s, the letter w, the number zero, the letter r, and the letter d). Please see “Working with the Visual Studio 2010 RTM Virtual Machine” for instructions on how to log into the VM.
   2. Open Microsoft Visual Studio from **Start** | **All Programs** | **Microsoft Visual Studio 2010** | **Microsoft Visual Studio 2010**.
   3. In **Team Explorer**, expand the **Tailspin Toys** project node, followed by the **Documents** and **Shared Documents** folders so that you can see the **Product Planning.xlsm** workbook.
      1. 
      2. Figure
      3. Product Planning workbook location
   4. The new Product Planning workbook enables you to manage your backlog of product work items from within Microsoft Excel. **Double-click** on **Product Planning.xlsm** to open it, selecting the **OK** button when prompted.
      1. 
      2. Figure
      3. File download confirmation dialog box
      4. **Note:** When prompted to enter a product key, select the **Continue** button followed by the **No** button to continue using the trial version. Check the task bar as this prompt may be in the background.
   5. To make sure that the data accurately reflects the current state in Team Foundation Server (TFS), select the **Refresh** button within the **Team** tab in Excel.
      1. 
      2. Figure
      3. Refresh button location
   6. The Product Planning workbook, which shows all of the existing user stories, is initially loaded in read-only mode. Each user story is represented as a work item in TFS. Select the **Edit Workbook** button near the top of the workbook.
      1. 
      2. Figure
      3. Edit Workbook button location
   7. Note that each user story represented here has a state and iteration assigned to it. All of the user stories associated with Iteration 1 have already been completed, the three user stories associated with Iteration 2 are currently in progress or resolved, and the rest are assigned to the backlog.
      1. 
      2. Figure
      3. Initial state for State and Iteration Path columns
   8. Select the **Iterations** worksheet (tab near bottom of workbook). This worksheet helps define iterations and balance the workload between those iterations. The **Planned** and **Delivered** columns represent the quantity of work that we planned to complete and the quantity of work actually completed during the iteration, where work is measured in story points. While planning a new iteration, we can take into account the planned and delivered work from previous iterations in order to help assign an appropriate level of work for future iterations.
      1. 
      2. Figure
      3. Initial state of Iteration planning worksheet
   9. Now let’s start planning a new iteration. Under the **Story Points per Iteration** heading, find the row for Iteration 3 and enter a **Start Date** for next Monday. It is important that this date be in the future. Enter the **End Date** for the end of that week (the following Friday). Finally, enter **3** for team size.
      1. 
      2. Figure
      3. Adding Iteration 3
   10. The next step is to start assigning active customer story tasks to the new iteration. Return to the **Product Backlog** worksheet and assign the first three Backlog items to **Iteration 3**. Note that each customer story has story points assigned to it. These points, which represent a rough order of magnitude estimate for the amount of work anticipated, are used later on to help load balance the planned development iterations.
       1. 
       2. Figure
       3. Backlog work items changed to Iteration 3
   11. Switch back to the **Iterations** worksheet to see what the workload is looking like in comparison with previous iterations. The three user story tasks that we added to the new iteration account for **10** story points, which is less than the 14 that were scheduled with the team size for Iteration 2, and less than the 20 that were delivered on with a slightly larger team during Iteration 1.
       1. 
       2. Figure
       3. Iteration Planning worksheet showing Iteration 3 work load
   12. It looks like we have more capacity, so let’s assign some more work to Iteration 3. Return to the Product Backlog worksheet and find the backlogged user story with an **ID** of **15**. This user story is worth **4** story points, which would bring the total story points for Iteration 3 up to where we want it. Note that some other backlogged user stories were skipped over because their story points indicate that they would be too much work for this iteration.
   13. Change the **Iteration Path** for user story **ID = 15** to **Iteration 3**.
   14. Switch back to the Iterations worksheet and note that Iteration 3 does indeed show that we are taking on the right amount of work considering the story points, size of team, and length of iteration.
       1. 
       2. Figure
       3. Iteration Planning worksheet showing updated Iteration 3 work load
   15. Note that the **Product Planning** workbook also has an **Interruptions** worksheet that allows you to account for holidays or other day-long events for the entire team. This data is represented in the **Days** column of the **Iterations** worksheet if it falls within an iteration date range.
       1. 
       2. Figure
       3. Interruptions worksheet for Product Planning workbook
   16. Now that we have modified the user stories and completed our new iteration planning, we should save the changes to TFS. Return to the **Product Backlog** worksheet, click anywhere in the table to put the cursor there, and then select **Publish** from the Team tab in Excel.
       1. 
       2. Figure
       3. Publish button location
       4. **Note:** It may take a moment for the changes to be published to TFS.
   17. Now that we are done working with the Product Planning workbook, close it and **save** the changes when prompted.

# Next Step

Exercise 2: Managing Iteration Work Items and Team Capacity

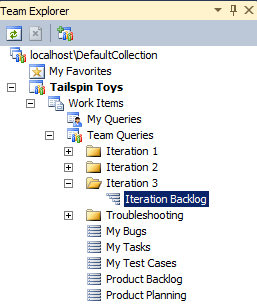
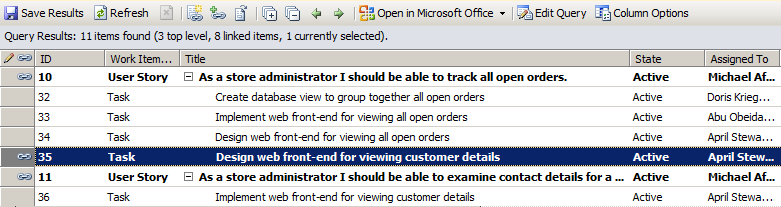
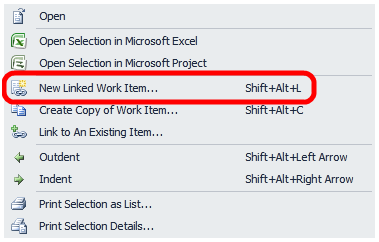
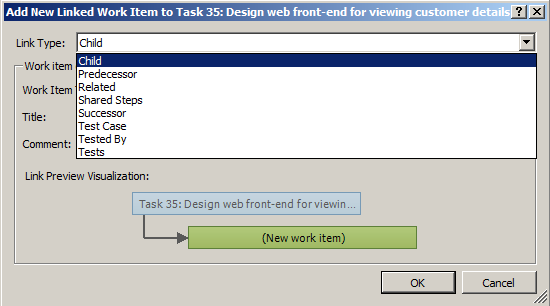
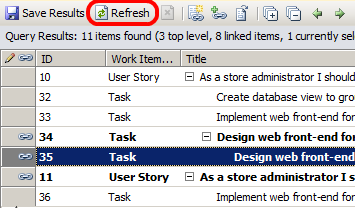
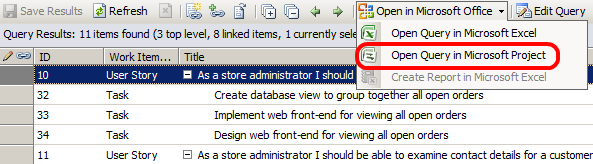
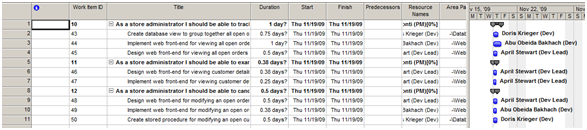
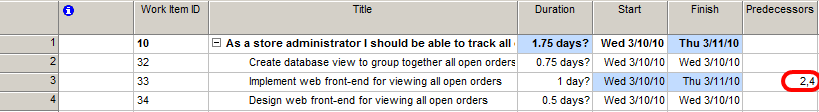
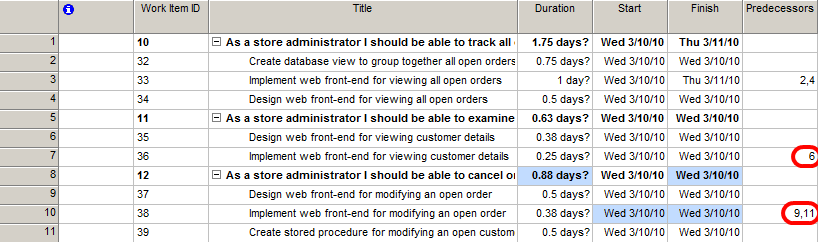
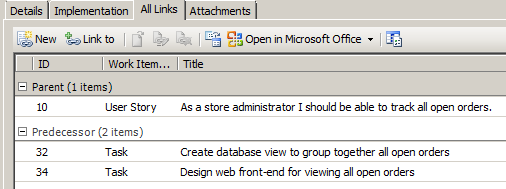
Exercise 2: Managing Iteration Work Items and Team Capacity

* 1. In this exercise, you will learn how to take a user story and break it down into individual tasks that can be assigned to appropriate team members based on skill and available work capacity. This will all be done within the context of Iteration 3.
  2. In **Team Explorer**, navigate to the **Iteration 3 Backlog.xlsm** workbook under **Tailspin Toys | Documents | Shared Documents | Iteration 3**. **Double-click** on **Iteration 3 Backlog.xlsm** to open it Excel, selecting the **OK** button when prompted.
  3. To make sure that the data accurately reflects the current state in Team Foundation Server (TFS), select the **Refresh** button within the **Team** tab in Excel.
  4. Select the **Edit Workbook** button to exit read-only mode.
     1. 
     2. Figure
     3. Edit Workbook button location
  5. This worksheet shows a hierarchical view of the work items that are currently associated with Iteration 3. For example, the first user story item shows that there are three associated child tasks under it. Hierarchical work item relationships, which are new to Team Foundation Server 2010, include parent/child, successor/predecessor, tests/tested by, etc.
  6. The first three user stories listed have already been assigned child tasks and delegated to available employees. Now we will flesh out the last user story by adding additional tasks. Select the cell associated with the last User Story work item (**ID = 15**) and the **Title 1** column.
     1. 
     2. Figure
     3. Selecting the title for the last user story
  7. Under the Team tab, select the **Add Child** button **3** times. You should now see three new placeholder rows appear.
     1. 
     2. Figure
     3. Adding three new work item children
  8. Change the **Work Item Type** for each new row to **Task**.
     1. 
     2. Figure
     3. Changing work item type to Task
  9. In the **Title 2** column for the first new task, enter “**Create stored procedure for password reset**”. For the second task enter “**Design password reset UI**”, and finally for the last task enter “**Implement password reset UI**”. Entering the new titles under the Title 2 column will make them children of the user story above them.
     1. 
     2. Figure
     3. Adding titles for new task work items
  10. Now we will assign the new tasks to appropriate team members. Assign the first new task to **Doris**, the second to **April**, and the third to **Abu**.
      1. 
      2. Figure
      3. Assigning team members to task work items
  11. The **Remaining Work** column represents the number of hours remaining for the task. Enter **2**, **3**, and **6** the new tasks.
      1. 
      2. Figure
      3. Assigning remaining work hours to task work items
  12. Scroll over to the **Area Path** column and enter **Database**, **Web site**, and **Web site** for the new tasks.
      1. 
      2. Figure
      3. Changing development area for task work items
  13. Finally, assign all tasks to Iteration 3, even the tasks that were already in place. The easiest method to accomplish this is to copy the first cell (Ctrl+C) that shows Interation 3, select the remaining cells in the Iteration Path column, and then paste (Ctrl+V).
      1. 
      2. Figure
      3. Changing task work items to Iteration 3
  14. Place the cursor somewhere within the table and then select **Publish** from the Team tab in Excel. This will create the new tasks as children of the last user story within TFS.
  15. Now that we have defined the tasks that make up the last user story, we will spend some time making sure that the right team members are assigned to each task and make sure we have not taken on too much work. Start by navigating to the **Settings** worksheet.
      1. 
      2. Figure
      3. Initial view of Settings worksheet
  16. For Iteration, select **Iteration 3**.
      1. 
      2. Figure
      3. Selecting Iteration 3
  17. For **Start Date** and **End Date**, enter the same dates that you did in the previous exercise. This should be next Monday through Friday.
      1. 
      2. Figure
      3. Entering Start and End Date
  18. Go to the **Capacity** worksheet, find the **Individual Capacity** section, and select the three team members that will be working on tasks for Iteration 3, namely **Abu**, **April**, and **Doris**. We don’t select Michael because he is the project manager and is just responsible for the user stories.
      1. 
      2. Figure
      3. Entering team members assigned to Iteration 3
  19. Under the **Hours/Day** column, which represents the estimated hours per day that each team member will devote to this project, enter **5** for all three team members.
      1. 
      2. Figure
      3. Current team and individual capacity for Iteration 3
  20. Note that the Individual Capacity chart shows that Abu and Doris are under-utilized, whereas April is near max work capacity. Let’s offload some of April’s work to Abu. Return to the **Iteration Backlog** worksheet and re-assign the “Implement web front-end for viewing all open orders” (ID = 33) from **April** to **Abu**.
      1. 
      2. Figure
      3. Re-distributing work between team members
  21. Switch back to the **Capacity** worksheet to see that the workload distribution looks better.
      1. 
      2. Figure
      3. Individual capacity graph looks good
  22. This workbook also allows you to account for holidays and other work interruptions that have an effect on team capacity. Open the **Interruptions** worksheet and add an entry under Holidays for an upcoming company meeting next Friday. This will apply to the entire team.
      1. 
      2. Figure
      3. Team interruption
  23. Under **Planned Interruptions**, add an entry for **April** as she will be attending the **MIX Web Design Conference** next Thursday and Friday. Note that the **Days** column is the expected 2 days. The **Remaining Days** column is 1 because it is already taking into account the company meeting that applies to the team.
      1. 
      2. Figure
      3. Individual interruption
  24. Switch back to the **Capacity** worksheet to see the effect of these scheduled interruptions. Note that April is now over capacity.
      1. 
      2. Figure
      3. Individual capacity showing overworked team member
  25. One potential remedy to offload some work from April would be to shift some of it to Doris, who currently has some extra capacity. Rather than doing that this time, consider the scenario where Doris does not have the skills necessary to take on the work that is assigned to April. Return to the **Iteration Backlog** worksheet and change the last four tasks (corresponding to the last four rows) so that their **Iteration Path** is set to the **Backlog**.
      1. 
      2. Figure
      3. Returning tasks to the backlog
  26. Now that we have deferred some work to the backlog, the team should be able to complete the assigned tasks with the available capacity. Return to the **Capacity** worksheet one last time to verify this.
      1. 
      2. Figure
      3. Team and individual capacity looks good
  27. Save the changes made to the work items by returning to the **Iteration Backlog** worksheet, placing the cursor somewhere within the table, and selecting the **Publish** button.
      1. **Note:** The Burndown worksheet contains a chart that you can use to track work progress during the iteration. As tasks are completed, you can come here to get an idea of iteration progress.
      2. 
      3. Figure
      4. Burndown worksheet
  28. Close Excel and save changes when prompted to return to Visual Studio.

# Next Step

Exercise 3: Working with Hierarchical Work Items in Visual Studio and Project

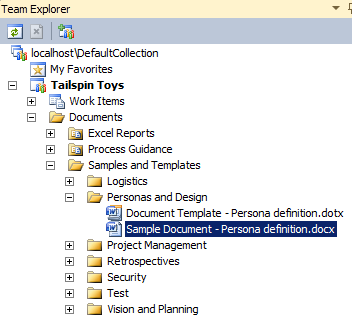
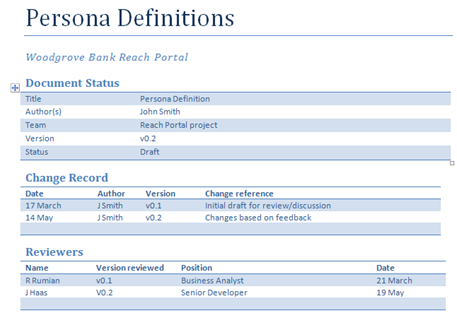
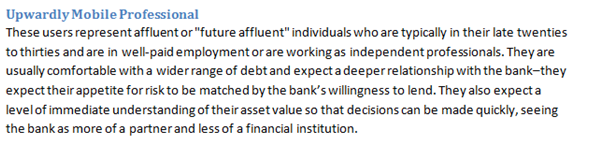
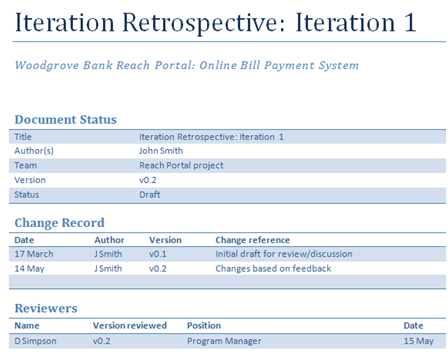
Exercise 3: Working with Hierarchical Work Items in Visual Studio and Project

* 1. In this exercise, you will learn how to manage hierarchical work items from within Visual Studio and get a taste of how to take advantage of Project for more detailed project management.
  2. It is possible to work with the same iteration tasks that we did in the previous exercise from within Visual Studio. In **Team Explorer**, navigate to **Work Items | Team Queries | Iteration 3** and **double-click** on the **Iteration Backlog**.
     1. 
     2. Figure
     3. Iteration Backlog query for iteration 3
  3. There are a number of beneficial UI mechanisms that make working with these work items easier from within Visual Studio. Try clicking the **+** and **–** buttons associated with the user story work items to see that they can collapse and expand if they contain children.
  4. Drag and drop a task from one user story to a different user story. Notice that the changes are displayed in bold.
     1. 
     2. Figure
     3. Changes are displayed in bold
  5. Select a child task with a single left-click and then use the green Indent arrow to change the hierarchy level.
     1. 
     2. Figure
     3. Indent arrow location
  6. To add additional nested work items that may use any of the supported link types, **right-click** on an existing task (it does not matter which one) and select **New Linked Work Item** from the context menu.
     1. 
     2. Figure
     3. New Linked Work Item…
  7. Expand the **Link Type** drop down to see all of the supported link types.
     1. 
     2. Figure
     3. Supported link types
  8. We are not going to save these changes for the purposes of this lab, so select the **Cancel** button to return to the Iteration 3 work items.
  9. Select the **Refresh** button to return to the original list of work items, selecting **No** when prompted to save the modified work items.
     1. 
     2. Figure
     3. Refresh button location
  10. Microsoft Project can also be used to work with TFS work items in order to create a more detailed project plan. Select the **Open in Microsoft Office** drop down and click on the **Open Query in Microsoft Project** option.
      1. 
      2. Figure
      3. Open Query in Microsoft Project
  11. After Project loads, re-adjust the screen real estate so that you can clearly see the Titles of the work items. Note that this data is used to create a Gantt chart in the right-hand pane.
      1. 
      2. Figure
      3. Initial view in Microsoft Project
  12. TFS 2010 also introduces support for successor/predecessor relationship between work items. Select the **Predecessors** column for work item with **ID=33** and enter **2,4** into that cell. This designates that work items with **ID=32** and **34** must be completed first. The numbers **2** and **4** that you entered correspond to the row numbers within Project.
      1. 
      2. Figure
      3. Editing work item successor/predecessor relationship
  13. Add some more successor/predecessor relationships by making work item **ID=36** have a predecessor of work item ID=35 by entering **6** into the **Predecessors** column. Also make work item **ID=38** have predecessors for work items with IDs 37 and 39.
      1. 
      2. Figure
      3. Final view of Predecessors column
  14. **Note:** Although it is outside the scope of this lab to do so, you can use Microsoft Project to create much more detailed project plans by defining additional start and end dates for each work item.
  15. Save changes back to TFS by selecting a cell from the work item table and then select the **Publish** button.
  16. Return to Visual Studio and **double-click** on the work item task with **ID=33** titled “Implement web front-end for viewing all open orders”.
  17. Select the **All Links** tab to verify that work items 32 and 34 are listed as predecessors of this work item task.
      1. 
      2. Figure
      3. All Links tab for work item showing predecessors

# Next Step

Exercise 4: Introduction to MSF Agile Process Sample Documents

Exercise 4: Introduction to MSF Agile Process Sample Documents

* 1. In this exercise, you will be introduced to two of the sample documents that are provided with the MSF Agile Process Template.
  2. In **Team Explorer**, navigate to the **Documents | Samples and Templates | Personas and Design** folder and **double-click** on **Sample Document – Persona Definition.docx** to open it in Microsoft Word. You may need to confirm that you want to download the file as before.
     1. 
     2. Figure
     3. Persona definition sample location
  3. Take a quick look through the Persona Definition sample to get an idea of what it offers.
     1. 
     2. Figure
     3. Start of persona definition sample document
     4. Personas exist as a way to improve the design of software systems that will be used by people from different backgrounds and with different needs, experience, and skill levels. This document contains persona definitions for a sample project.
     5. 
     6. Figure
     7. Example persona
  4. Switch back to Team Explorer and open Sample Document – Iteration Retrospective.docx from the Retrospectives folder.
     1. 
     2. Figure
     3. Start of iteration retrospective sample document
     4. This document is used to record the discussions and action items from the Iteration Retrospective. Iteration Retrospective is held at the end of each iteration to reflect on how the team performed during the iteration. This sample document contains iteration goals, progress statistics, what went well and what didn’t, and finally actions for improvement.

To give feedback please write to [VSKitFdbk@Microsoft.com](mailto:VSKitFdbk@Microsoft.com)

Copyright © 2010 by Microsoft Corporation. All rights reserved.