Introduction to Team Development using Visual Studio Online

Overview

In this lab, you'll learn about some of the Visual Studio Online features that can help software development teams quickly plan, manage, and track work across the entire team. In addition, you'll be introduced to some of the features that help enhance communication and collaboration throughout the project lifecycle, helping teams to reach their full potential.

Objectives

In this hands-on lab, you will learn how to do the following using Visual Studio Online:

- Create a new account and team project
- Administer projects, team members and permissions
- Schedule and manage project resources
- Manage project requirements, tasks, and other work
- Collaborate and communicate

Prerequisites

The following are required to complete this hands-on lab:

- Microsoft Visual Studio 2013 (with Update 2 RC applied)

Notes

Estimated time to complete this lab: **60** minutes.

Note: You can log into the virtual machine with user name "**User**" and password "**P2ssw0rd**".

Note: This lab may make references to code and other assets that are needed to complete the exercises. You can find these assets on the desktop in a folder named **TechEd 2014**. Within that folder, you will find additional folders that match the name of the lab you are working on.

Exercise 1: Getting Started with Visual Studio Online

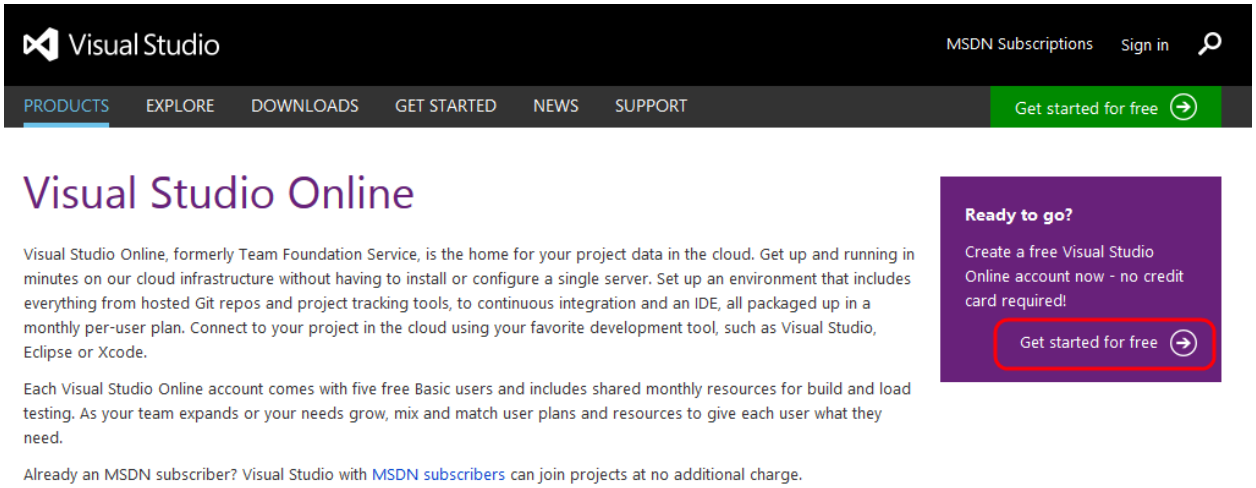
In this exercise, you'll walk through the process of setting up a free Visual Studio Online account and creating a new team project. If you already have a VSO account that you would like to use for this exercise, go ahead and log in and then skip to the next task.

Task 1: Creating a Visual Studio Online account

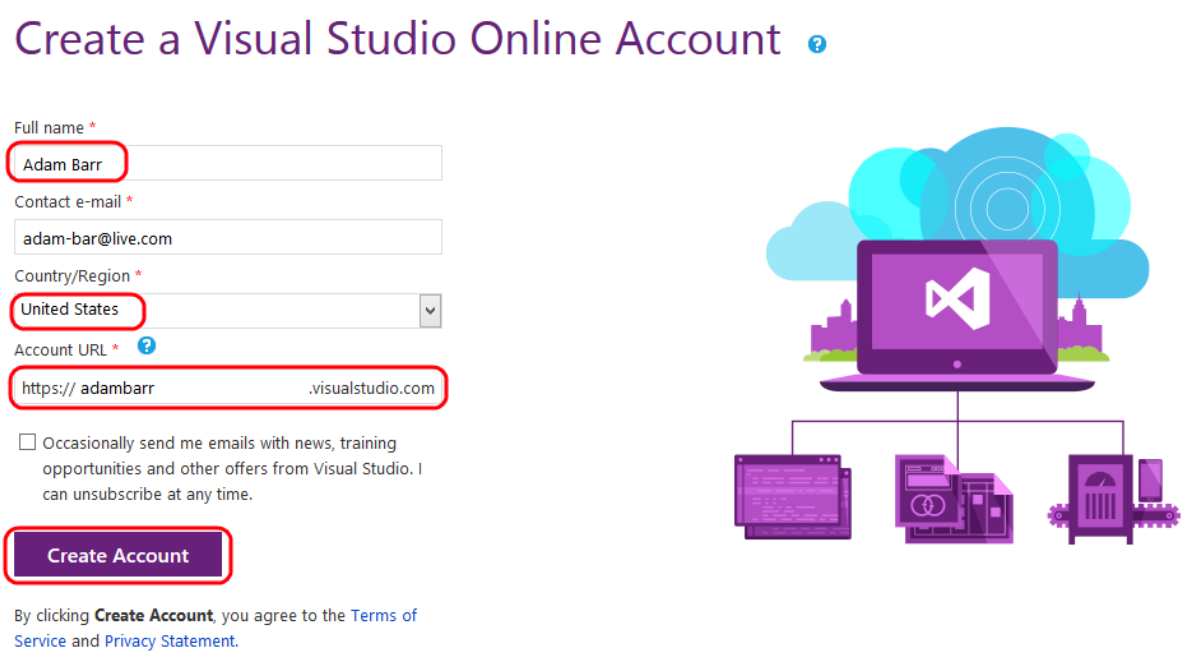
In this task, you'll create a free Visual Studio Online account.

Note: If you are running within a virtual environment, it is recommended that you use Internet Explorer within that virtual environment to make it easier to open your new Visual Studio Online account in Visual Studio 2013.

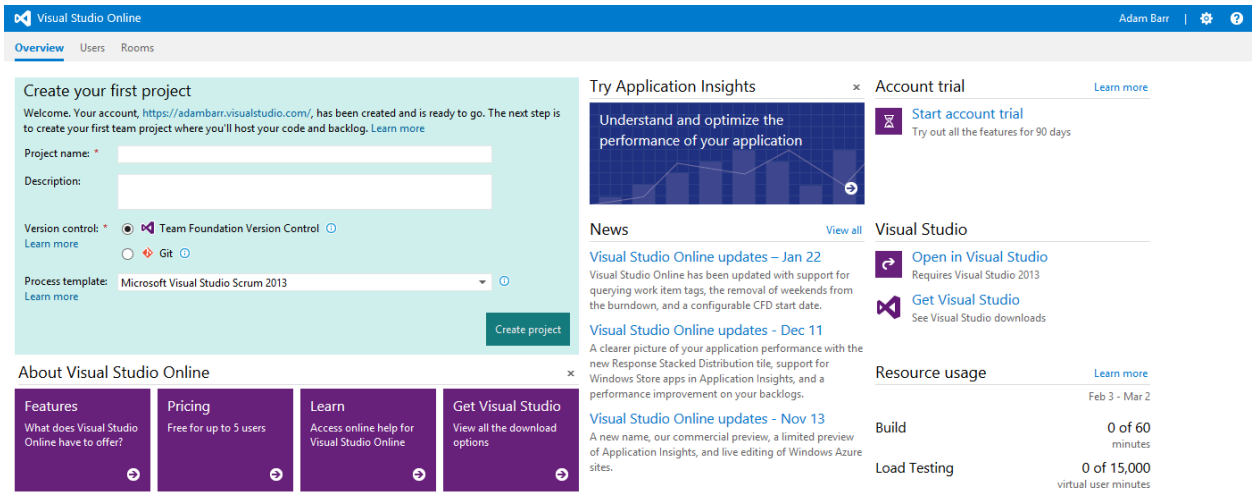
1. Navigate to http://www.visualstudio.com/products/visual-studio-online-overview-vs and select the "**Get started for free**" link.



2. Log in with your Microsoft account. If you do not have one, you need to create one before moving on.
3. Provide the requested information. This varies depending on how much information could be retrieved from your Microsoft account. If requested, enter **Full Name**, **Country/Region**, and a name for Visual Studio Online account (https://ACCOUNT.visualstudio.com). Click the **Create Account** button. If your screen only shows the account URL, simply fill in that with what URL you will want your team to use to access the team environment.



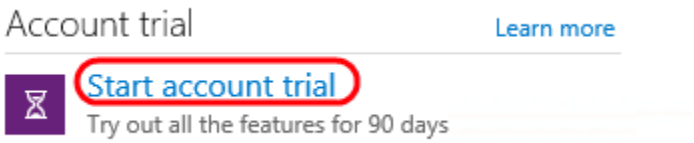
4. You should now be logged into your Visual Studio Online account.



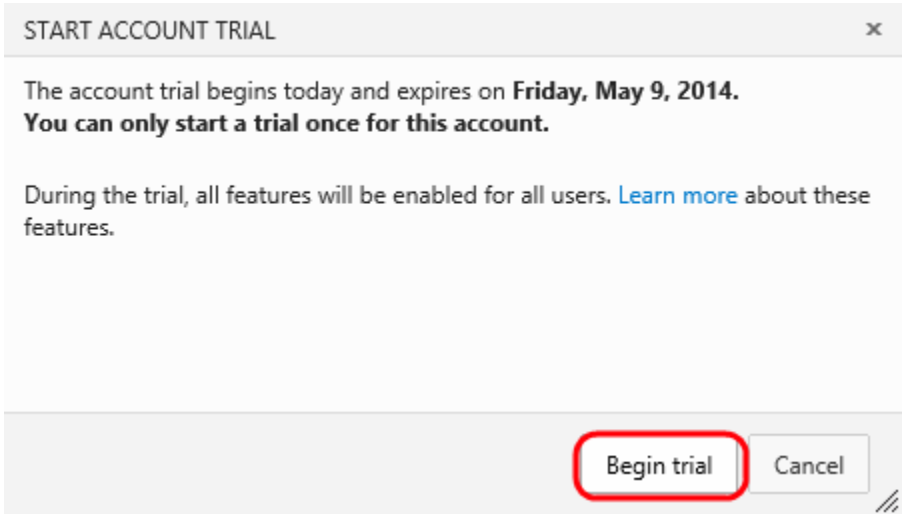
Task 2: Starting Your Free Trial

In this task, you'll start the free trial for your VSO account. This enables a number of features that you will explore later on in the lab. If you plan to use an existing VSO account and have already used up the trial period, you are advised to start this exercise over by creating a new VSO account for the purposes of this lab.

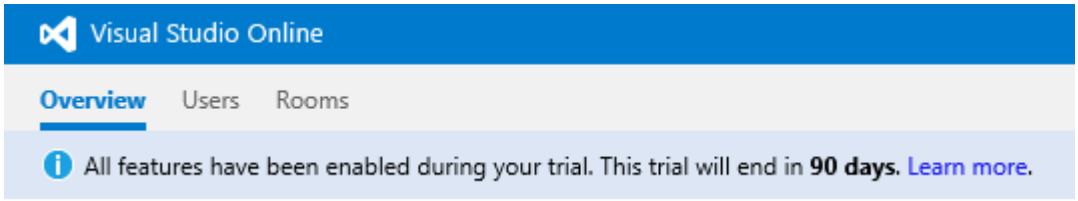
1. Select the "**Start Account Trial**" link. This link could be located to the right of the screen as shown above or below the first box depending on your screen size.



2. Click the **Begin Trial** button to enable all Visual Studio Online features for 90 days.



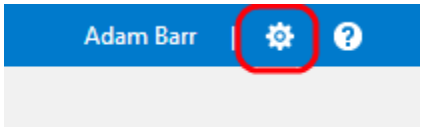
3. The features that are enabled by the trial include:
    - Team Room collaboration
    - Portfolio backlogs
    - Requesting user feedback
    - Work item charting
    - Web test case management



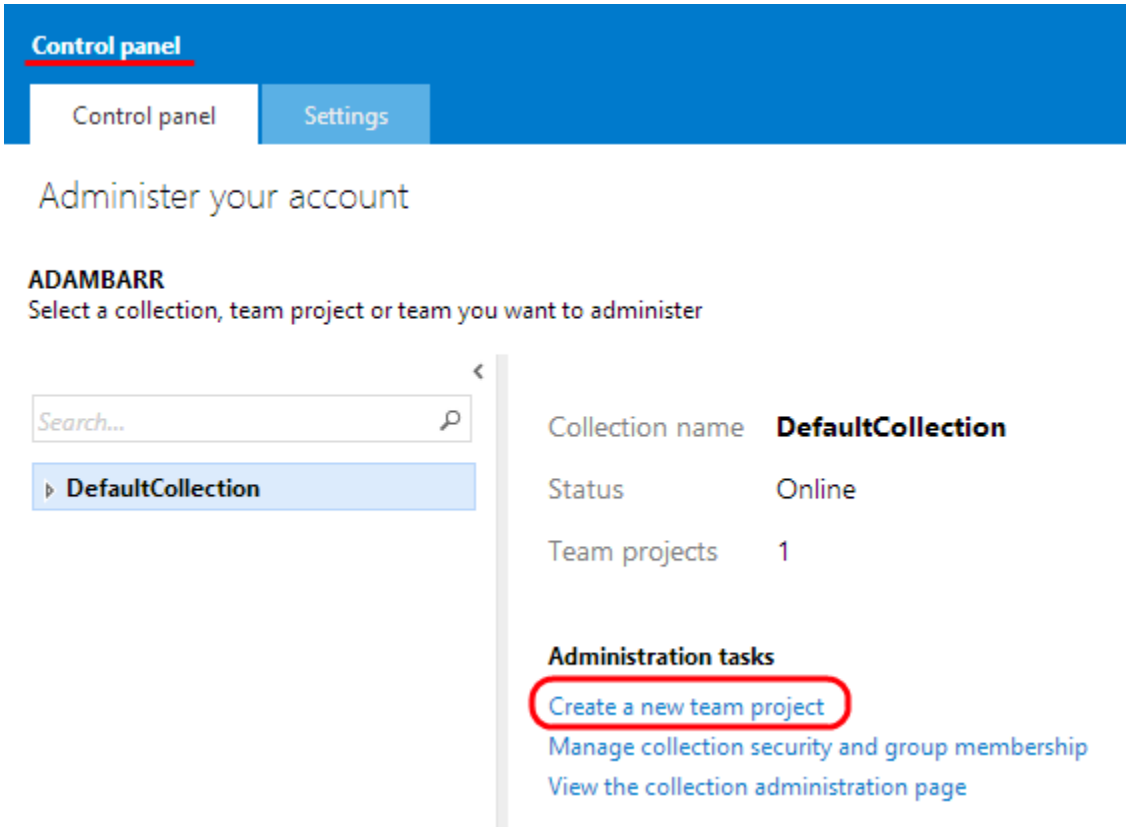
Task 3: Creating a Team Project

In this task, you'll create a new team project from within the Visual Studio Online (VSO) portal.

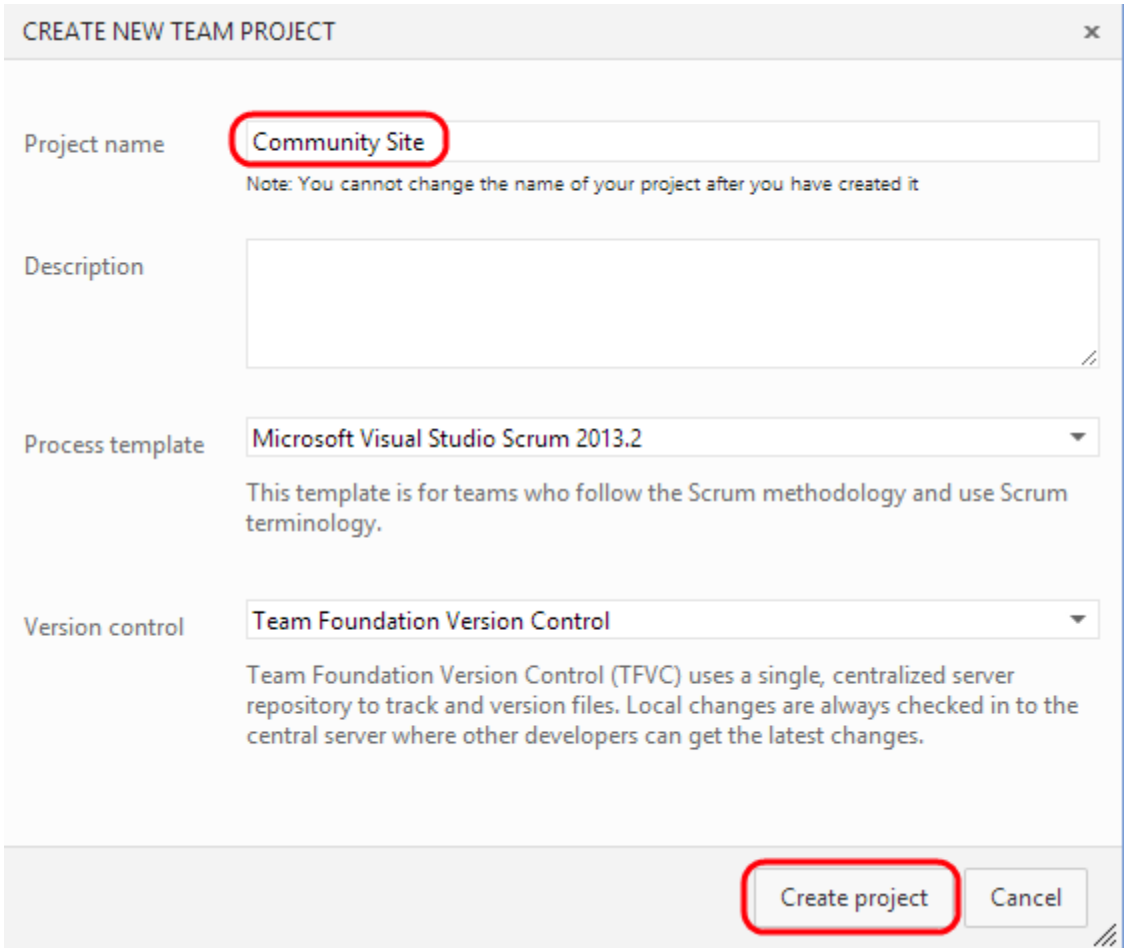
1. Select the **gear** icon in the top right corner of the page to load the administrative control panel.



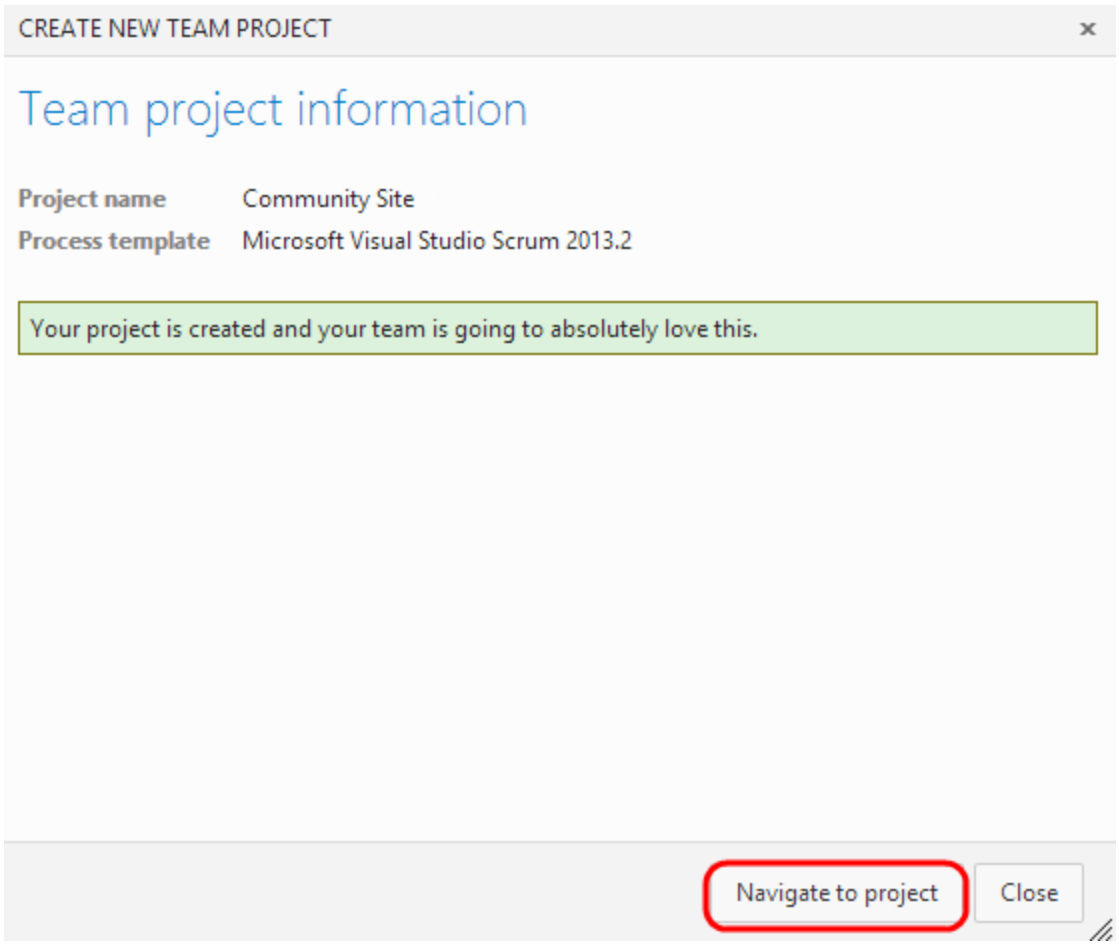
2. Ensure that you are viewing the root of the control panel (see underlined link in screenshot) and have selected the **DefaultCollection** node. Select the "**Create a new team project**" link.



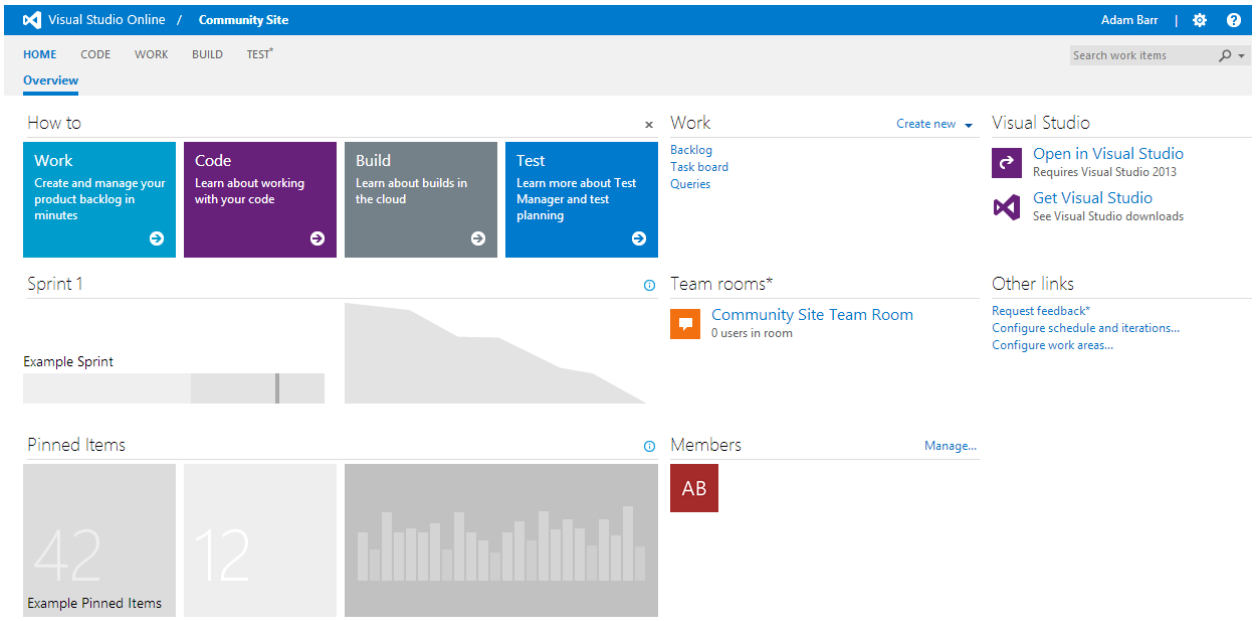
3. For the **Project Name** field, enter "**Community Site**". Click the **Create Project** button to create the new project using the defaults of Team Foundation Version Control and Microsoft Visual Studio Scrum 2013. The focus of this lab is to introduce the team development features available with Visual Studio Online rather than the chosen methodology. The Scrum (http://en.wikipedia.org/wiki/Scrum_(development)) methodology is just a specific implementation of an agile development process.



4. Once the new team project is created, click the **Navigate to Project** button.



5. After navigating to the project, you should now see the default project landing page.



Exercise 2: Administration of Projects, Teams and Permissions

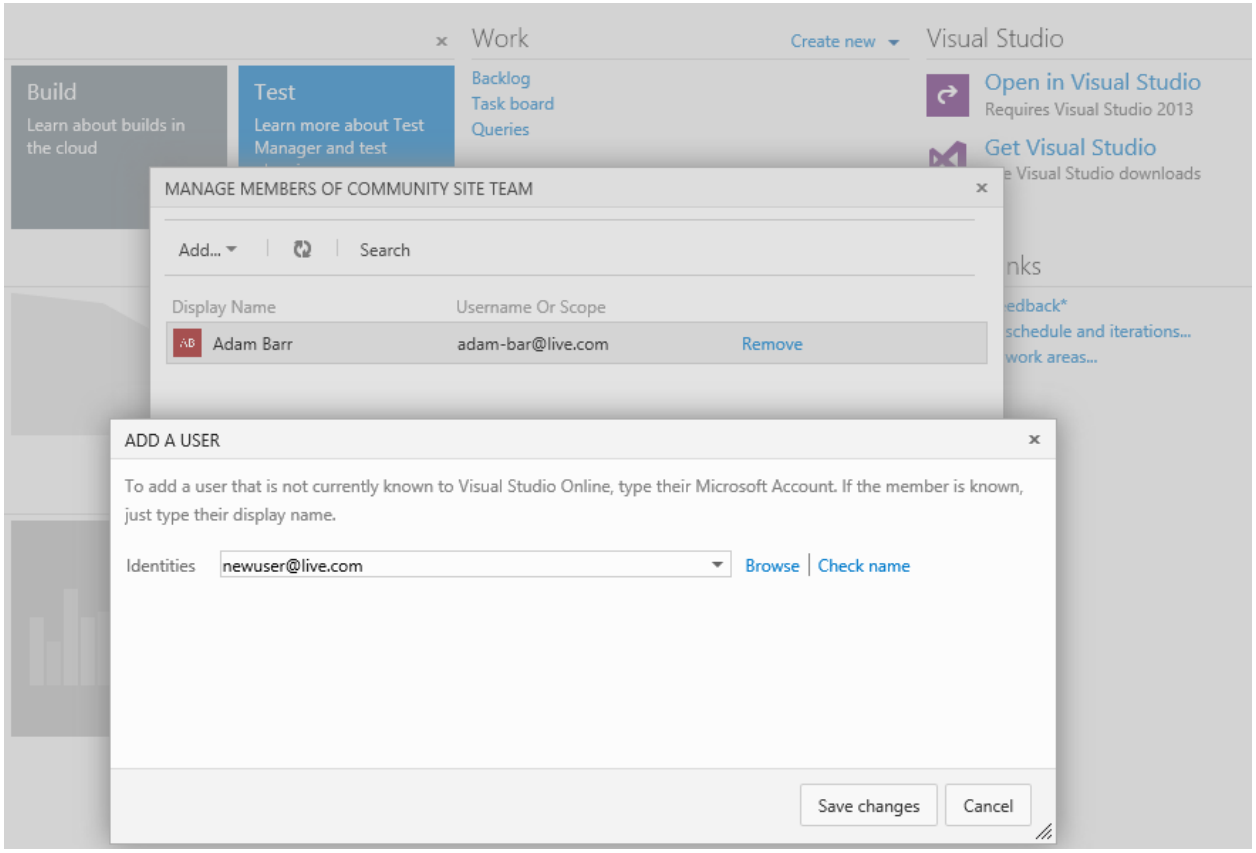
In this exercise, you'll take a quick survey of project and team administration using Visual Studio Online.

Task 1: Administration of Team Members (Quick Start)

In this task, you will learn just how quickly you can get up and running with your new team project by adding additional members.

1. Navigate to the project overview page.
2. Select the **Manage…** link from the Members section.

3. The **Manage Members** dialog window allows you to quickly add additional team members directly, or by including a Visual Studio Online group. For users that have not previously connected to VSO, you can simply provide their email address and they will be added and receive a welcome message. New users will also need to create a Microsoft account if they do not currently have one. For the purposes of this lab, there is no need to need to add additional team members here, so **close** the dialog windows after you are done viewing them.

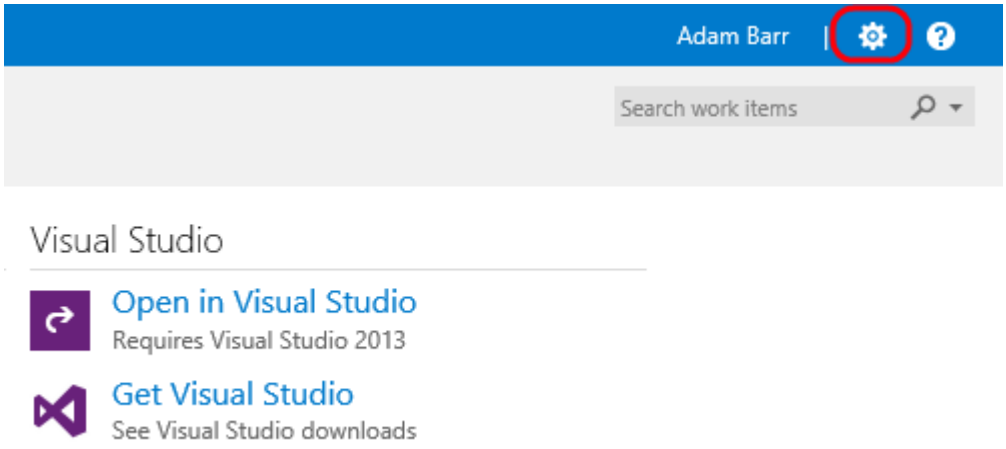


4. After adding some team members to your project, you should be able to start development using your Visual Studio Online account with the defaults that are in place, but before you get to that, you'll survey some additional configuration and customization points available from within the VSO control panel.

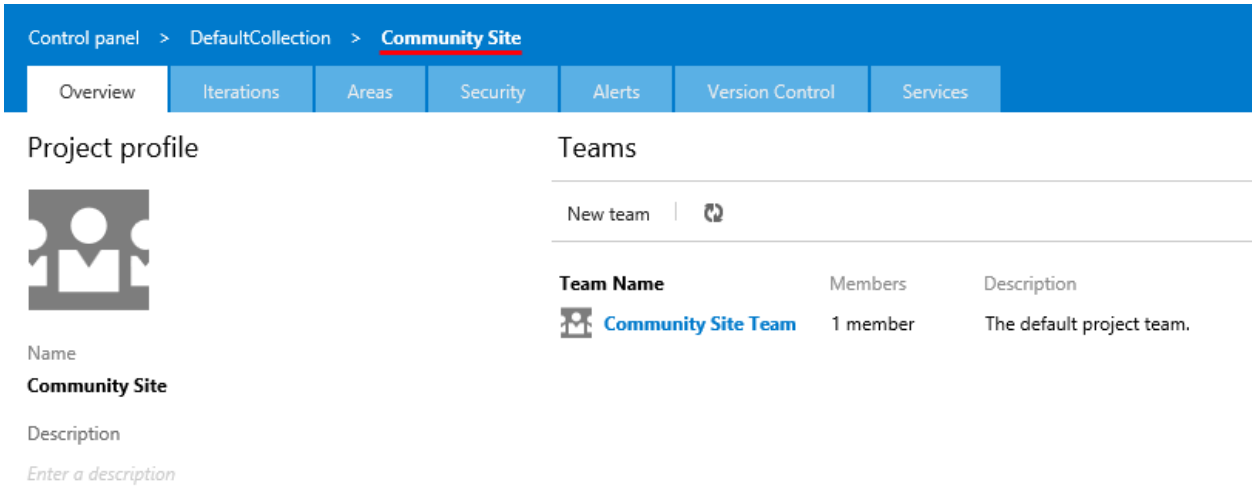
Task 2: Administration of Projects, Teams, and Permissions (Administrative Control Panel)

In this task, you'll take a quick look some of the other project and team configuration/customization points available in the control panel.

1. Select the **gear** icon in the top-right of Visual Studio Online to navigate to the administrative control panel. The control panel should open in a new tab.



2. There are a number of different scoping levels in the control panel that allow management of everything from the VSO account to the teams. You should now be looking at the control panel that is scoped to the new "**Community Site**" project.

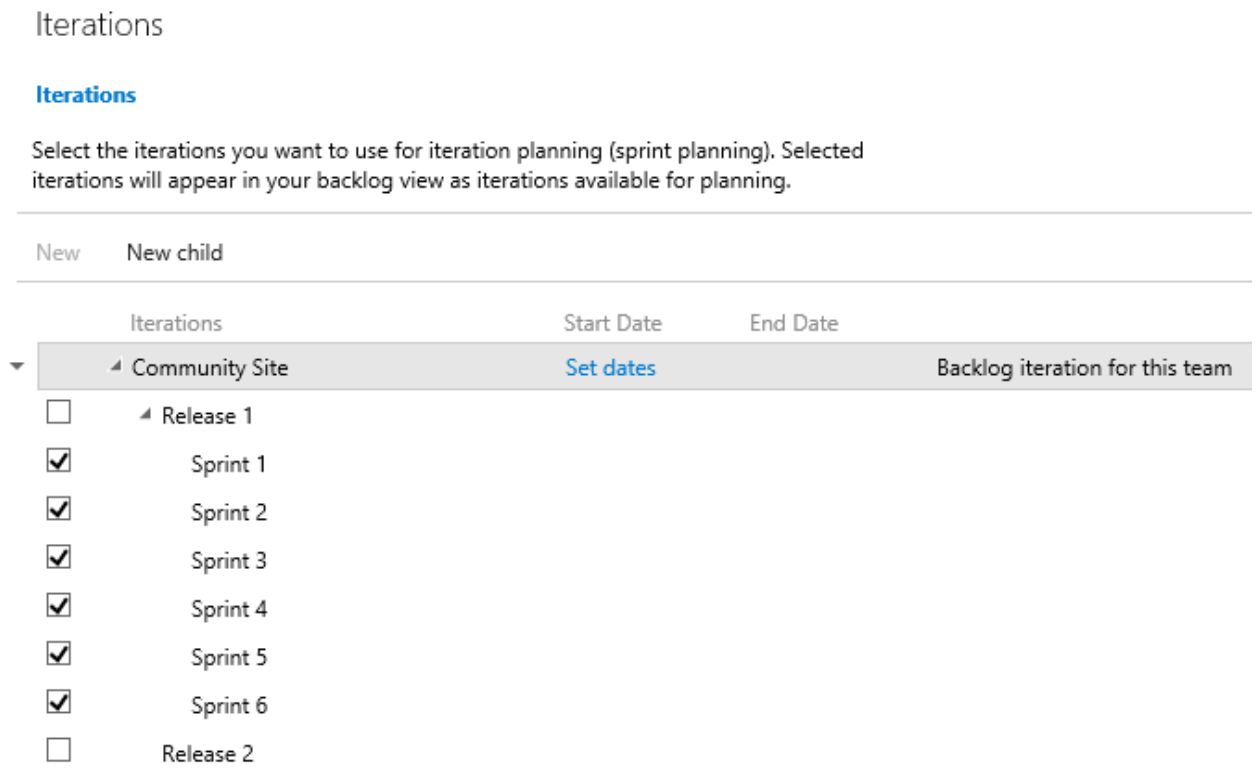


Note: You can add additional teams to a project. This would be useful in a scenario where you had multiple teams contributing to a project and wanted to keep their backlogs of work separated, or

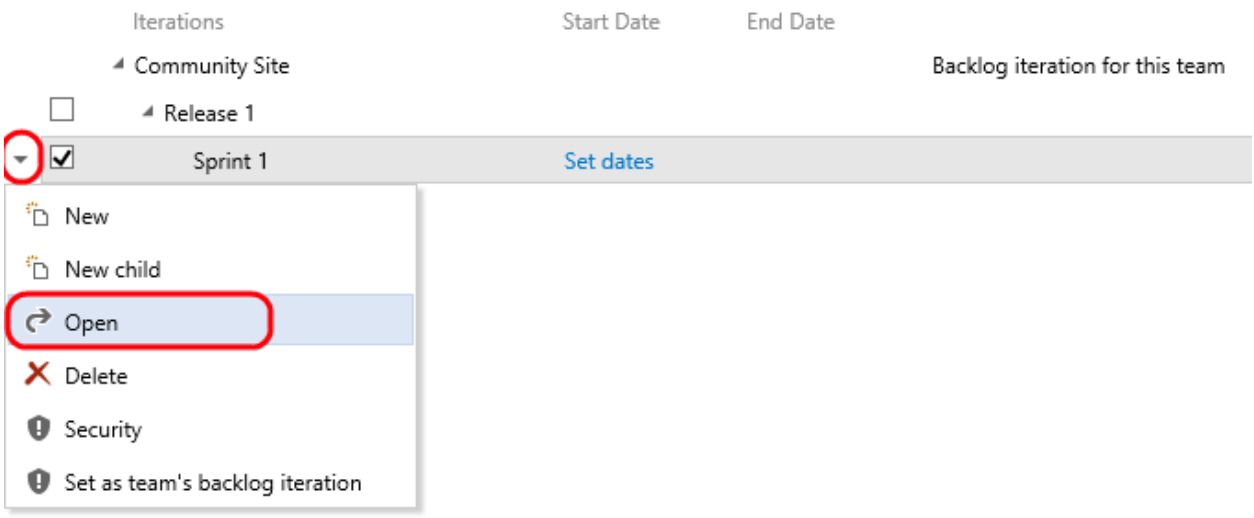
perhaps assign different permissions. For the purposes of this lab, however, we will stick with using this default team. To learn more about adding additional teams to a project, please take a look at the article on MSDN: **http://msdn.microsoft.com/en-us/library/hh500414.aspx**.

3. Select the **Iterations** tab.

4. As stated on the web page, these iterations are used during sprint planning for the project and selected iterations will show up on the backlog view. The Scrum methodology is an iterative process that is typically broken up into "smaller" chunks of time. By default, a hierarchy is created that defines a few releases, with a few sprints already defined for the first release.

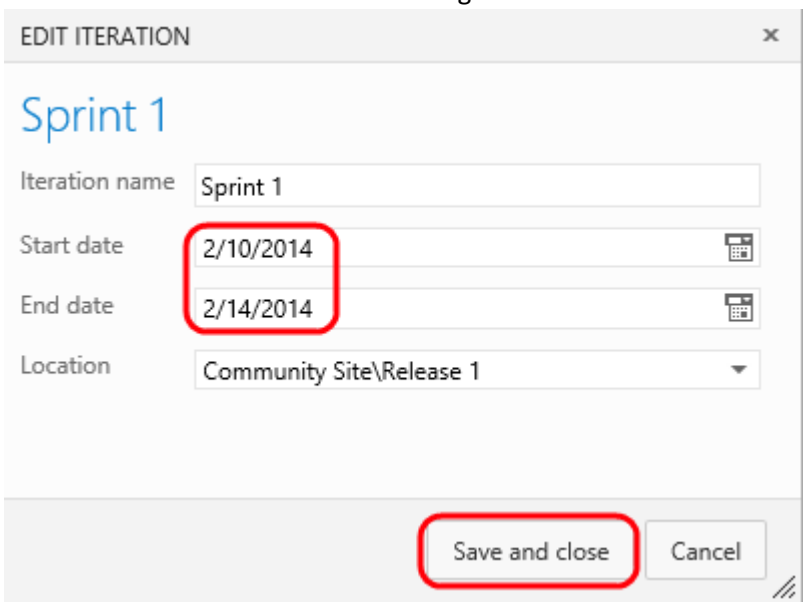


5. Select the down arrow to the left of the "**Sprint 1**" iteration to expose the context menu and then select the **Open** option.



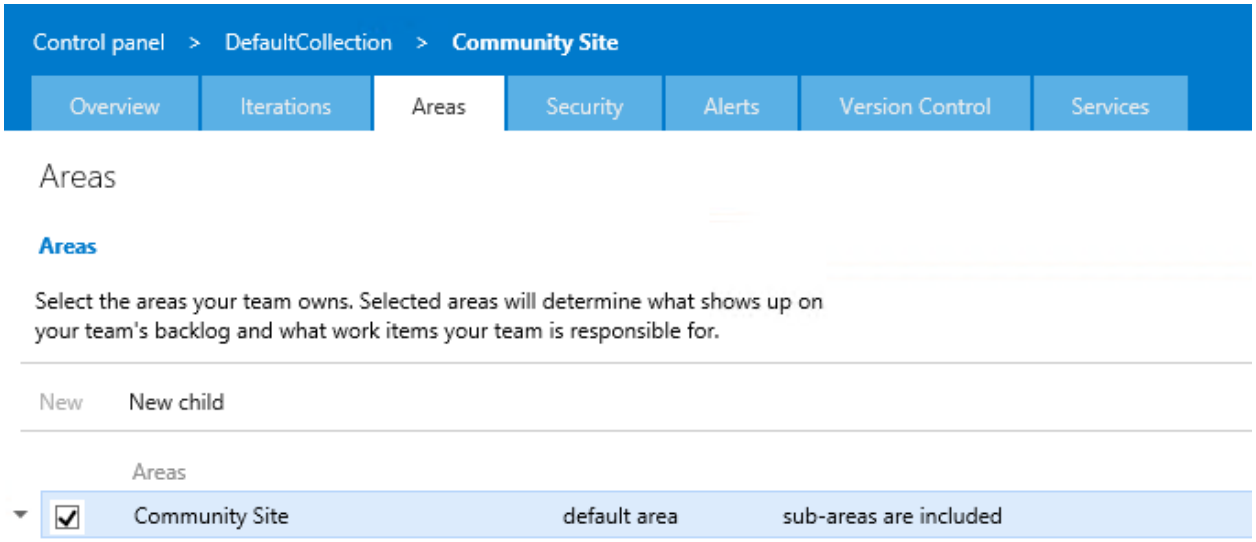
6. Imagine that the team has decided to use 1 week sprints. Although we may typically start a sprint at the beginning of a week, go ahead and assign a start date of today, and an end date four business days later (which are Monday through Friday). Click the **Save and Close** button.

   Note: The dates shown in the following screenshot will be different from yours.



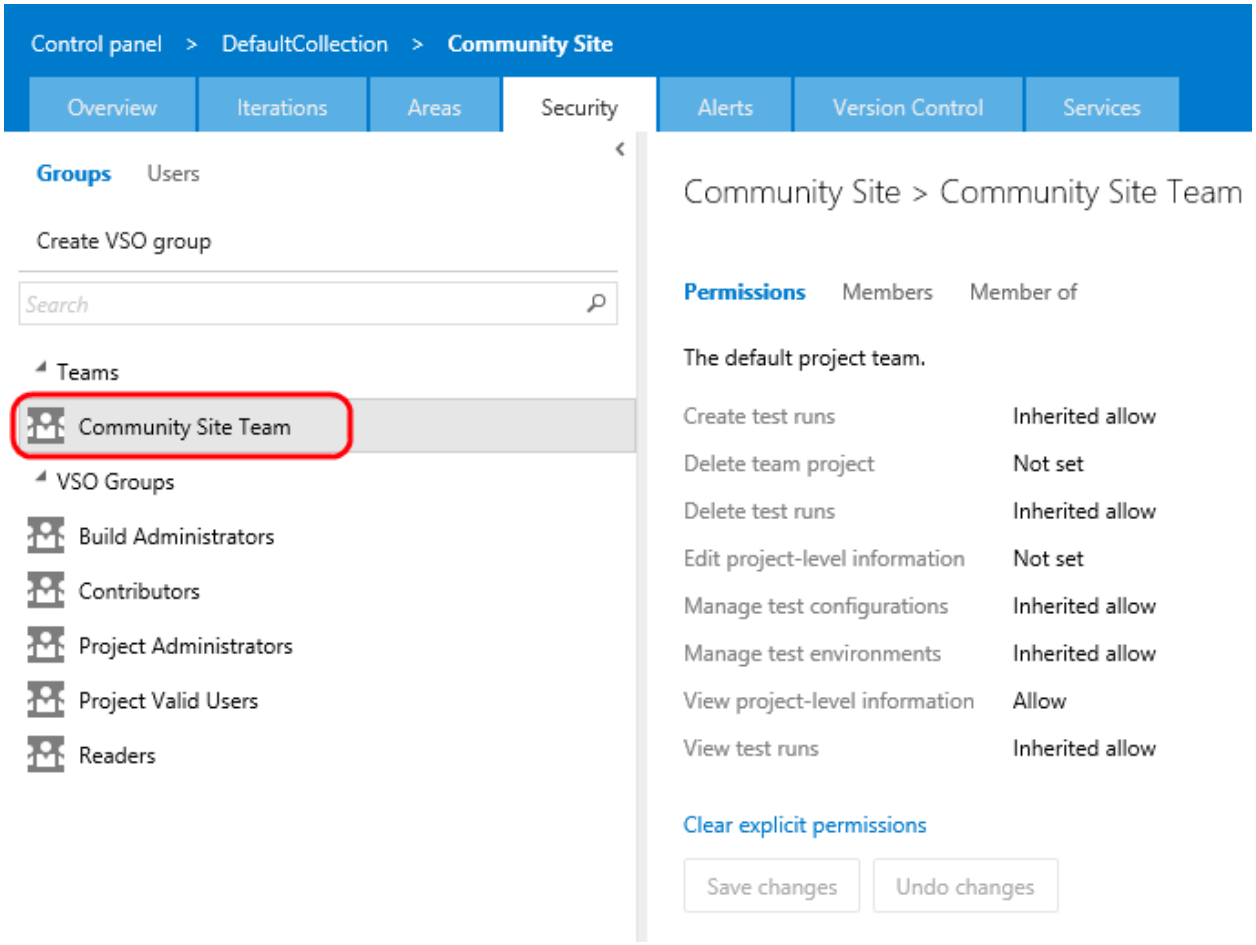
7. Select the **Areas** tab.

8. Area paths and their configuration determines what work will show up on a team backlog. In this simple case, we only have one team and one area defined. If you were to create another team, this would create additional child areas underneath the root “Community Site” area.

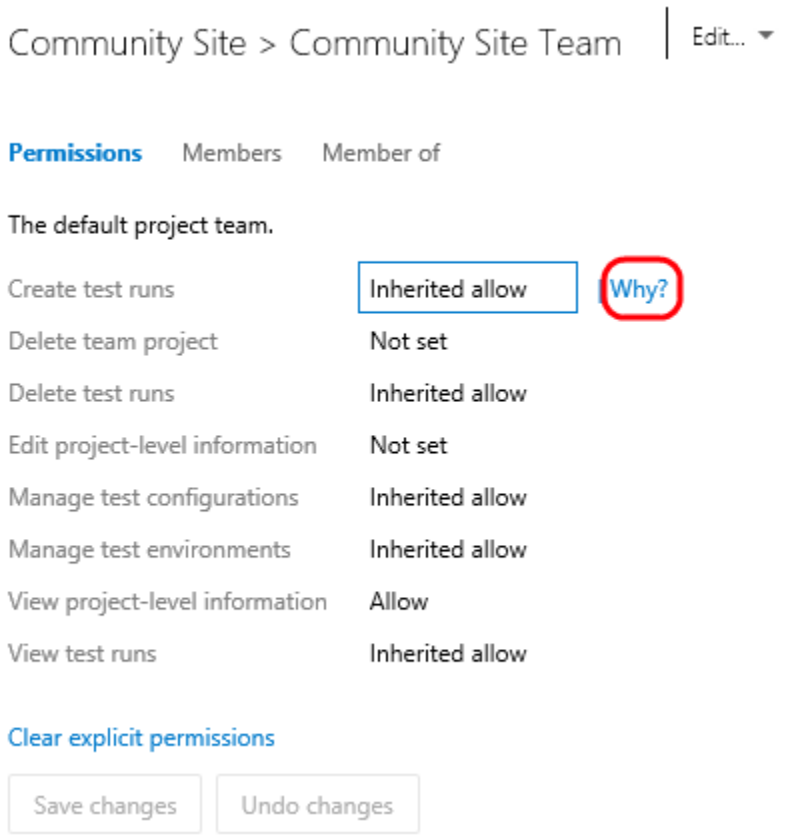


9. Select the **Security** tab.

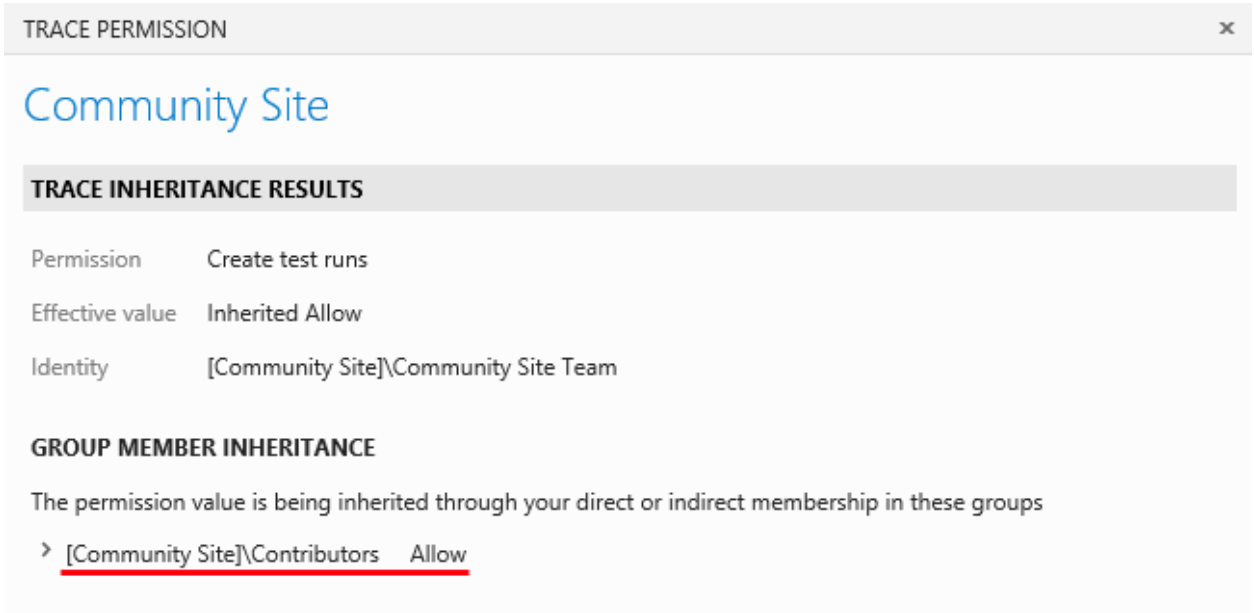
10. You should now be looking at the default security settings for the "**Community Site Team**" group, which is the default team for the project. Note that members of the team have the permission to create/delete test runs and view project-level information, but not to delete the team project.



11. This interface makes it easy to understand effective permissions for groups and users. Each permission shows that it is set to some variation of "**allowed**" or is listed as "**not set**". Hold the mouse cursor over the "**Create Test Runs**" permission and then click on the "**Why?**" link that appears on the right side.

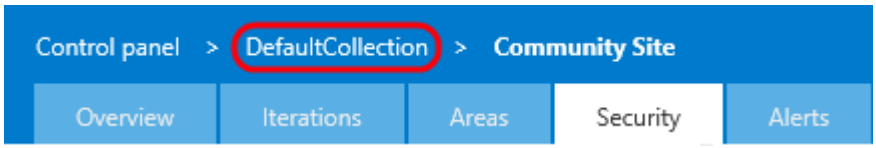


12. Tracing the permission inheritance shows us that **Community Site Team** members are allowed to create test runs because the security object representing the team is a member of the **Contributors** group, which explicitly allows this permission.



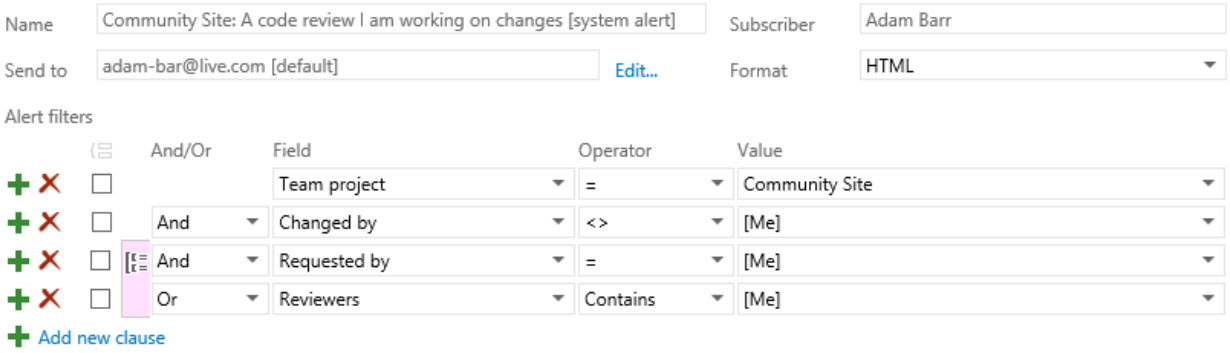
13. **Close** the **Trace Permission** window. In most simple scenarios, these default permissions will suit you just fine without further modification.

    Note: Although we will not do this as part of this lab, you could use this same interface to modify project collection security by clicking on the DefaultCollection navigation link near the top.

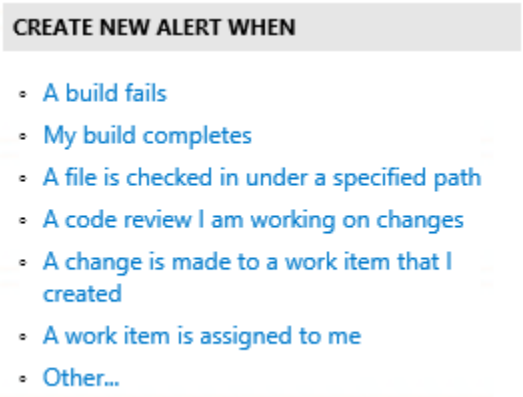


14. Select the **Alerts** tab. Team members can create and configure alerts to send out notifications for check-ins, code reviews, work items, and builds.

15. These alerts are highly customizable, as you can see by the definition of the default code review alert that was created for the logged in user. You can add in an arbitrary number of clauses and make use of various logical operators.

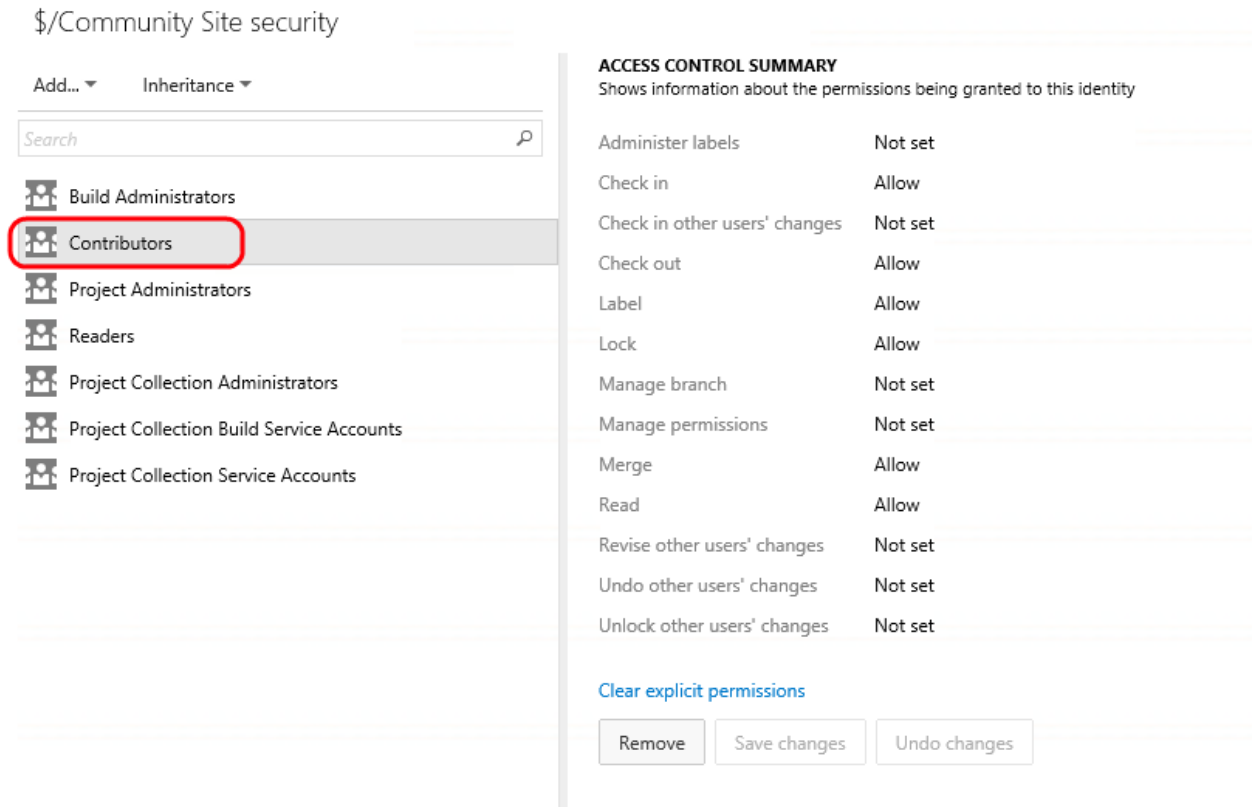


16. There are a number of common alert templates that you can use to get started listed in the "**Create New Alert When**" section. For example, you could setup a team alert (that goes out to members' default alert address) for the failure of a specific build.



17. Select the **Version Control** tab. This is where you can view and modify the security of the source control component of the project.

18. On the left side you can see the source control tree, which allows you to select the root source control node, or something more granular if you so choose. The middle column allows you to select groups or users, and then view/modify the source control permissions on the right. Select the **Contributors** group to view the default permissions.



19. **Close** the browser **tab** to return the project overview page.

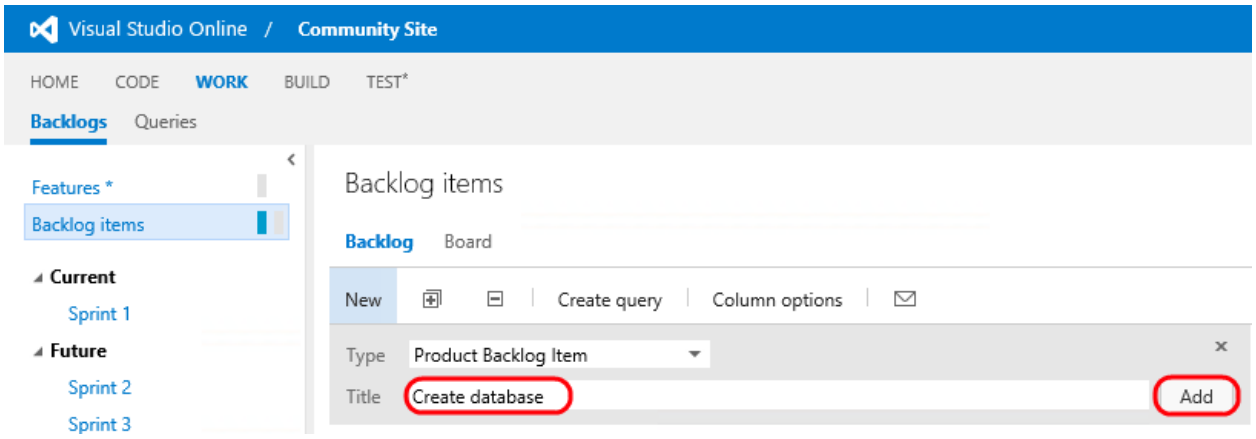
Exercise 3: Planning and Execution

In this exercise, you'll use the Visual Studio Online portal to walk through the creation of a project backlog, estimation of team resource capacity, estimation of effort, and the planning of iterations. Once the project is underway, you'll learn how to update and view project status using the available tools.

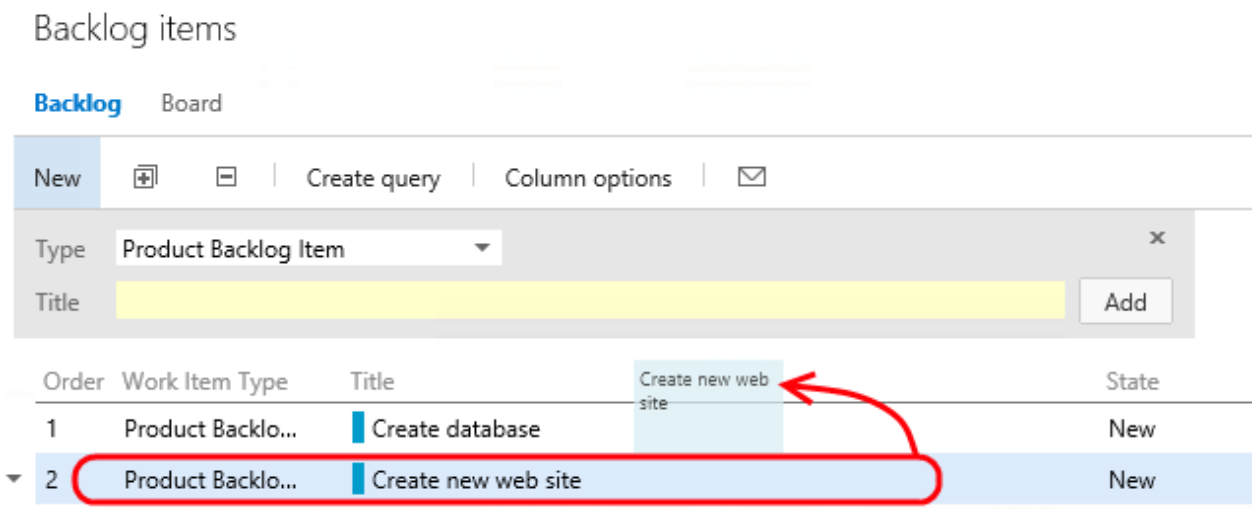
Task 1: Creating a Product Backlog

1. Select the **Backlog** link from the **Work** section of the project overview page.

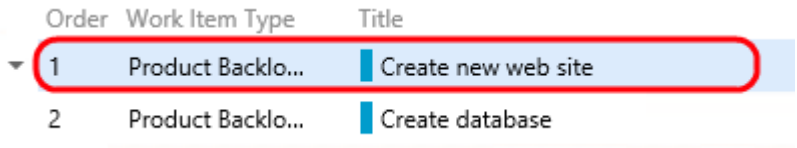
2. The process template that was selected when creating the project determines the types of work items that can be created on the backlog. Create a new Product Backlog Item (PBI) titled "**Create database**" and then click the **Add** button.



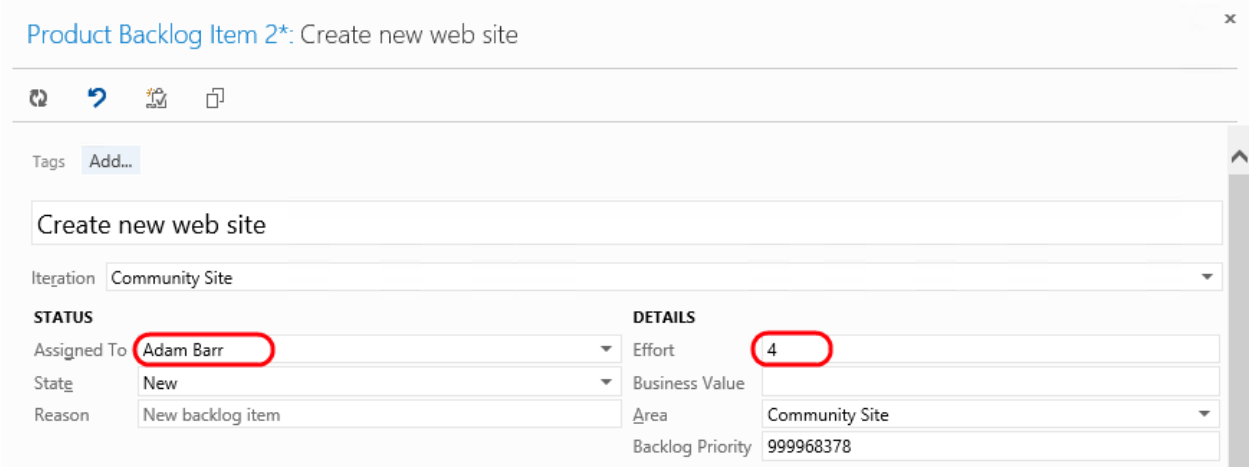
3. Create a second PBI titled "**Create new web site**".
4. You can order the backlog items by dragging and dropping them. This can help assign an initial priority that the team can further discuss during future sprint planning. Try dragging the "**Create new web site**" work item and dropping it just above the "Create database" work item.



5. Double-click on the "**Create new web site**" PBI to open it.

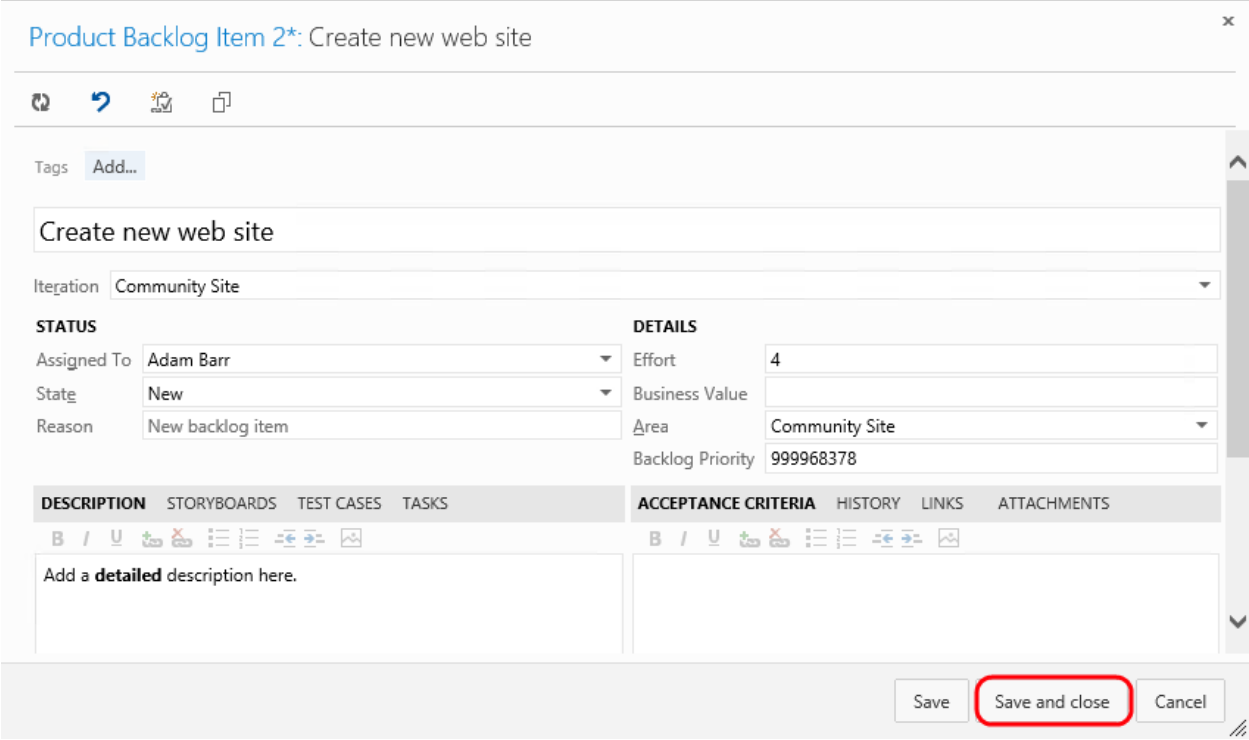


6. **Assign** the PBI to yourself (user that you are logged in with) and assign an **Effort** of '**4**'. The effort here is a relative measure (to other PBIs) that indicates how much work it will take to complete. This field is synonymous with what are called Story Points in most agile methodologies.

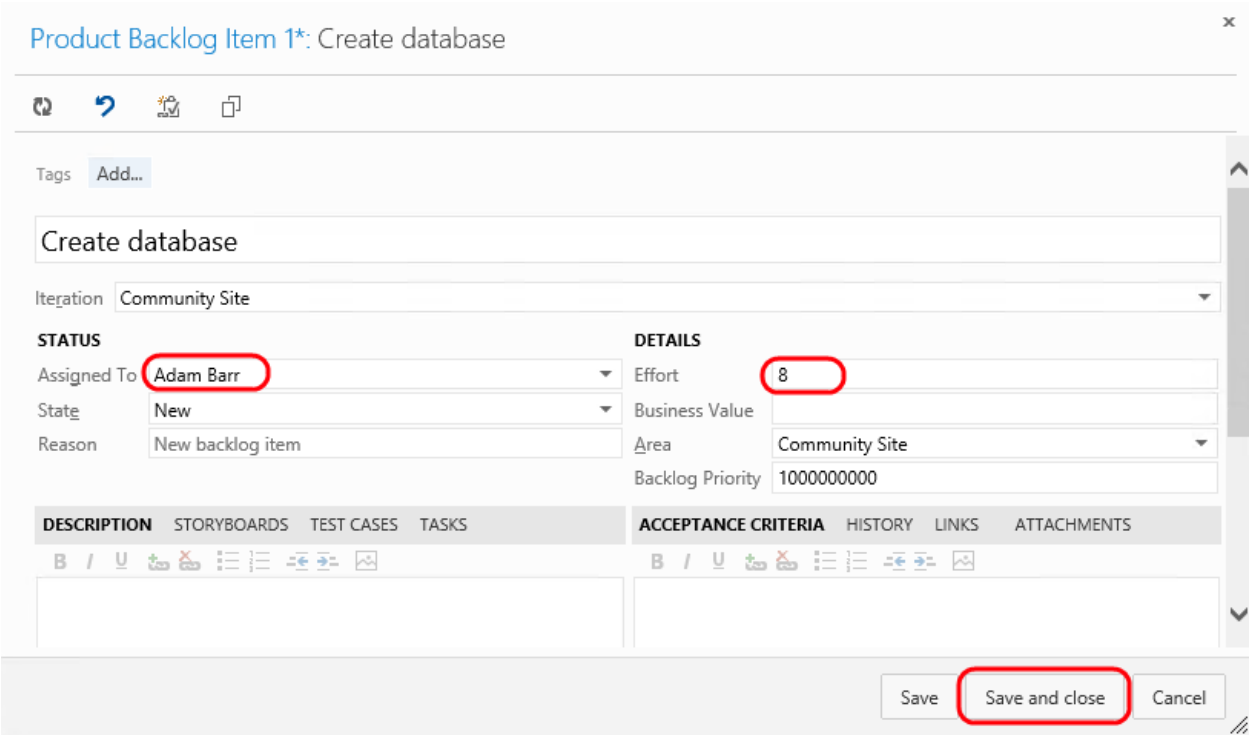


Note: The Backlog Priority number is a relative measure between work items, with smaller numbers representing higher priority. When you drag and drop items in the backlog, this number will automatically be adjusted for you.

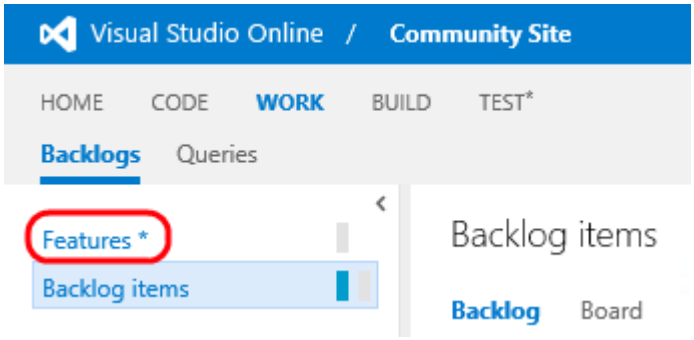
7. Take note of the additional metadata and links to additional data that can be associated with work items. It is possible to add in a description using a rich text editor, link to a storyboard, create or link to test cases, link to other miscellaneous artifacts, and define acceptance criteria. Go ahead and click the **Save and Close** button to continue.



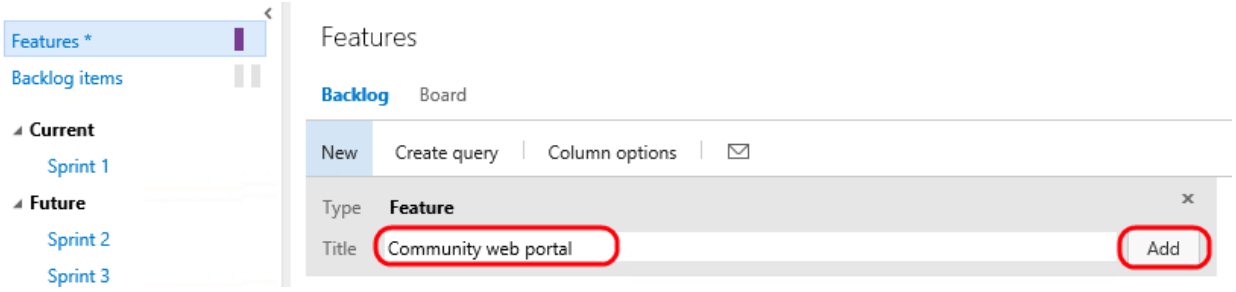
8. Double-click on the "**Create database**" PBI to open it.
9. **Assign** the PBI to yourself and assign an **Effort** of '**8**'. With this effort value you are stating that you think it will take twice the amount of effort to complete. Select the **Save and Close** button.



10. Although it is beyond the scope of this lab to go into much detail, you can also create feature backlogs in Visual Studio Online. These feature backlogs contain a higher level view of work in a project and can be used to provide a management team insight into the work across several agile development teams. Select the **Features** link to navigate to this backlog.

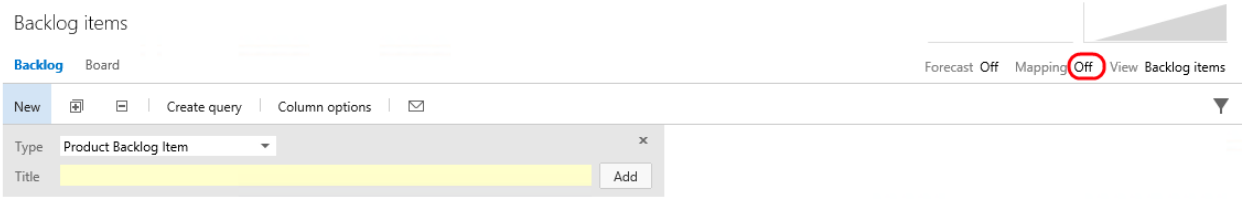


11. Create a new feature titled "**Community web portal**" and then click the **Add** button.

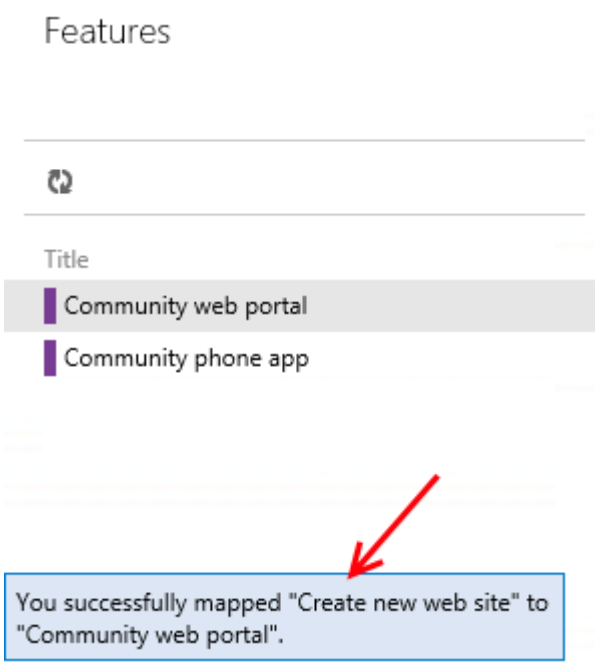


12. Create a new feature titled "**Community phone app**" and then click the **Add** button.
13. Select the "**Backlog Items**" link to return to the list of PBIs.

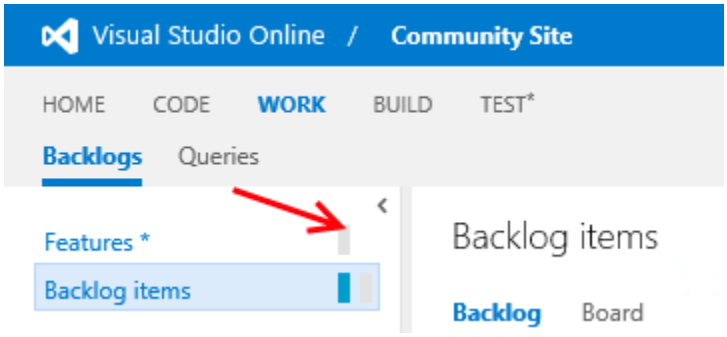
    Note: If you are interested in learning more about the feature backlog and supporting scenarios with multiple agile teams, please see the MSDN article that discusses Agile Portfolio Management here: **http://msdn.microsoft.com/en-us/library/dn306083.aspx**.
14. To finish the loop with feature management, you will now map the PBIs to the features they are associated with. Select the **Mapping** link to toggle the mapping panel on.



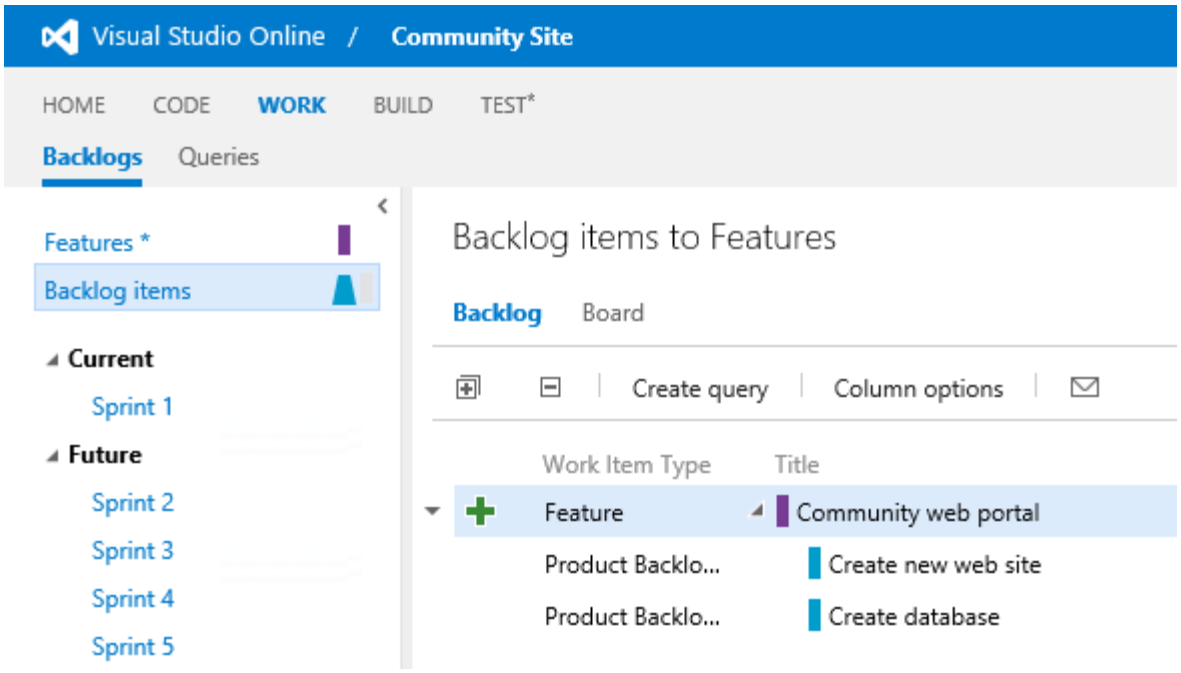
15. **Drag and drop** both of the PBIs to the "**Community web portal**" feature. After you drop each item, you will see a notification pop up for a few seconds that indicates success.



16. Select the **Mapping** link to toggle the mapping panel to off.
17. There is a hierarchical relationship that is made between work items that represent features, product backlog items, and tasks (which we will look at in more detail soon). From the Backlog Items view, we can either drill up to view associated features or we can drill down to see tasks that contribute to the PBIs. Select the **Features** bar (currently gray) to view all features that are contributed to by the current backlog items of the team.



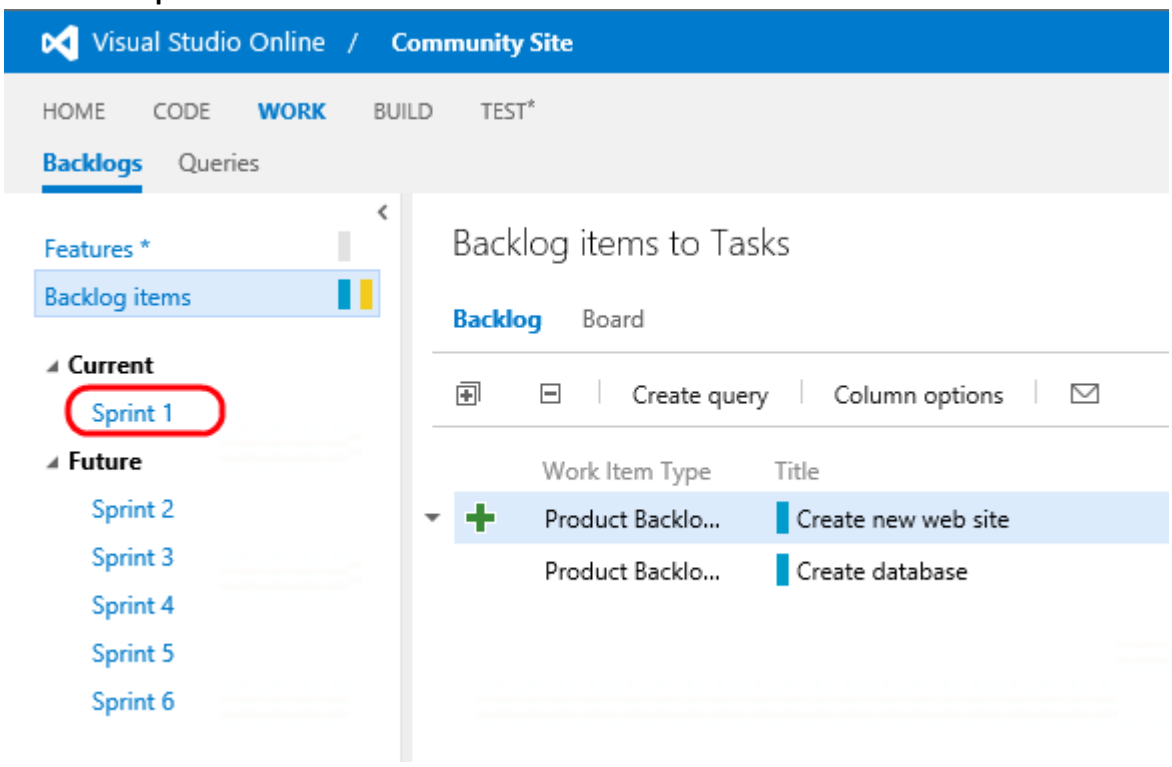
18. This shows that both PBIs are contributing to the "**Community web portal**" feature. Also note the green '**+**' button that allows you to create new PBIs which are automatically linked to the feature.



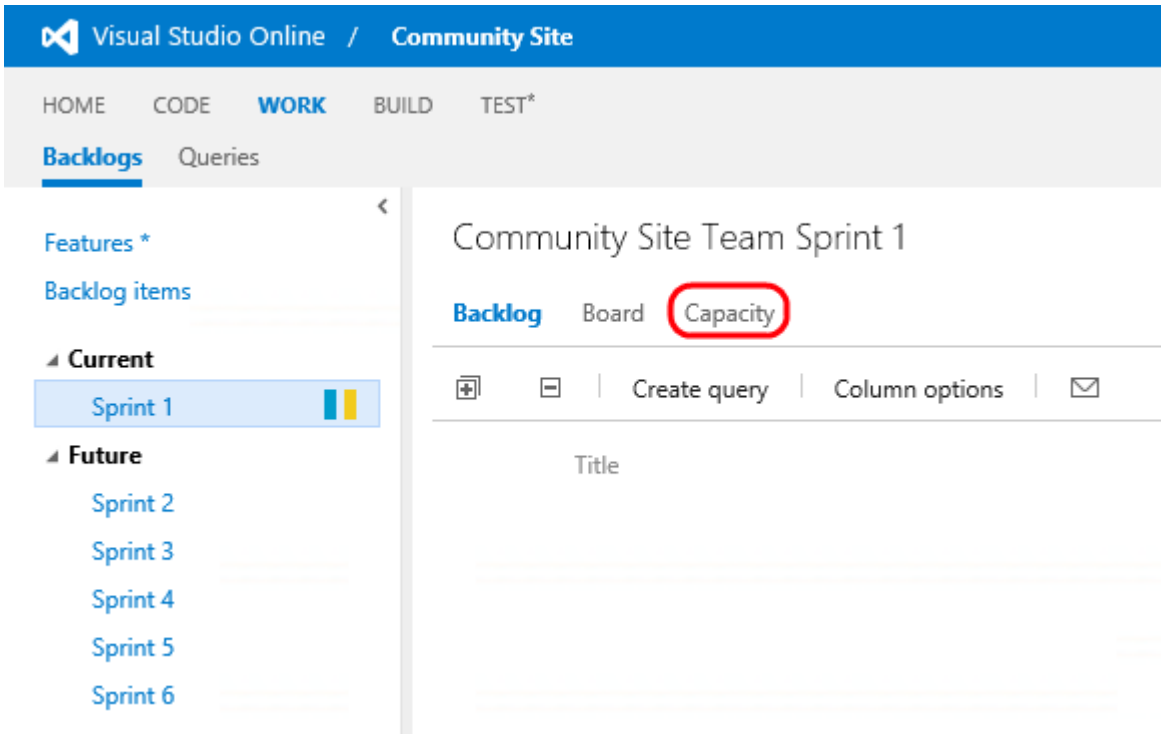
Task 2: Managing Team Capacity

Each sprint length and team member work capacity can be set independently, although sprint lengths are commonly held consistent during development. In this task, you will start the process of detailed iteration planning by assigning team capacity.

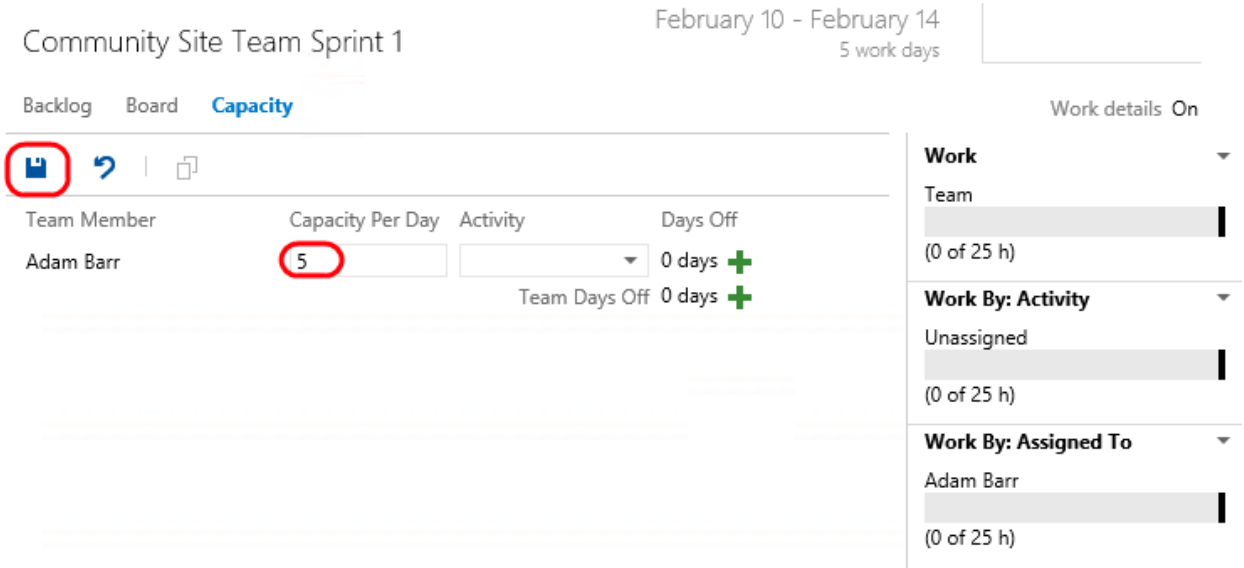
1. Select the "**Sprint 1**" link.



2. You should now be looking at the backlog for the current iteration. Select the **Capacity** tab.



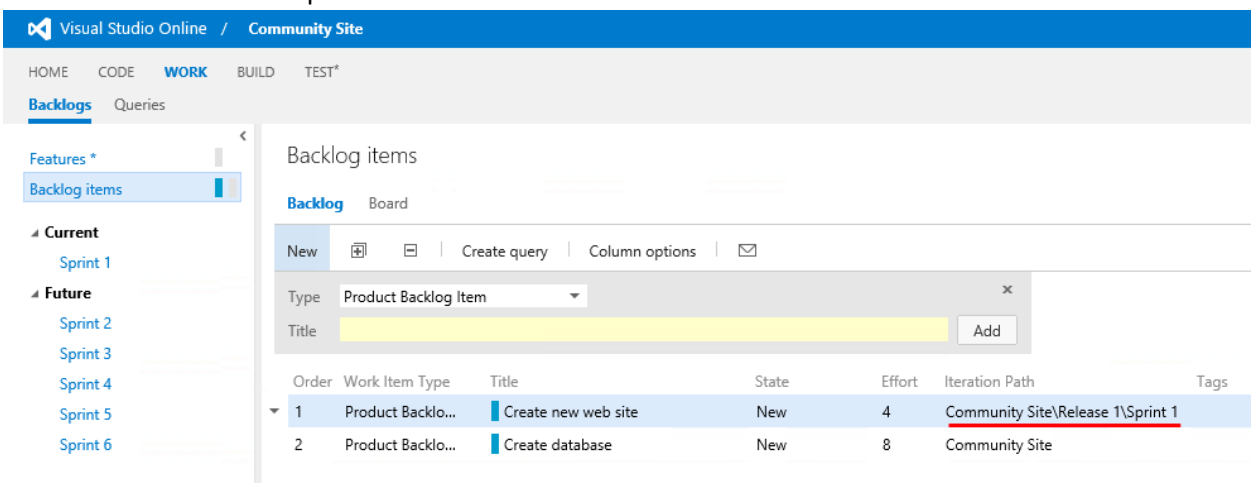
3. In this lab you only have one team member, but imagine that you have a few more. This capacity view allows you to define each member's capacity per day (in hours) during the sprint dates, as well as planned days off. Planned days off can also be defined for the entire team if necessary.
4. Assign a **Capacity Per Day** value of '**5**' and then click the **Save** button. Note that the work details pane shows the total work capacity available for the sprint (5 hours a day for a 5-day work week equals 25 hours).



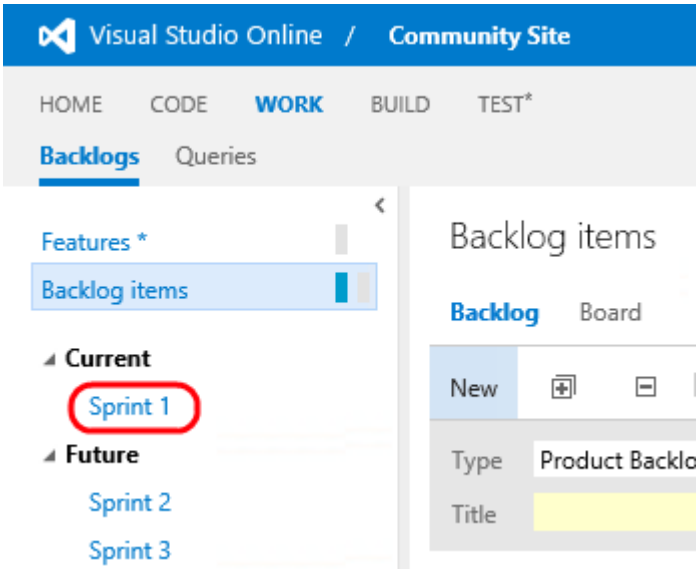
Task 3: Planning Iterations

With the iteration length, team member capacity, and backlog items in hand, the team can now tackle the task of assigning work to the iterations (sprints).

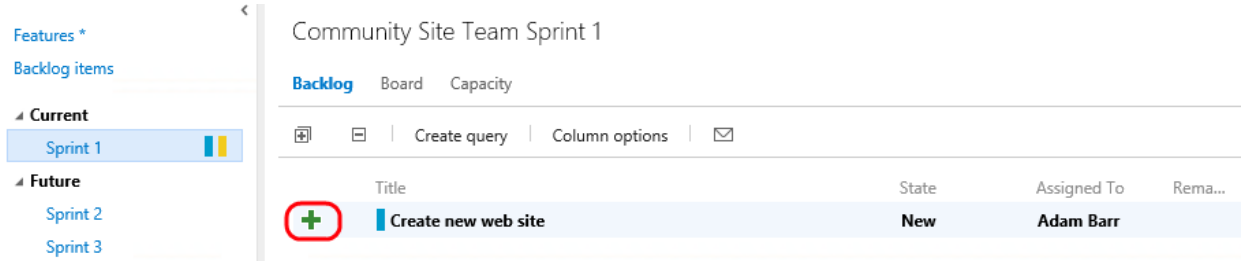
1. In the **Backlog Items** view, drag and drop the "**Create new web site**" PBI onto the "**Sprint 1**" iteration. Note that this updates the Iteration Path of the PBI.



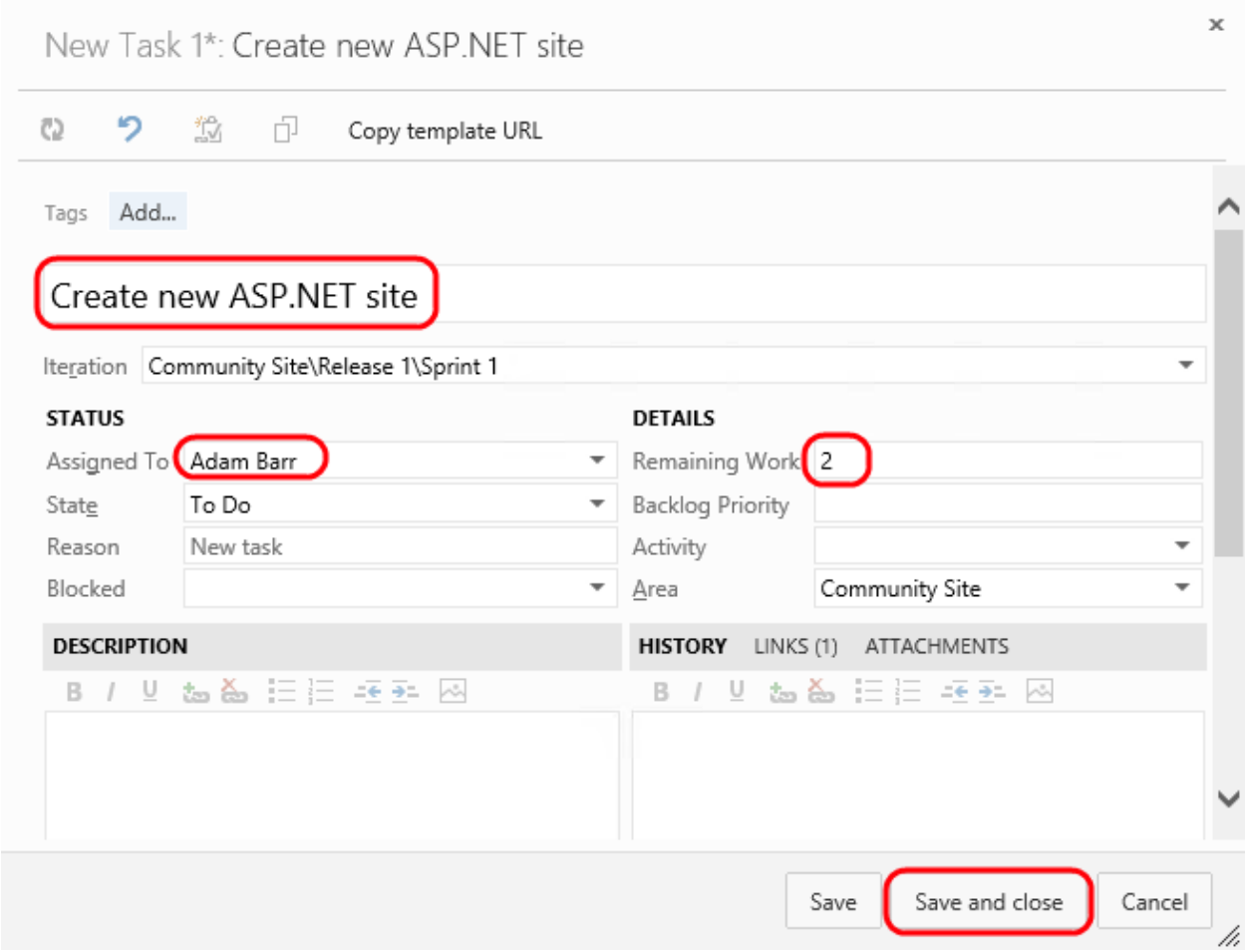
2. Select the "**Sprint 1**" link to open the iteration backlog.



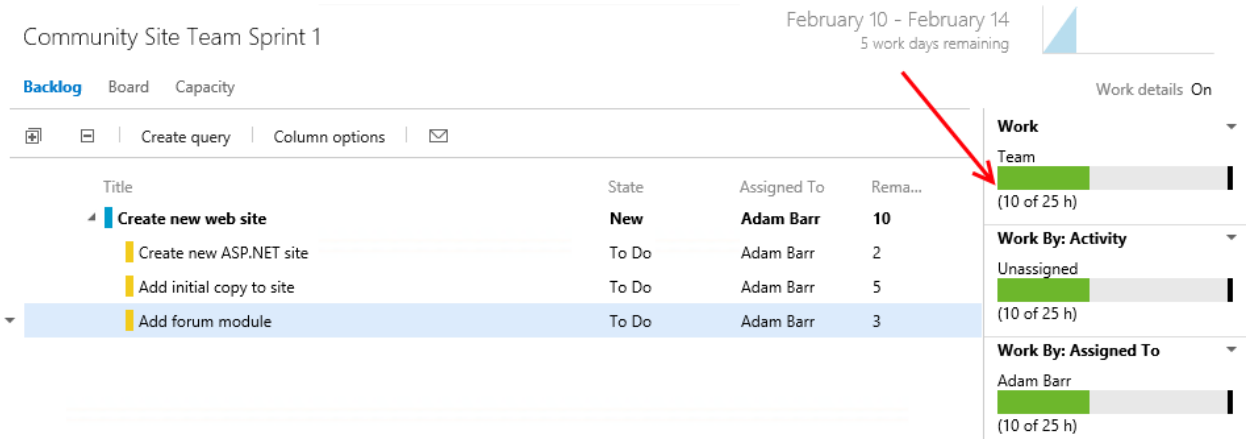
3. Hold the mouse cursor over the "**Create new web site**" backlog item and then click the green '**+**' button that appears to create a new task.



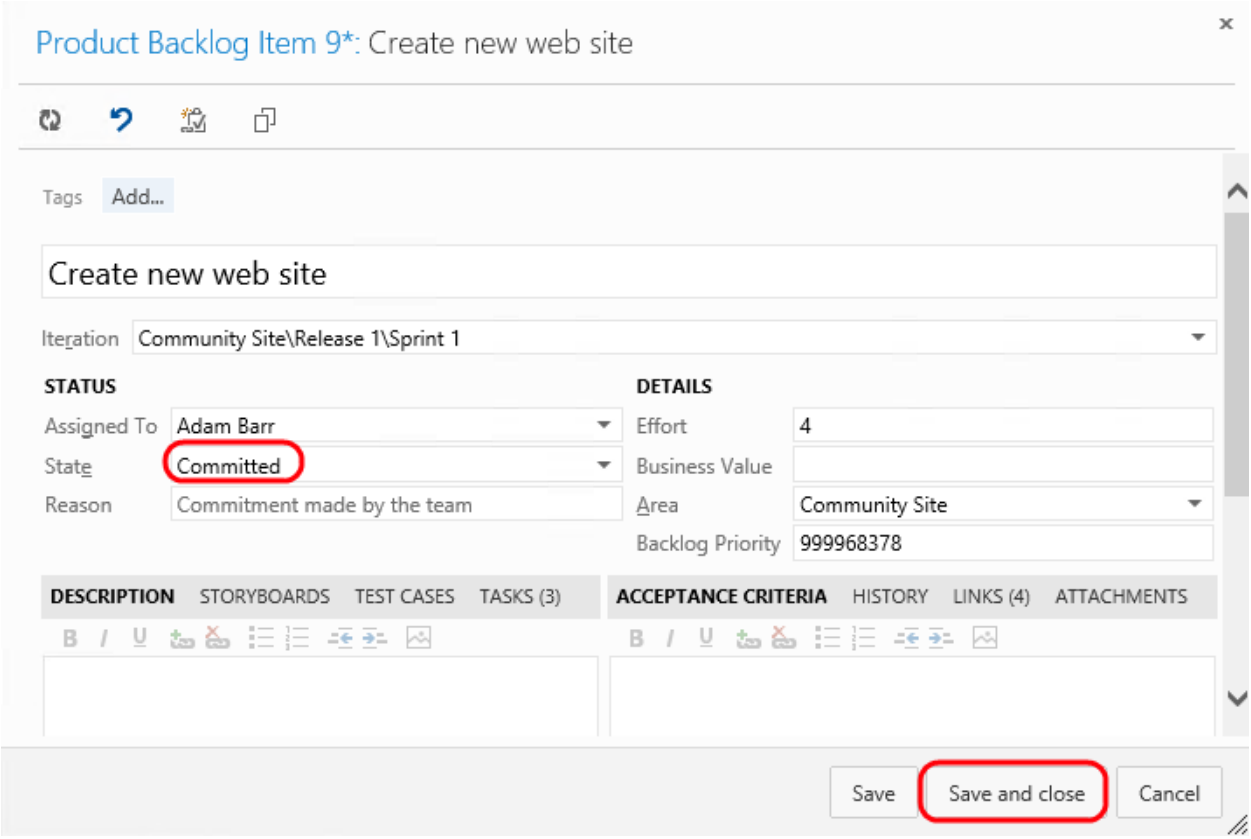
4. For the new task, enter a **Title** of "**Create new ASP.NET site**", **assign** to yourself, enter **Remaining Work** of '**2**', and then click the **Save and Close** button. Note that the Remaining Work of a task is measured in hours, so this is the point where your estimation skill will be fully utilized.



5. Create another task with **Title** of "**Add initial copy to site**", **assign** to yourself, and enter **Remaining Work** of '**5**'. Click **Save and Close**.
6. Create another task with **Title** of "**Add forum module**", **assign** to yourself, and enter **Remaining Work** of '**3**'. Click **Save and Close**.
7. Note that the sum of all remaining effort for tasks is rolled up and listed in the **Work Details** pane on the right side.



8. At this point of the sprint planning phase, nothing has been committed to yet. You are still free to move work between sprints, do additional capacity planning for teams, and adjust the tasks that are necessary in order to accomplish the work.
9. Commit to the PBI titled "**Create new web site**" by double-clicking it, changing the **State** to '**Committed'** and then clicking the **Save and Close** button.

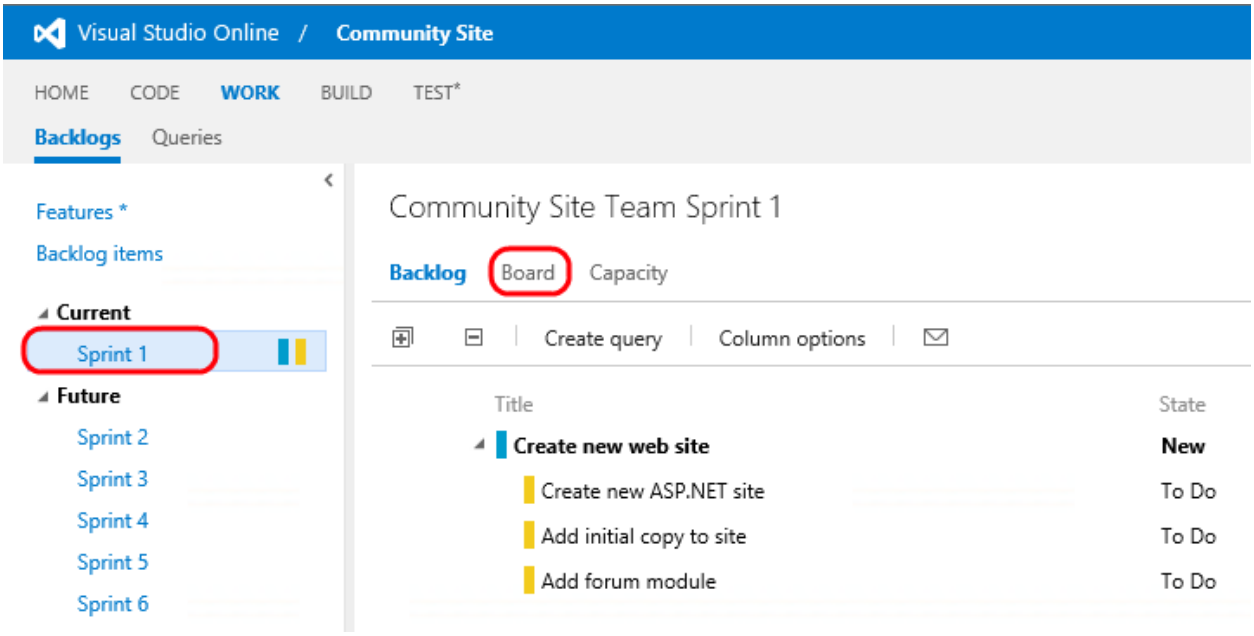


10. Although we are currently well within capacity for the first sprint, let's go ahead and move on rather than repeating the planning process with the other PBI.

Task 4: Managing Work during Iterations

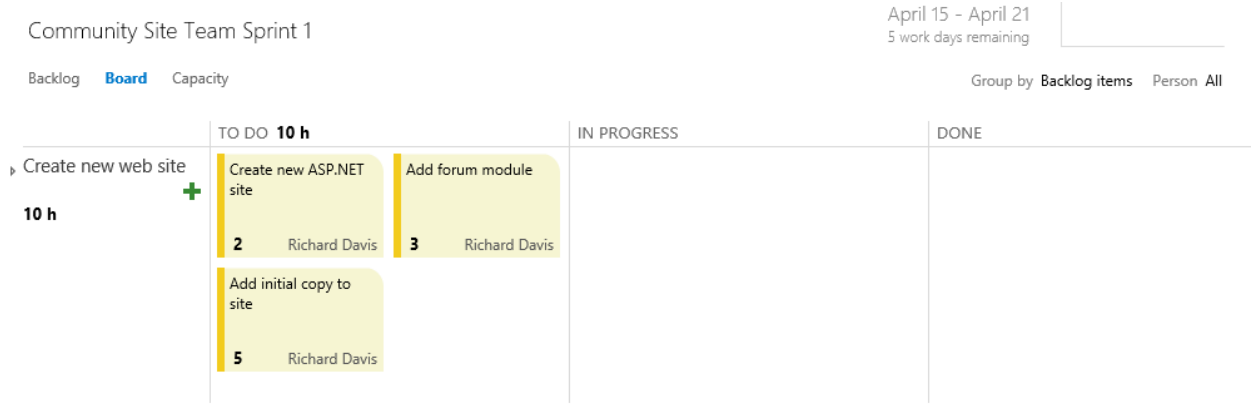
With iteration planning complete, now it's time to take a look at how an agile team might use the Visual Studio Online portal to manage the sprint backlog, as well as how management can gain insight into progress as it is being made during the sprint.

1. Select the **Board** link for the current sprint to open the task board.



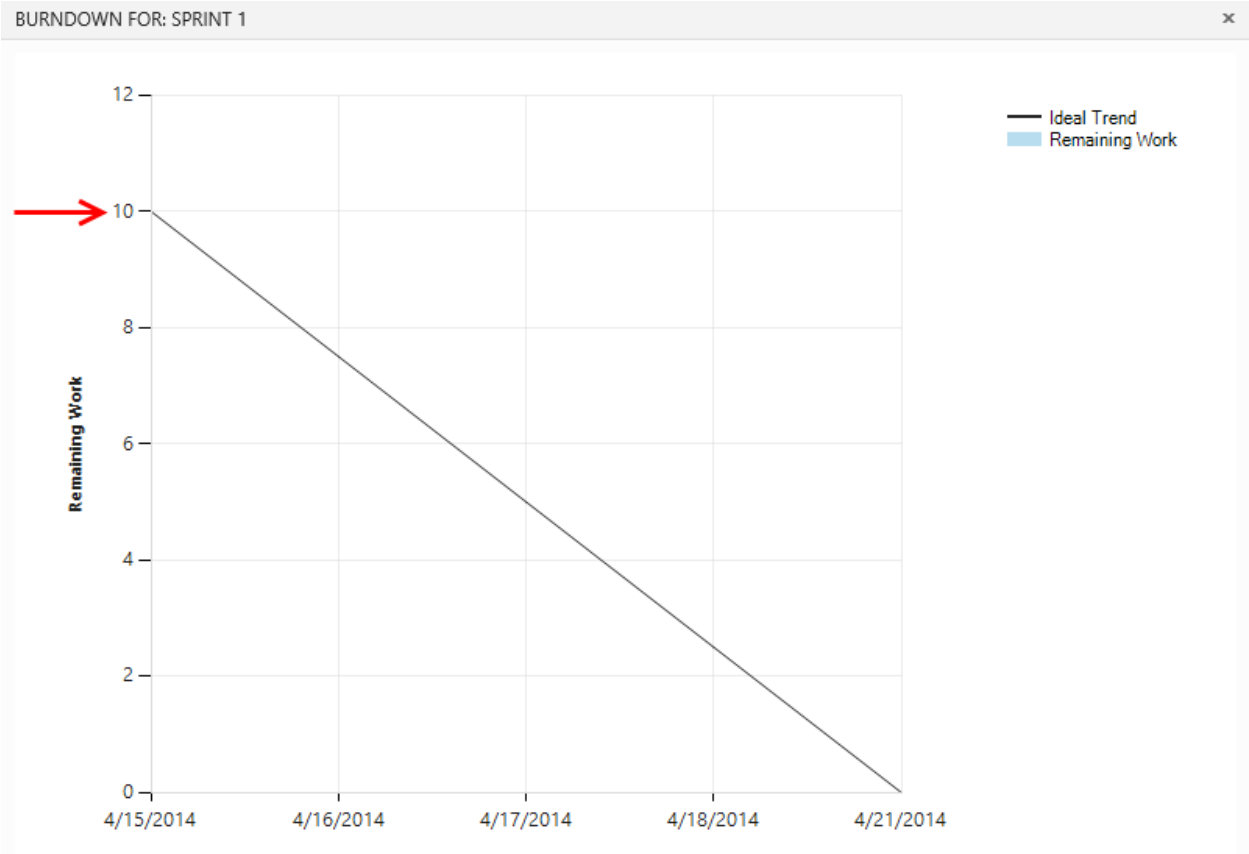
2. The task board shows rows of backlog items, with columns containing the current state and hours remaining for the associated tasks.

   Note: The states for tasks are driven by the process template selected for the project. In the case of the Scrum template that you selected (by default), the states include "To Do", "In Progress", and "Done".



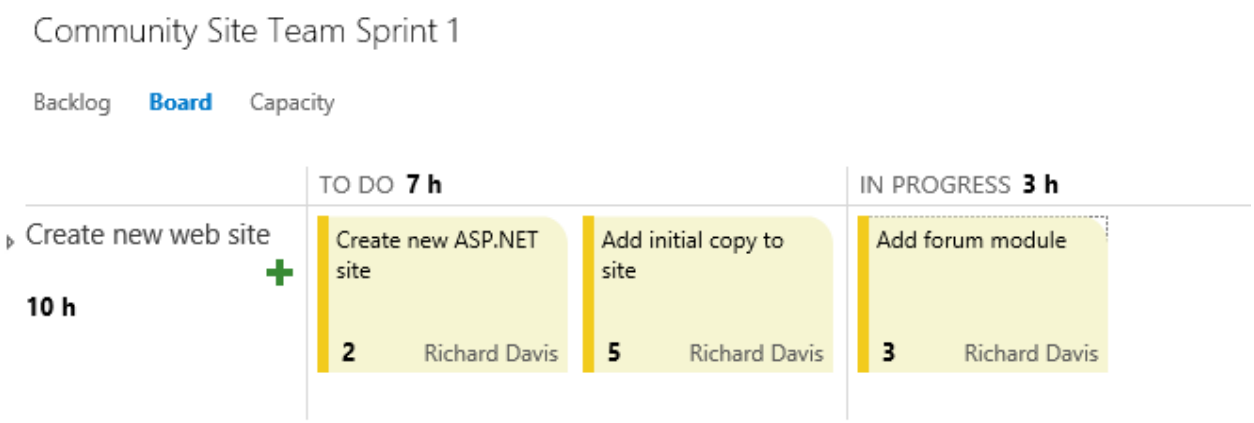
3. In the top-right corner is the **Burndown** chart, which can help track how much work remains in the sprint backlog. **Click** on the **chart** to expand it.

4. Although there is little value in the Burndown chart on the first day of the sprint, it is meant to show how much work remained at the end of specified intervals during the sprint. The horizontal axis shows days in a sprint, and the vertical axis measures the amount of work remaining to complete the tasks in the sprint. The black line represents the ideal trend line, which indicates that it will take about two hours per day in order to meet the schedule.

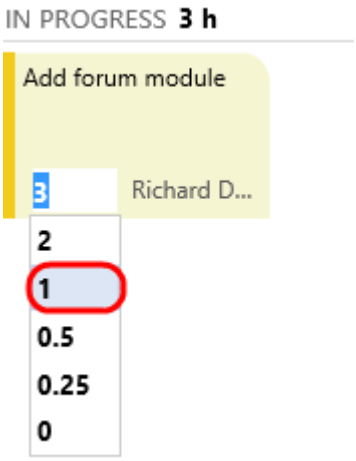


5. **Close** the Burndown chart.
6. Updates to task state and remaining work are easy to accomplish using the task board view. Drag and drop the "**Add forum module**" task to the **In Progress** column. This view shows a rollup of remaining hours for each Product Backlog Item (rows), and a rollup of remaining hours for each state (columns).

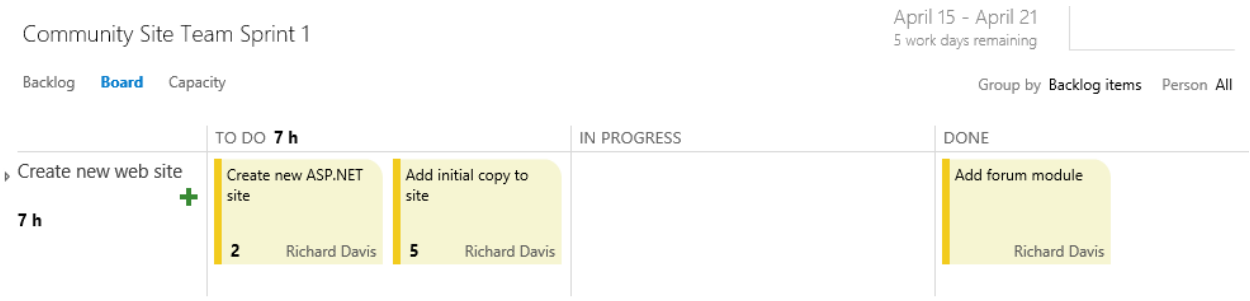
   Note: The task board is touch-enabled.



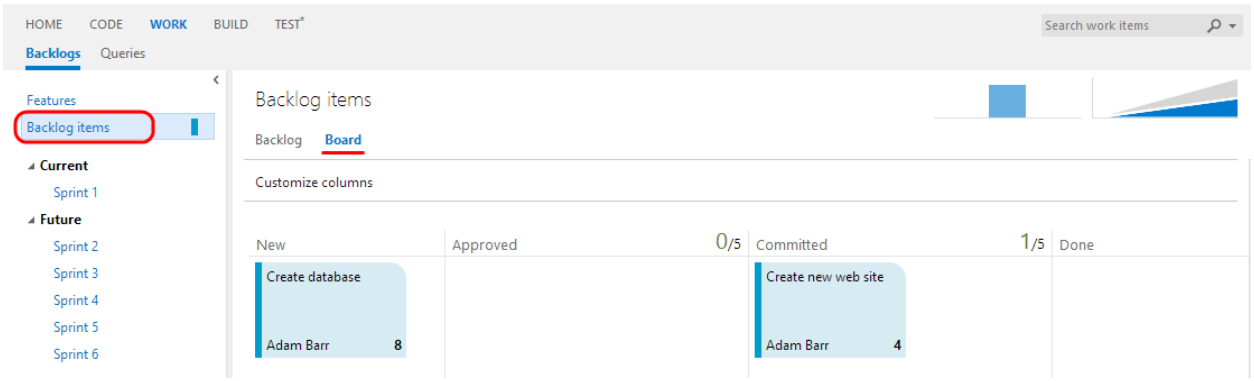
7. As the developers work, updates can be made to the remaining work field of tasks in order to help keep team leadership apprised of progress during the sprint. **Single-click** on the remaining work value for the '**Add forum module**' task and then select '**1**' hour to simulate working on the task for 2 hours.



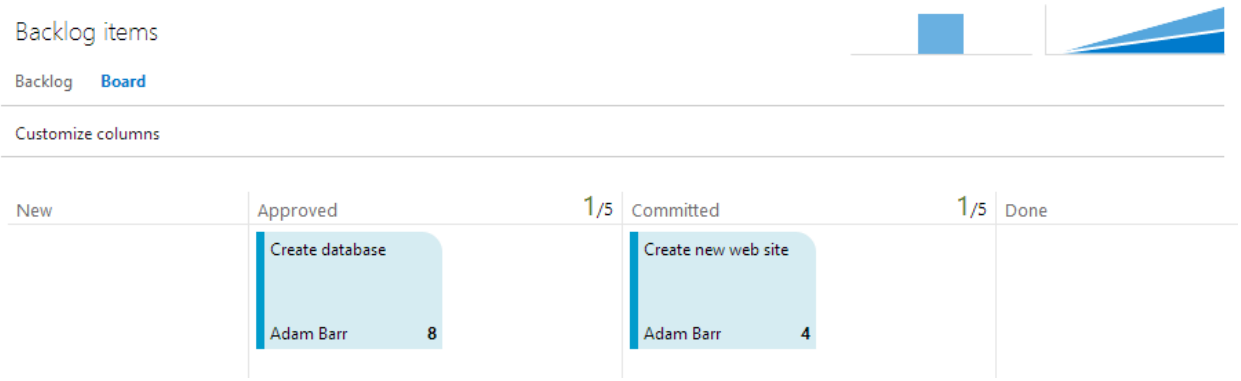
8. Each time remaining work is decreased (or increased due to unforeseen issues), the Burndown chart will be updated. If you were to view the chart now, you would notice that overall remaining work for the sprint went down by 2 hours. If you would like to get a better sense for how a real Burndown chart might look after a few days' worth of burndown, please take a look at this article on MSDN.
9. Drag and drop the "**Add forum module**" to the **Done** column to indicate that it has been completed. Note that the remaining work is automatically reduced to zero.



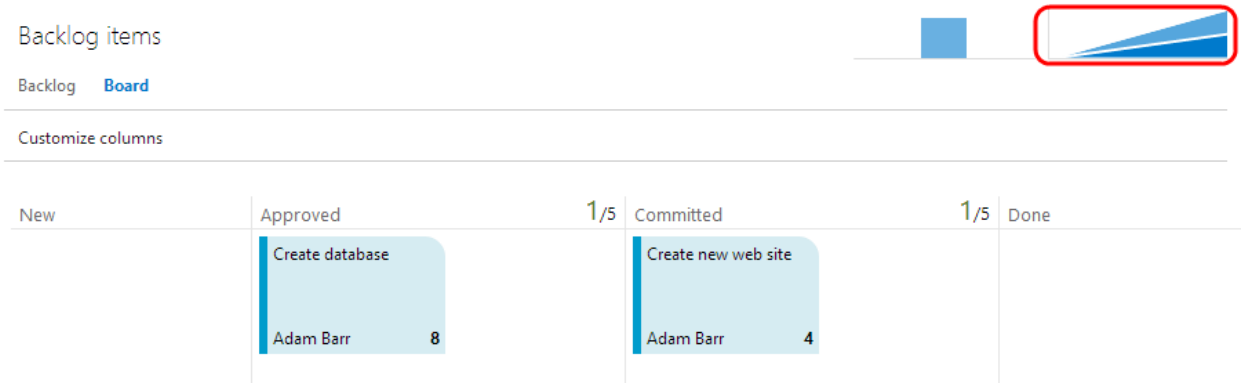
10. Select the "**Backlog Items**" link. Since you had the sprint task board open, you are automatically taken to the backlog board (rather than the backlog list view).



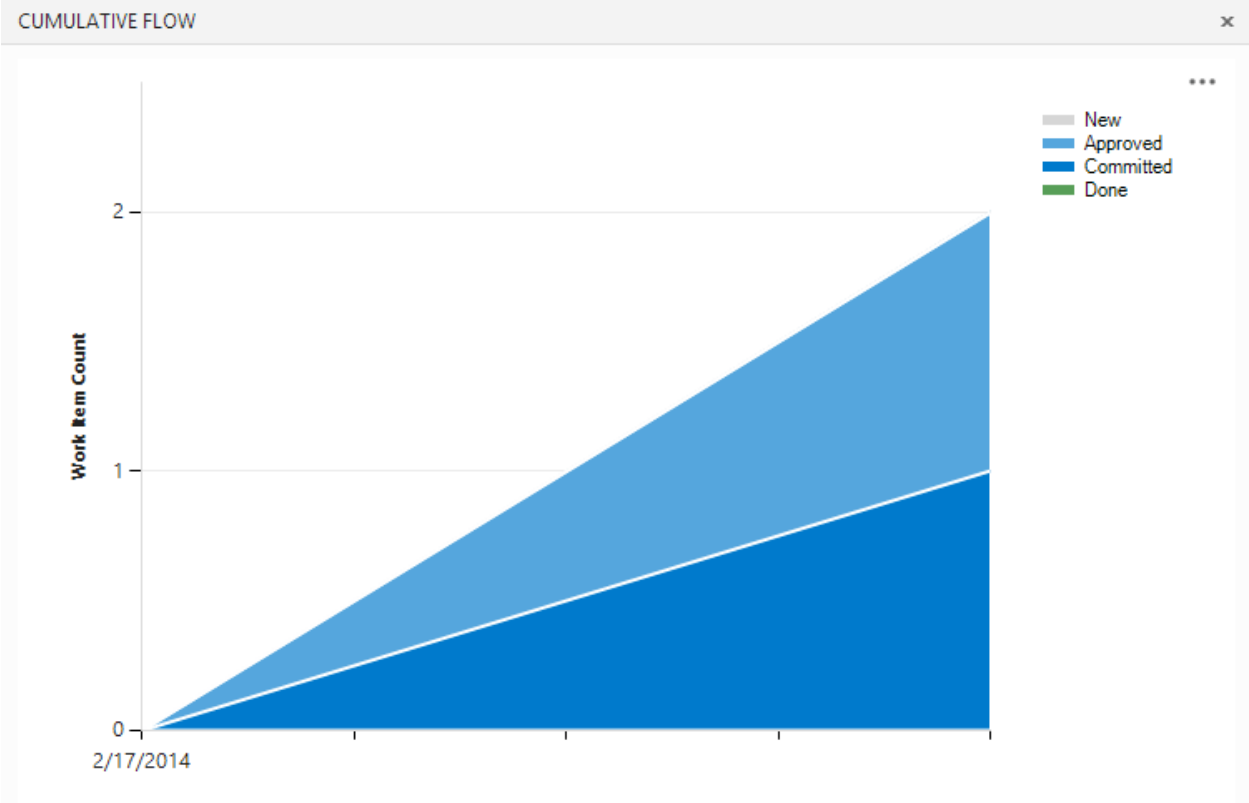
11. The board for the project backlog is implemented as a **Kanban** board. Kanban is a process improvement tool that can be used in an incremental fashion regardless of the current software development methodology that you are using. It assists with the throttling and tracking of work and illustrates the delivery of value over time to the project stakeholders.
12. Drag and drop the "**Create database**" item to the **Approved** column and take note that the Cumulative Flow Diagram is updated in the top-right corner of the Kanban board.



13. Click the small **Cumulative Flow Diagram** to open the larger view.

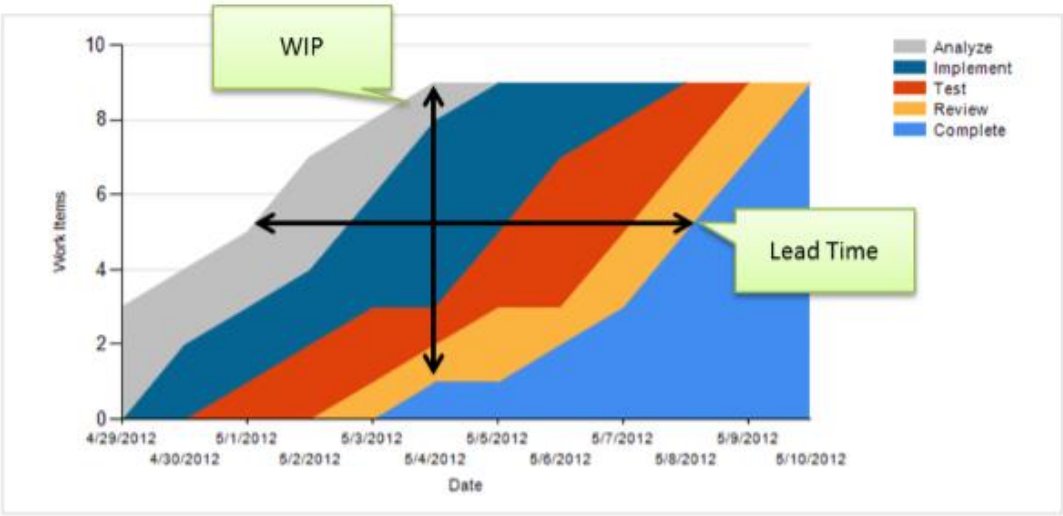


14. The **Cumulative Flow Diagram** shows the amount of work in various states over time, with horizontal axis showing lead time and the vertical axis showing work in progress.

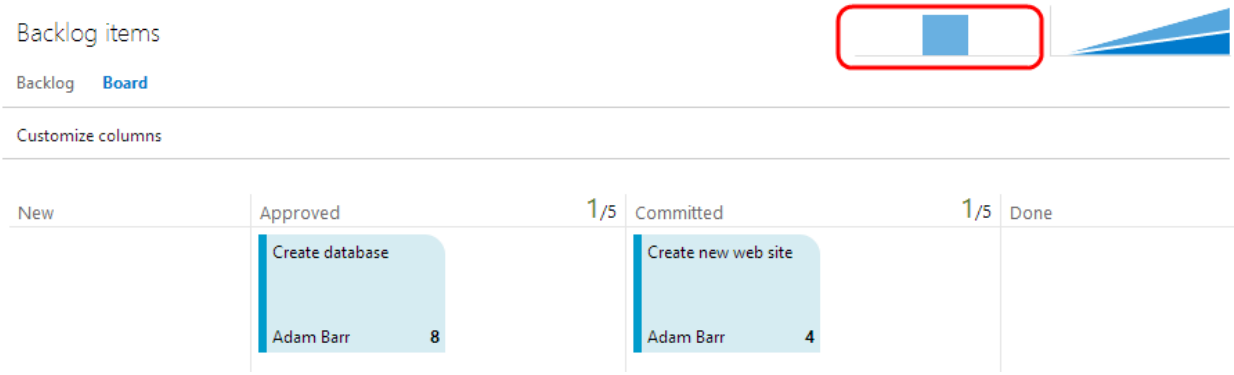


15. The diagram will not be a particularly illustrative of a real team that has days' worth of data represented, but you should be able to get a sense of how this would show work in progress over time. Lead time can be estimated by looking at the number of days it typically takes the work to go from the 'New' state to the 'Done' state.

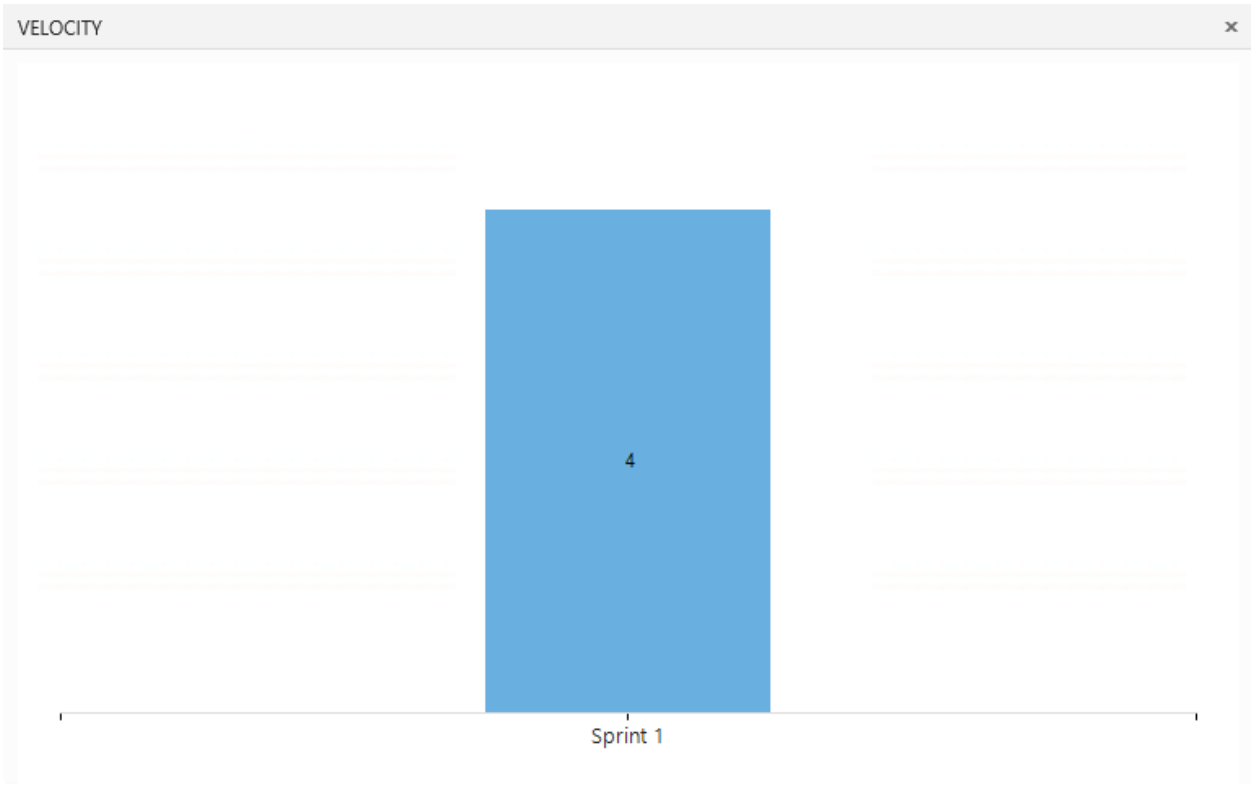
    Note: This simple graph illustrates what a real CFD might look like, better illustrating the value of the visualization.



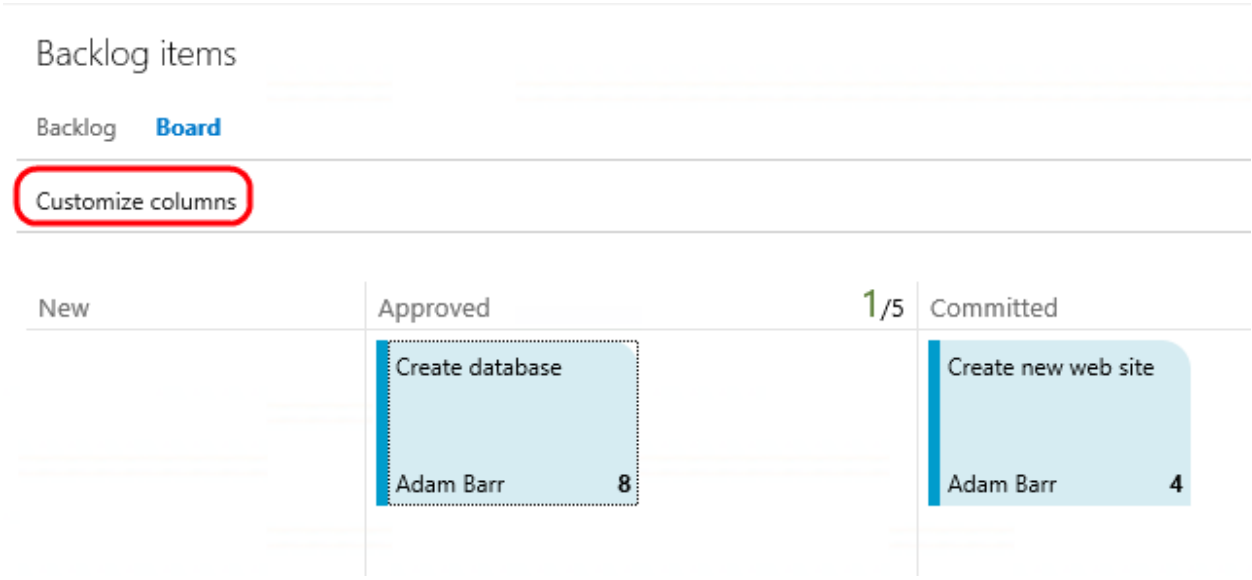
16. **Close** the Cumulative Flow Diagram window.
17. Click the small **Velocity** graph in the top-right corner of the Kanban board.



18. The **Velocity** graph is used to view how much effort that the team has committed to and completed, although currently we only see that '4' effort was committed to by the team during the first sprint. Although we will not do so in this lab, the team can use velocity to estimate how much backlog effort your team can handle in future sprints, assuming that your team composition and sprint duration stay constant.



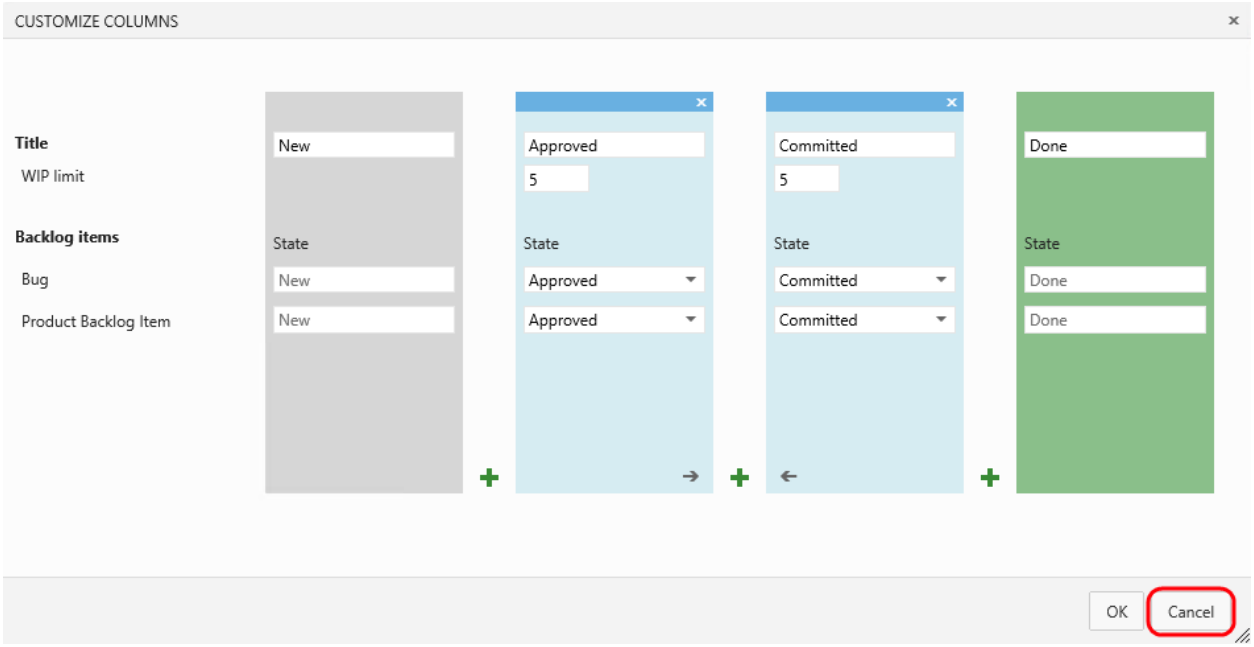
19. **Close** the Velocity graph.
20. The default columns match up with states allowed for Product Backlog Items, as defined by the Scrum process template. However, with Kanban boards you have some additional flexibility to customize the columns and even add in new states if desired. Click the **Customize Columns** button.



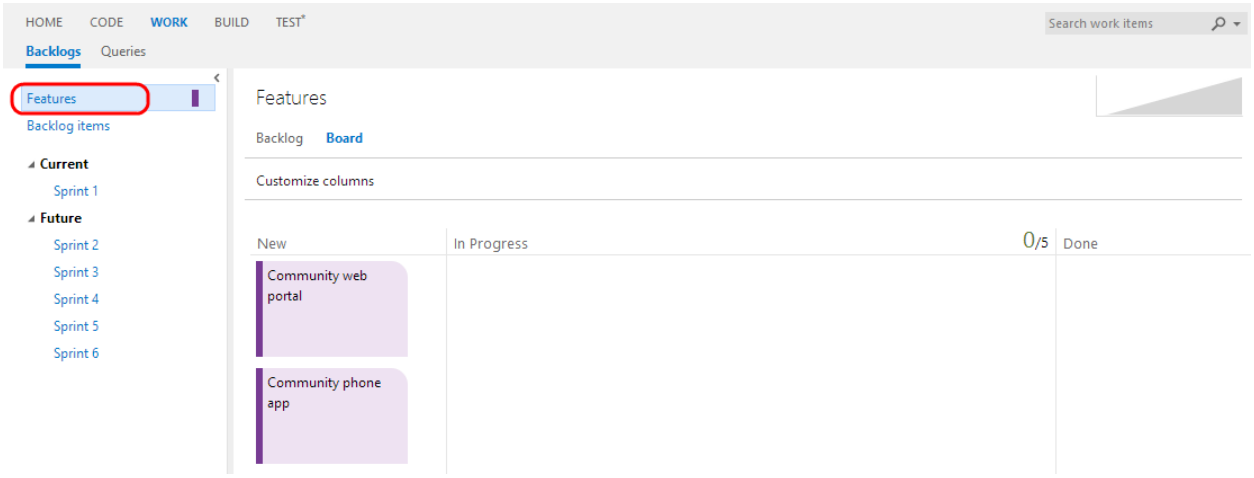
21. The **Title** field can be set to anything that you would like, and is independent of the requirements imposed by the process template itself. However, keep in mind that each column

ultimately needs to be mapped to a valid state from the process template and respect the specified state transitions. By default, each column title matches up directly with the actual back log item states.

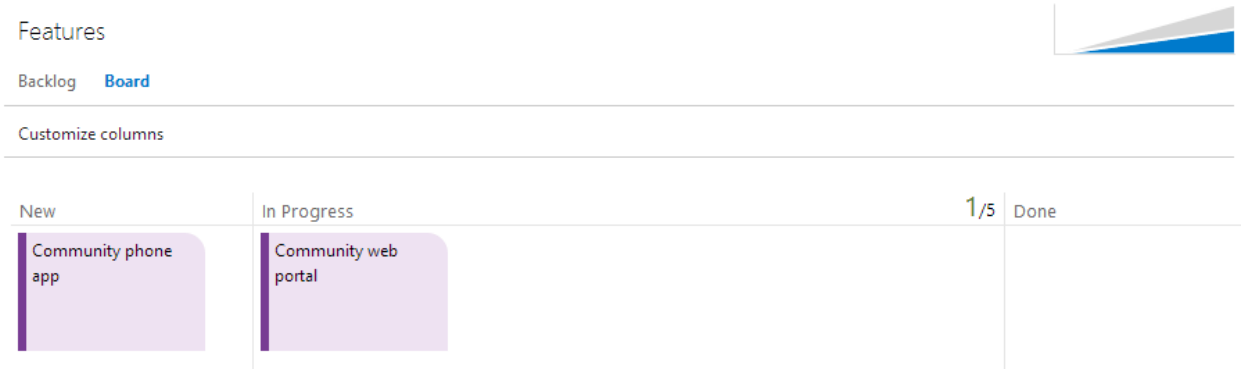
22. Click the **Cancel** button.



23. Select the **Features** link to load the feature Kanban board.



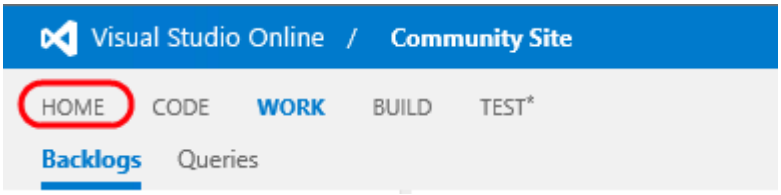
24. It is important to note that completing Tasks from the sprint backlogs do not automatically change the state of their parent Product Backlog Items, nor does updating Product Backlog Item state update their parent Features. It's ultimately up to team members to communicate and coordinate the updates to the various backlogs.
25. Drag and drop the '**Community web portal**' feature to the **In Progress** state so that you can better keep project management apprised of progress. Note that there is a Cumulative Flow Diagram shown at the Feature level as well.



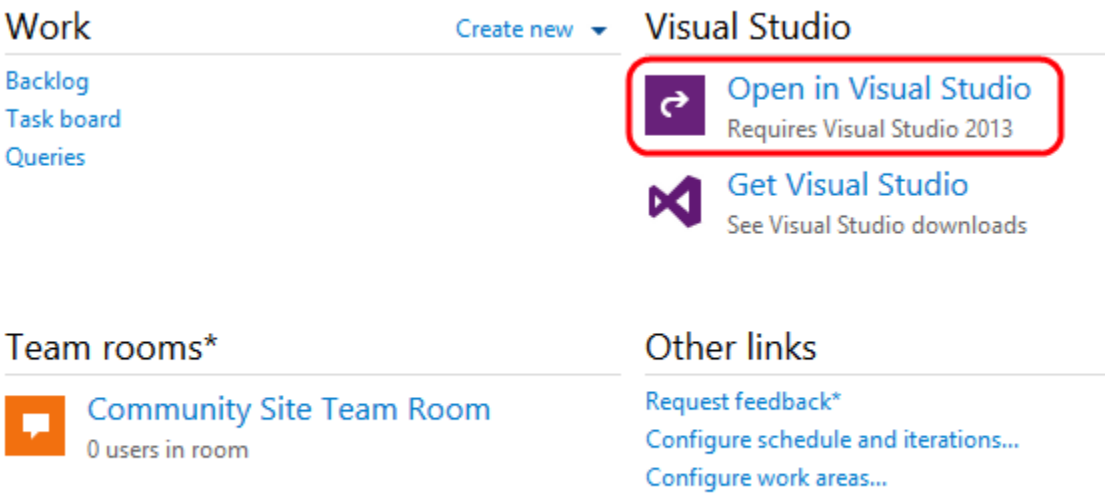
Task 5: Connecting to the Project with Visual Studio

In this task, you will see how easy it is to get started developing with your team project hosted on Visual Studio Online.

1. Let's say that the development team is ready to start working on their assigned tasks. Although they can certainly work with task and bug definitions in the VSO portal, they will probably be spending more of their time in Visual Studio. Select the Home link to return to the project overview page.

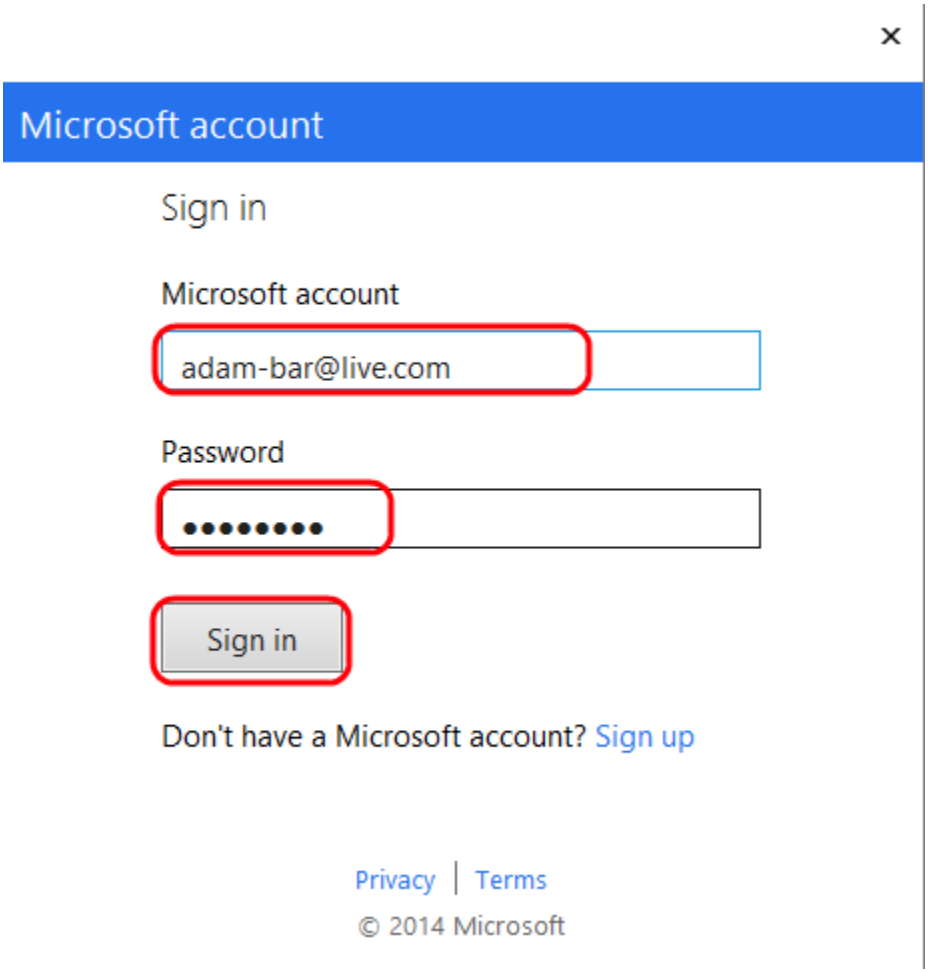


2. Select the "**Open in Visual Studio**" link found on the Visual Studio Online landing page for the project that you created during the previous task.

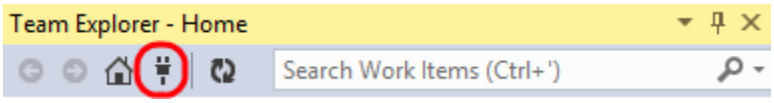


Note: You may be prompted to confirm the launching of the Microsoft Visual Studio Web Protocol Handler Selector. Click the **Allow** button to confirm.

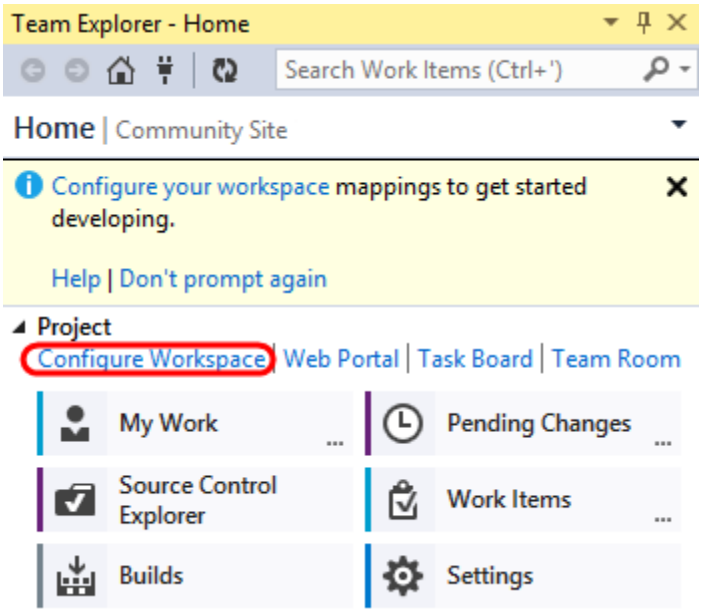
3. Sign into your Microsoft account when prompted in Visual Studio.



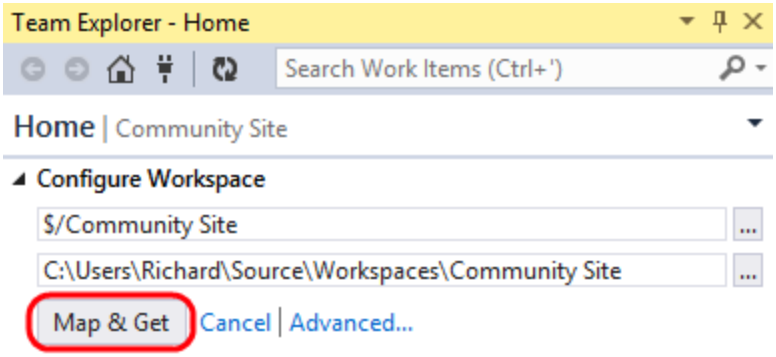
4. After signing in, you should automatically be connected to the new team project and see the **Team Explorer – Home** panel. For future reference, you can change the server and project that you are connected to by clicking on the **Connect to Team Projects** button in the Team Explorer toolbar.



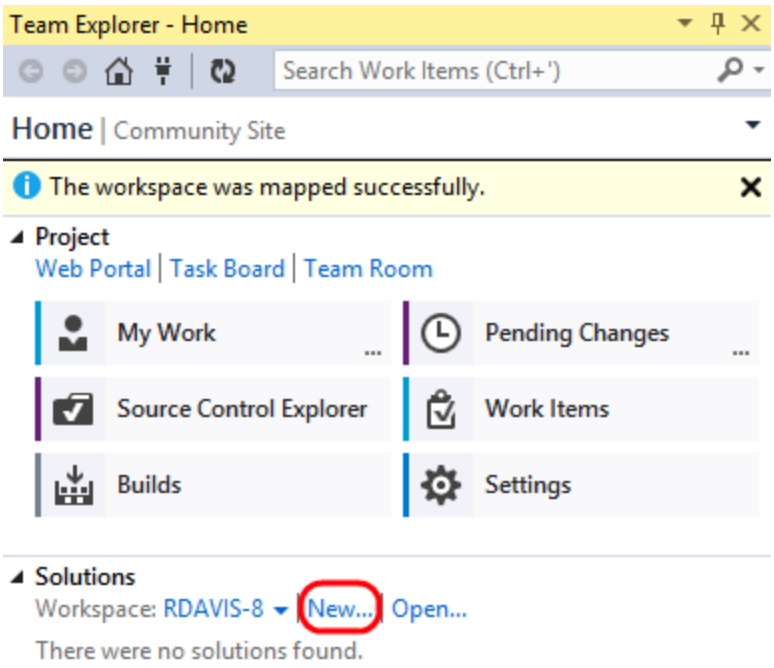
5. Choose the site then select the **Configure Workspace** link.



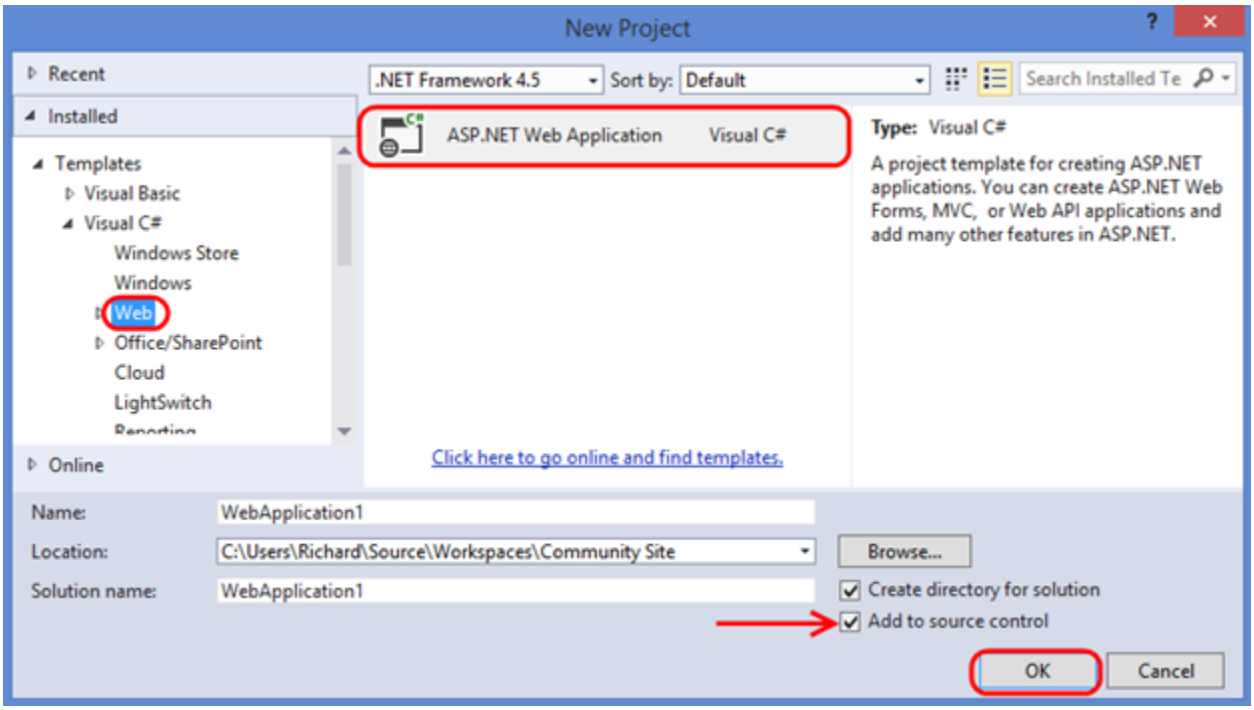
6. Click the **Map & Get** button to use the default local workspace mapping.



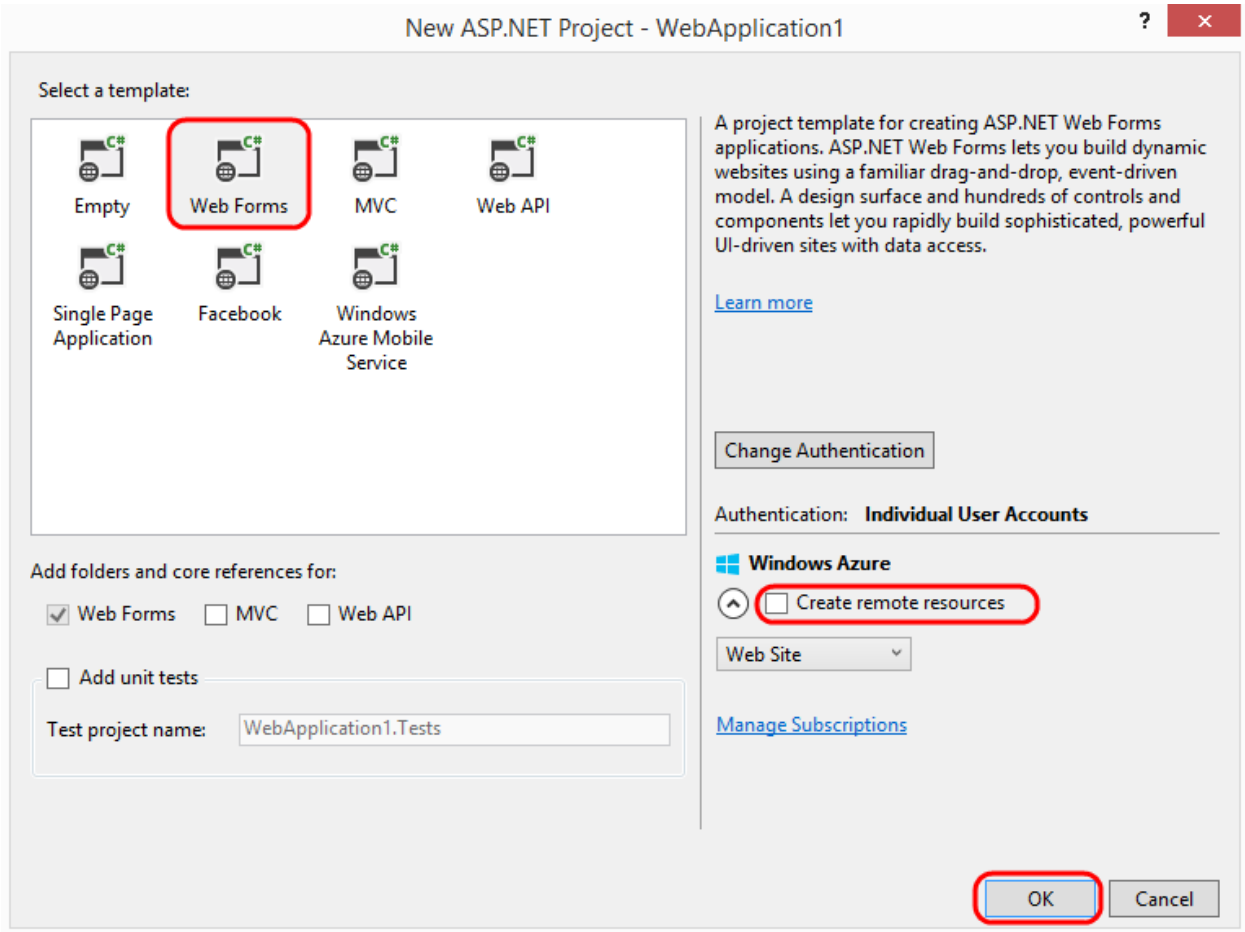
7. Select the **New…** link from the Solutions section of the Team Explorer – Home panel.



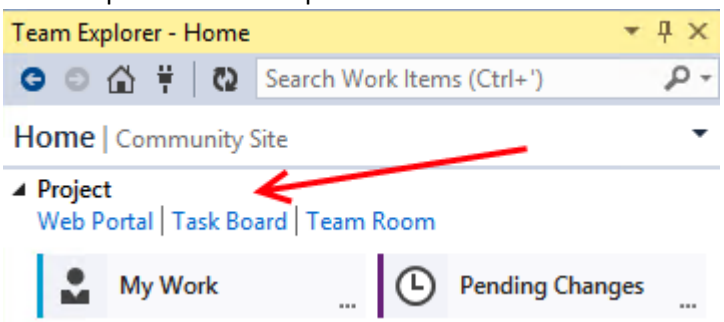
8. Select the **Visual C# | Web** node from the list of installed project templates, ensure that the checkbox for "**Add to source control**" is checked, and then click the **OK** button.



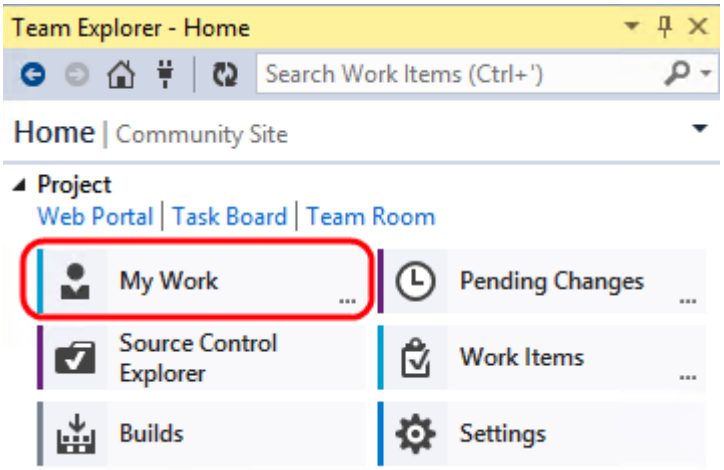
9. In the **New ASP.NET Project** window, select **Web Forms**, de-select the option to "**Create remote resources**", and then click **OK**.



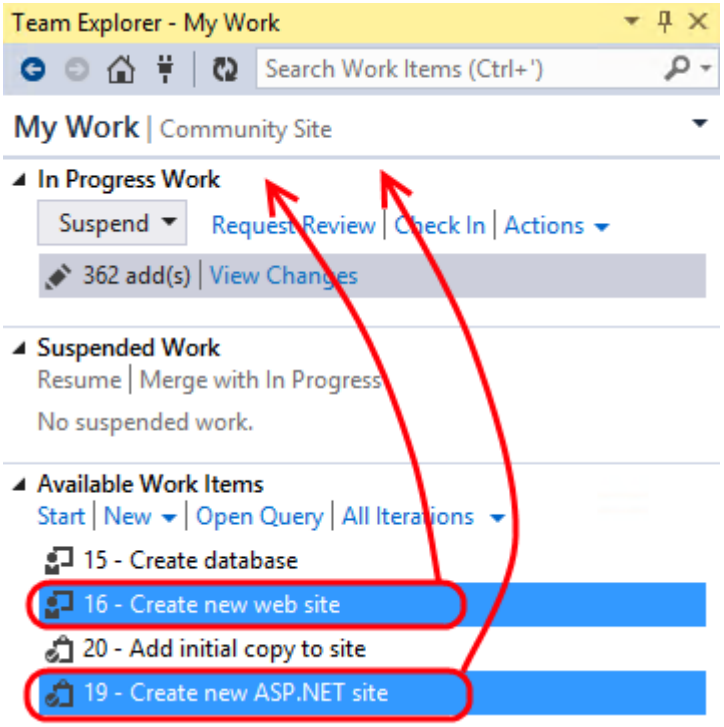
10. As a quick aside, note the links to the web portal, the task board, and team room for the project at the top of the Team Explorer – Home window.



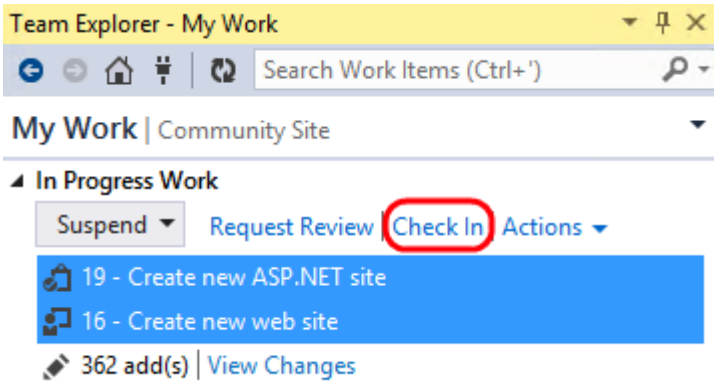
11. As a developer, you can now take a look at the tasks that have been assigned to you. Click the **My Work** tile.



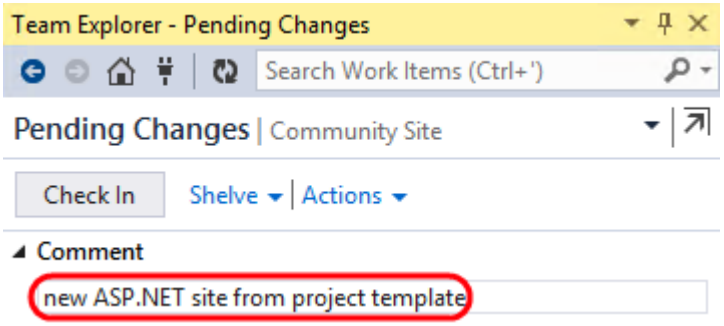
12. The work items that are assigned to you can be seen in the **Available Work Items** section. Drag and drop the Product Backlog Item titled '**Create new web site**' and the Task titled '**Create new ASP.NET site**' to the **In Progress Work** section. This will associate the work items with the source when we check it in.



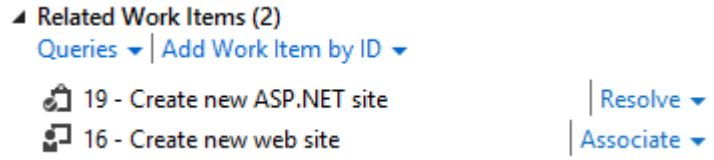
13. Select the **Check In** link.



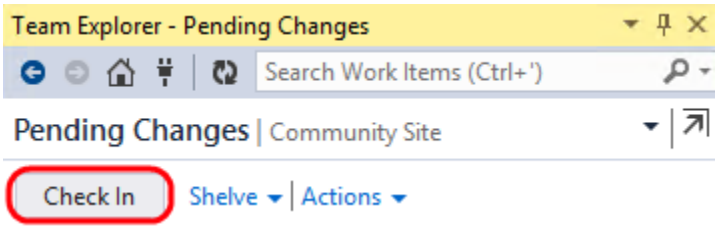
14. In the **Team Explorer – Pending Changes** window, add an appropriate comment such as "**new ASP.NET site from project template**".



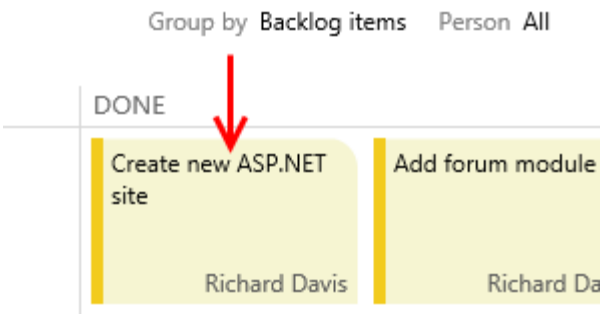
15. Note that the Task work item is set to **Resolve** by default, and the Product Backlog Item is set to the **Associate** option. Resolving a task will set the state to 'Done' and set the 'Remaining Work' field to zero.



16. Click the **Check In** button to upload the source and resolve/associate the work items in your Visual Studio Online project. You may need to confirm the check-in and/or save open files in order to proceed.



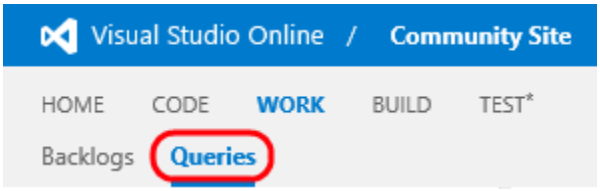
17. Return to the task board for the current sprint and note that the Task is shown as being **Done**.



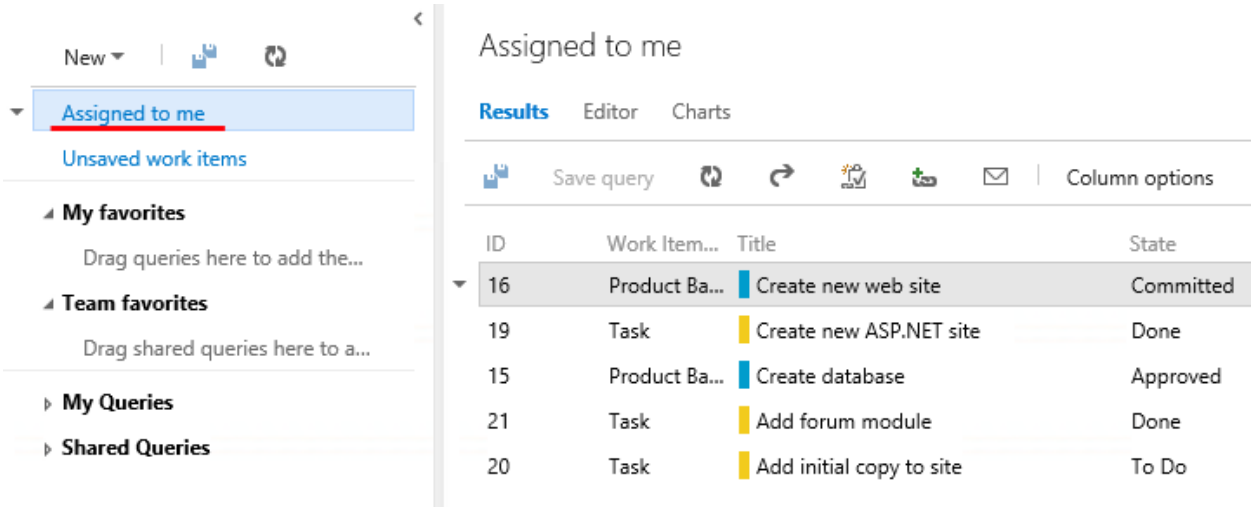
Task 6: Working with Work Item Queries and Charts

You can view and create detailed work item query definitions in Visual Studio Online.

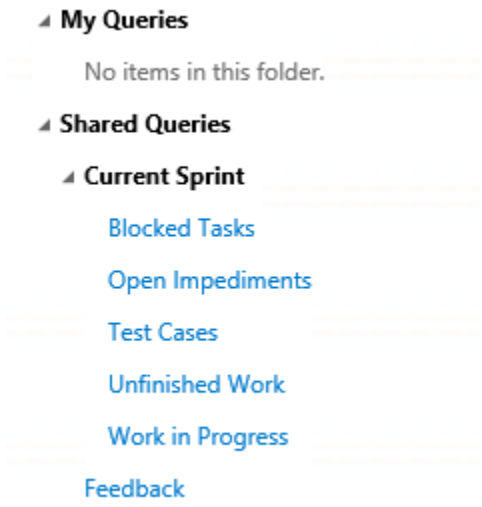
1. Navigate to the **Work | Queries** view in Visual Studio Online.



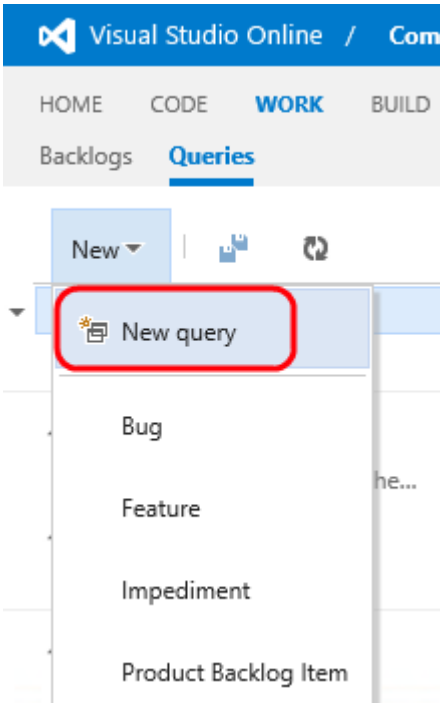
2. You should now be looking at the results of the built in "**Assigned to me**" query that shows all work items assigned to you. This query view offers a lot of functionality including the ability to multi-select work items, assign selected work items to a new or existing work item, send the query in email, and even bulk edit.



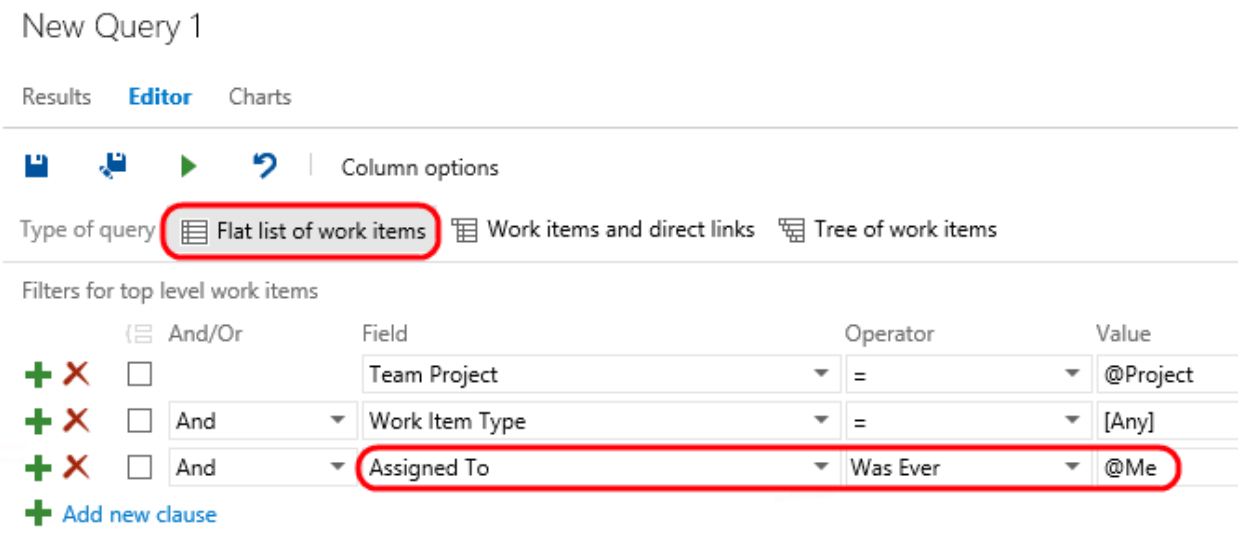
3. You can create and save your own queries, which can either show up under the "My Queries" section or the "Shared Queries" section. Expand the "**Shared Queries**" section to view the default queries that were created with the new project.



4. Create a new query by clicking the **New** drop-down and selecting the **New Query** option.

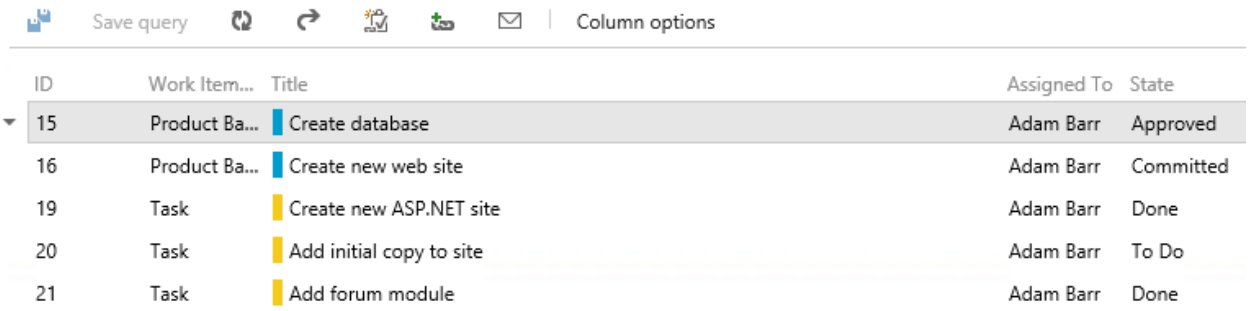


5. Accept the default option for query type, which is a flat list of work items. Modify the last filter clause so that it reads "**and Assigned To Was Ever @Me**".

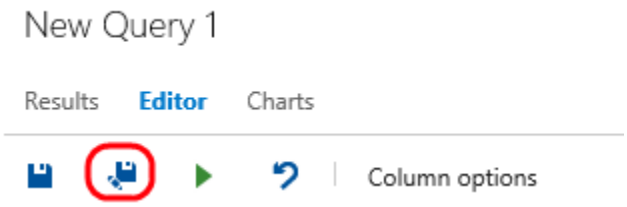


6. Select the **Run Query** button to view the results of the query definition.

7. The result of the query should show all of your work items for the current project that were ever assigned to you.

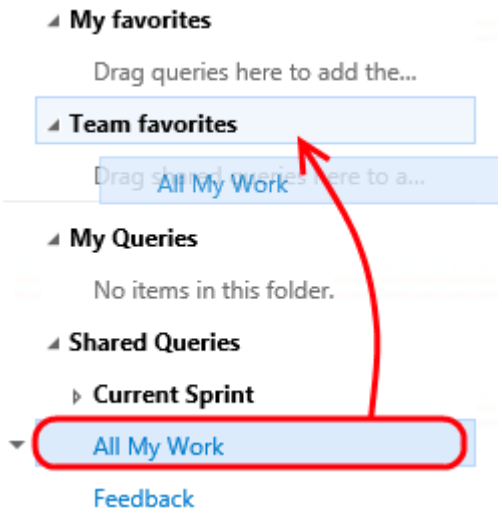


8. Click the **Save Query As** button.



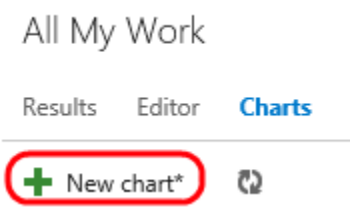
9. Name the query "**All My Work**", select the **Shared Queries** folder, and then click the **OK** button.

10. Drag and drop the new "**All My Work**" query to the team favorites section. You can use favorites as a way to group and provide quick access to commonly used queries.

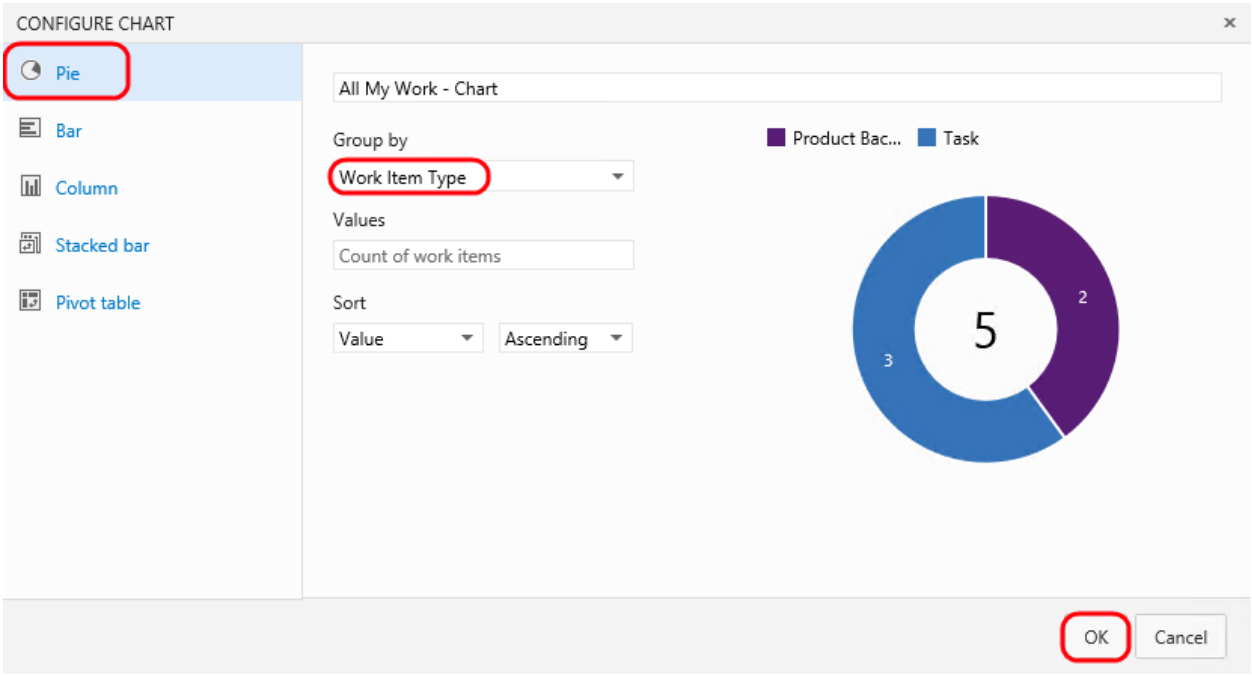


11. Select the **Charts** link at the top of the query results view. Work item charts are based on work item queries.

12. Click the **New Chart** button.



13. There are a few different chart types to choose from. Use the default **Pie** chart option, group by **Work Item Type**, and then click the **OK** button.



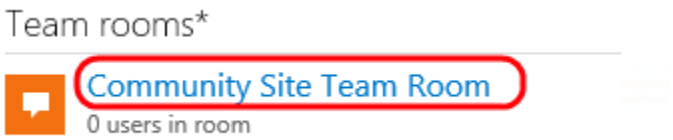
Exercise 4: Collaboration and Communication during Development

In this exercise, you'll learn about team rooms, code comments, requesting feedback, and how to add some additional project metrics and work item graphs to the team overview page.

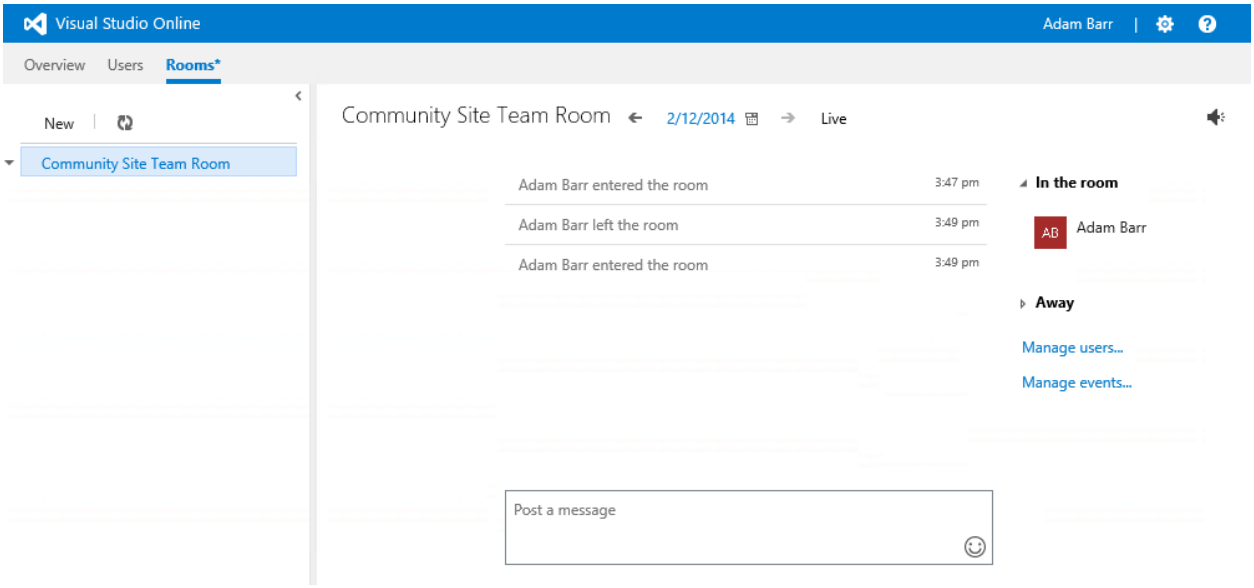
Task 1: Team Rooms

Team Rooms provide a durable collaboration space where members can chat and view pertinent events, thereby allowing them to remain in loose contact throughout the workday.

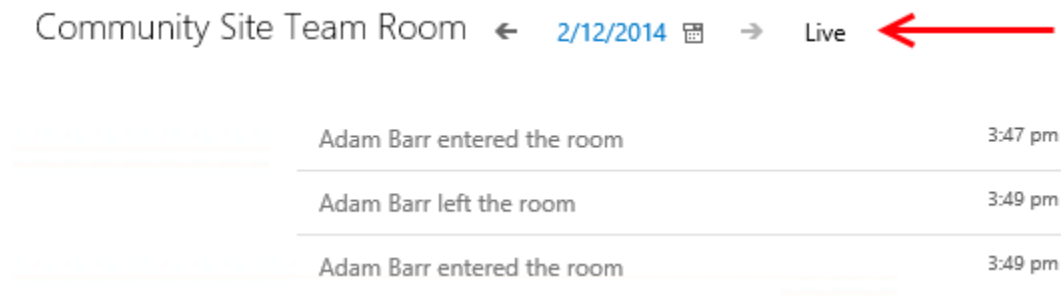
1. By default, each team that is created will also have a team room created automatically. Navigate to the project overview page and then select the **Community Site Team Room** link.



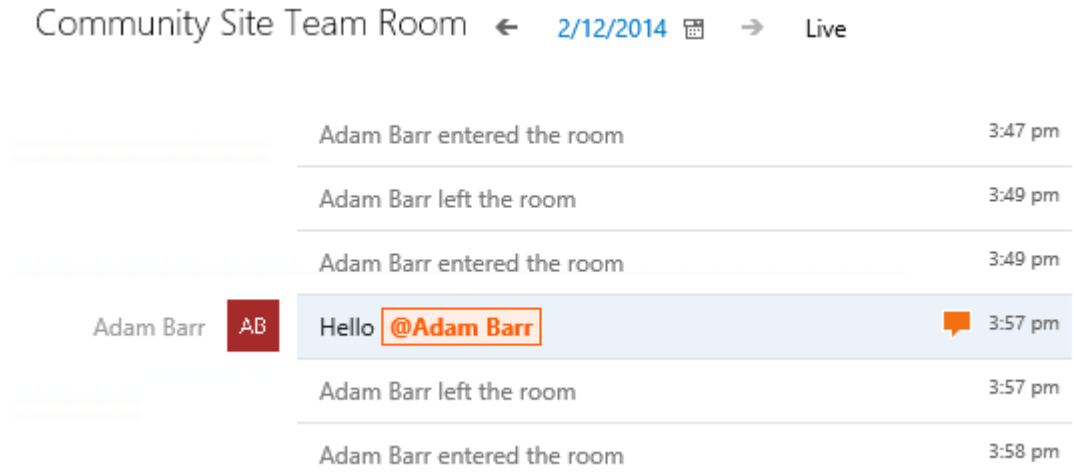
2. The left side shows the **Rooms Explorer** pane, which you can use to navigate between team rooms and perform some administrative capabilities. On the right side is the currently selected team room showing the recorded messages and activity for the current day.



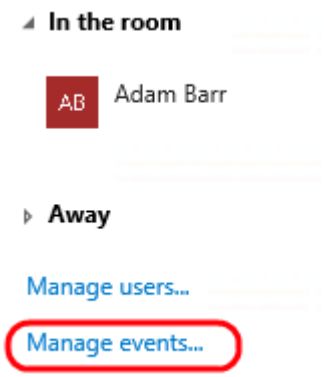
3. Since the team room content is persistent, team members can go back in time to see what happened while they were away from work. Take note of the left and right arrows and the date picker control at the top of the chat window. At the moment, you can see that you are "**Live**".



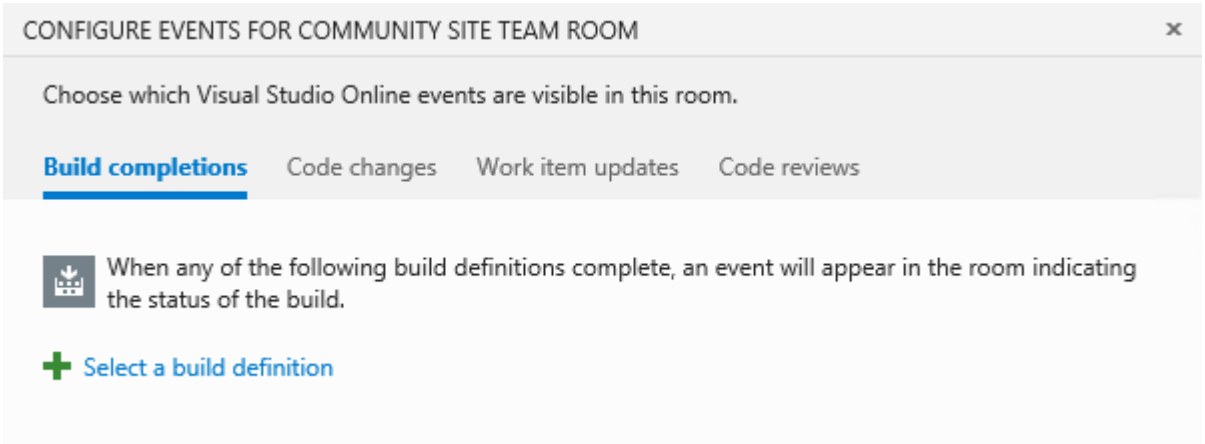
4. Although we won't go into too much detail in this lab, it is important to note that there are a number of different mentions that can be used in the team chat such as team member mention (type the '**@**' symbol), and work item mention (type the '**#**' symbol plus the work item number).



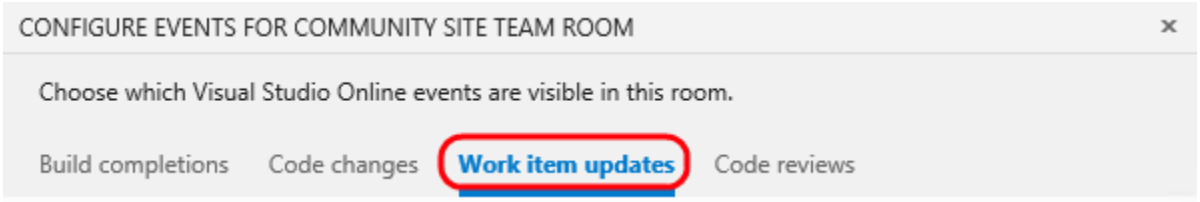
5. Another key feature of team rooms is that various events can be selected and configured to automatically feed in additional information, thereby providing additional context to active and historical discussions.
6. Select the **Manage events…** link.



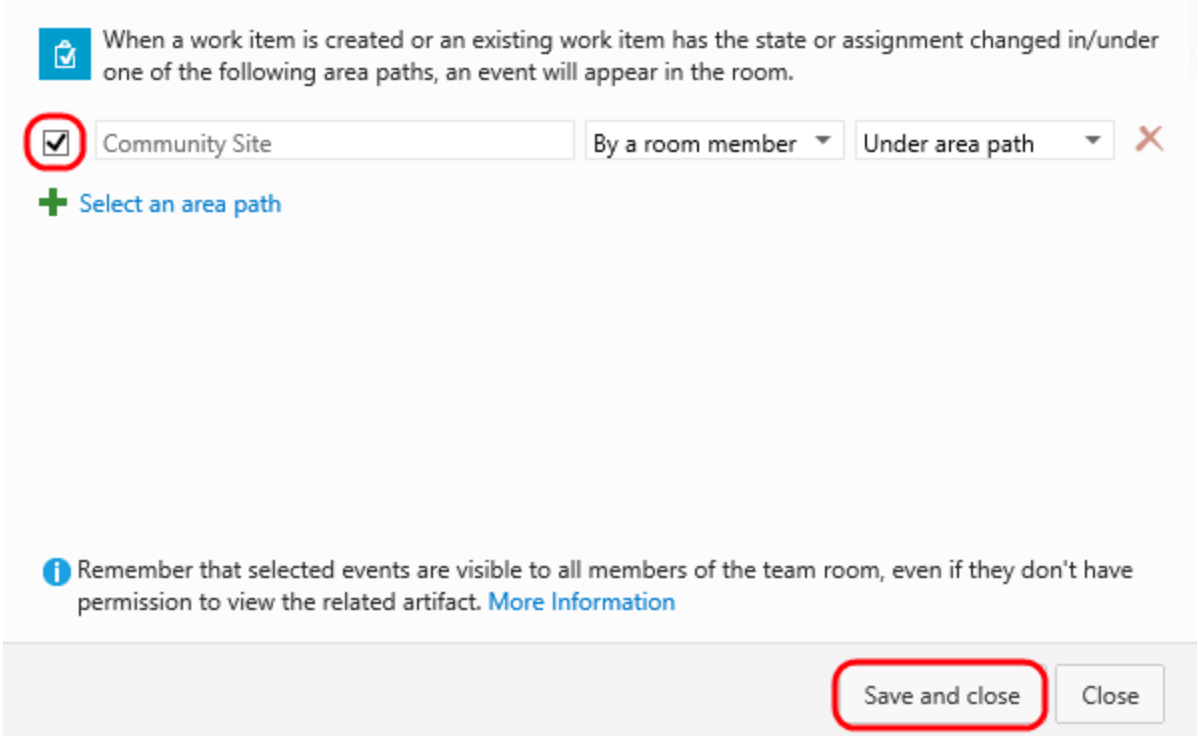
7. The Configure Events for Community Site Team Room window that appears shows four different options – **Build Completion**, **Code Changes**, **Work Item Updates**, and **Code Reviews**. Feel free to navigate through these different event sources to get an idea of what is available.



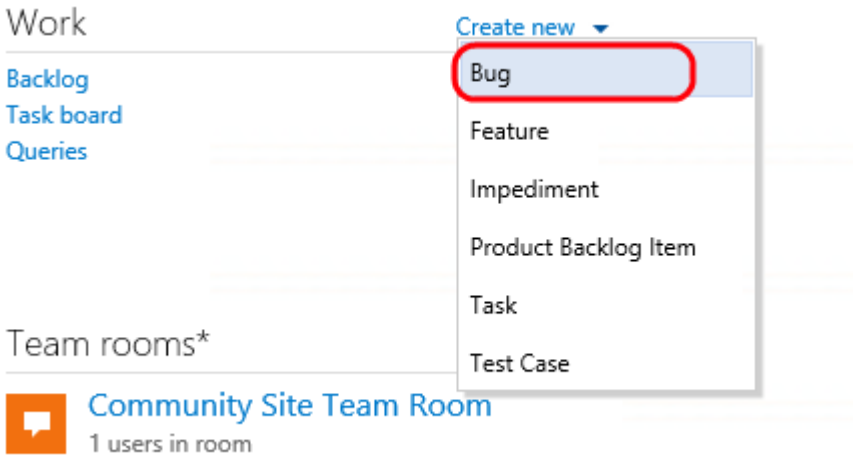
8. Select the **Work Item Updates** tab.



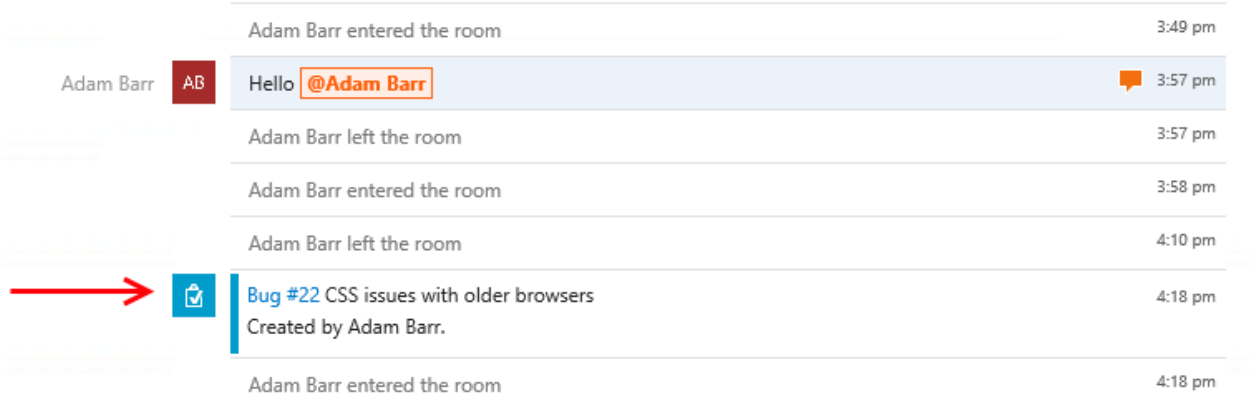
9. Check the box next to the "**Community Site**" area path and then click the **Save and Close** button.



10. Now you will change the state for one of the work items to verify that the team room event has been hooked up. Navigate back to the Community Site project overview page, select the "**Create new**" drop down from the Work section, and then select the **Bug** option.



11. Title the bug "**CSS issues with older browsers**" and then click the **Save and Close** button.
12. Return to the team room and take note of the event.

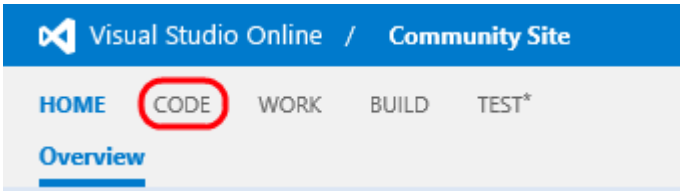


13. You can create as many team rooms as you would like to in order to keep the information flow focused and on topic for team room participants. For example, you may have a situation where you want to have your UI specialists in a room that is separate from the database gurus, although they are all on the same Community Site team.

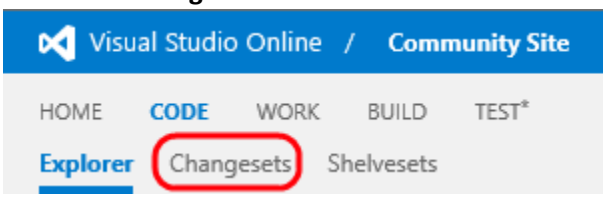
Task 2: Code Comments

It is possible for team members to comment on and have lightweight discussions about code within the Visual Studio Online account.

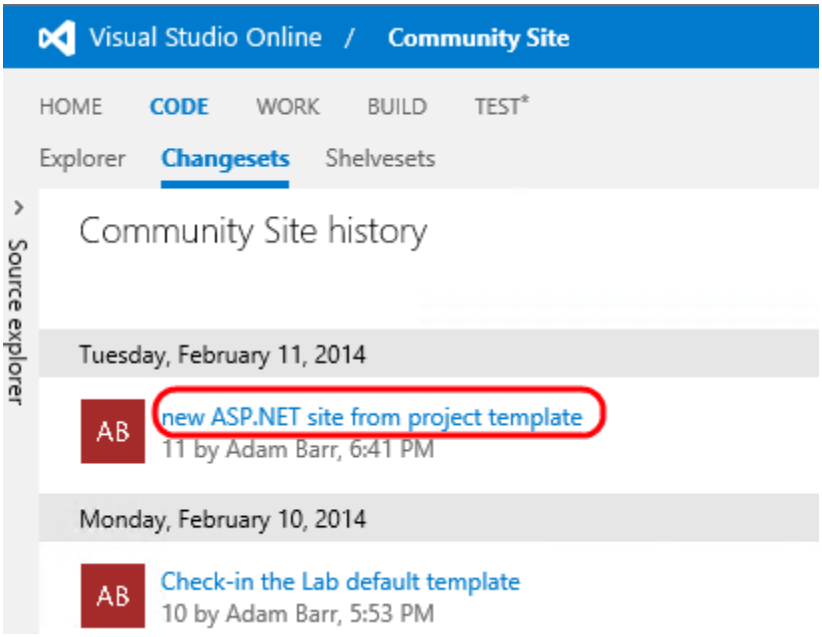
1. Navigate to the Community Site project overview page and then select the **Code** link.



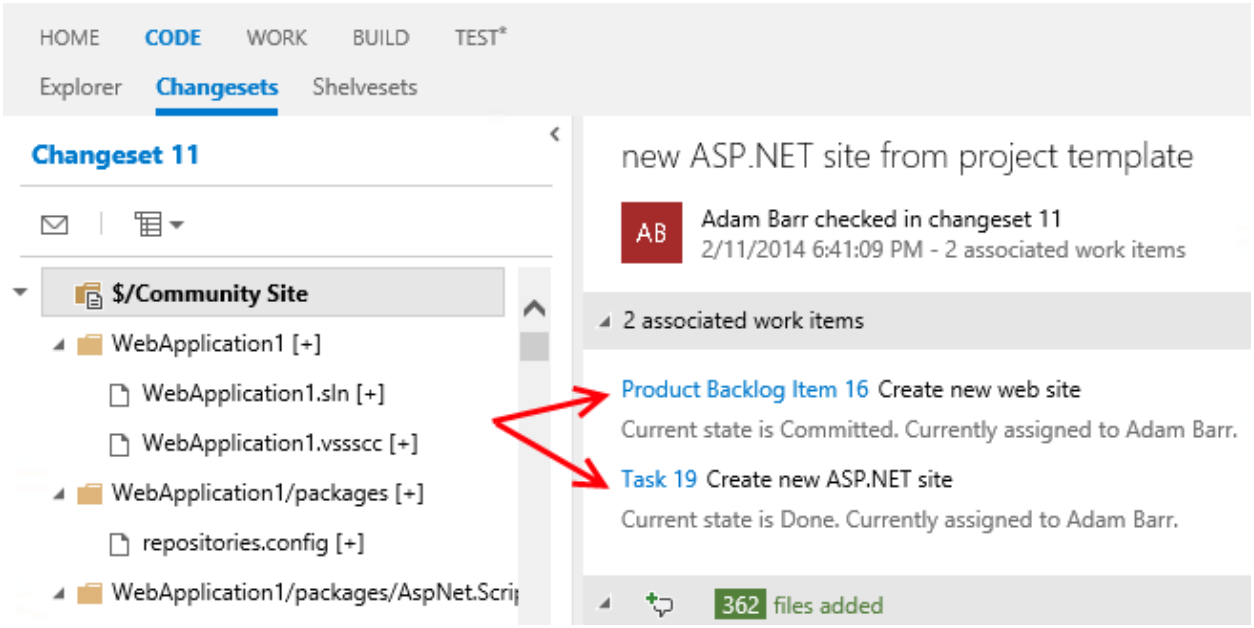
2. Select the **Changesets** link. Code comments can be applied to both changesets and shelvesets.



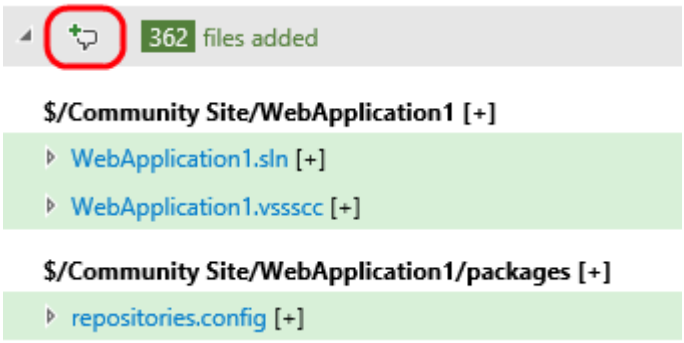
3. Select the link for the changeset that is associated with the check-in that you did earlier, titled "**new ASP.NET site from project template**".



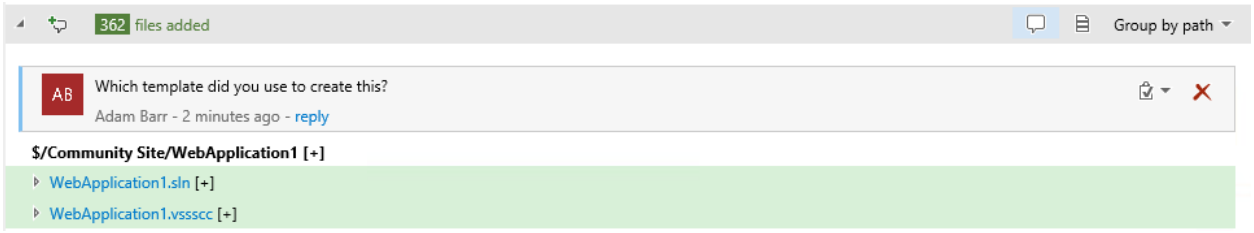
4. As a side note, you can see that this changeset was associated with the specified work items when you checked it in earlier.



5. Select the **Add Comment** button.

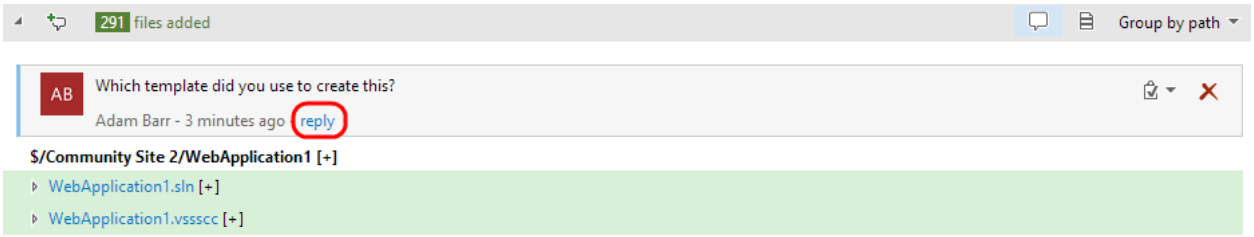


6. Type a comment and then press the **Enter** key when done.

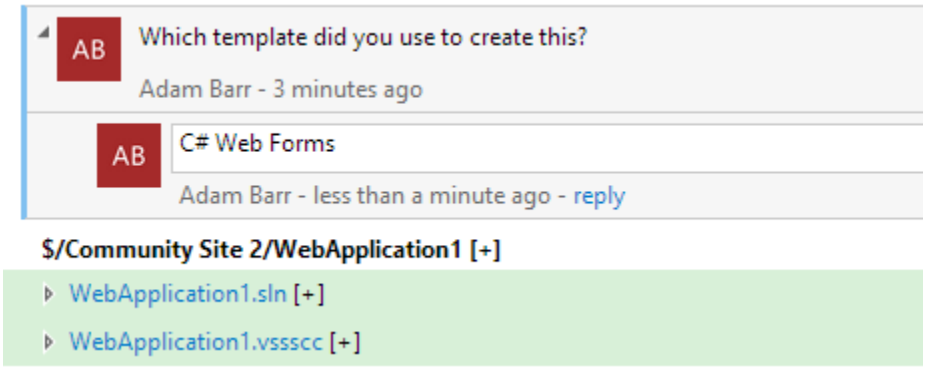


Note: Although we will not go through the steps in this lab, it is also possible to comment on specific files and specific lines within files for the changeset.

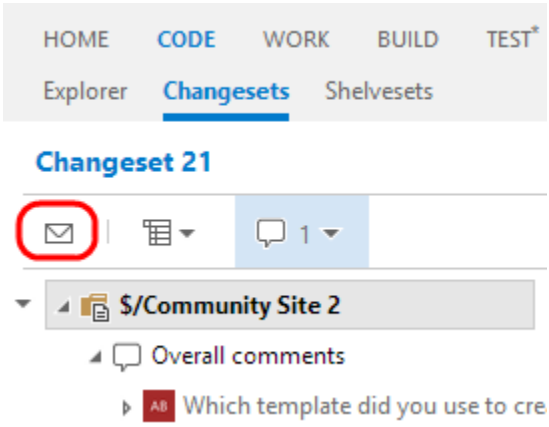
7. Team members can reply to comments and start a threaded conversation. Select the **Reply** link next to the comment that you just left.



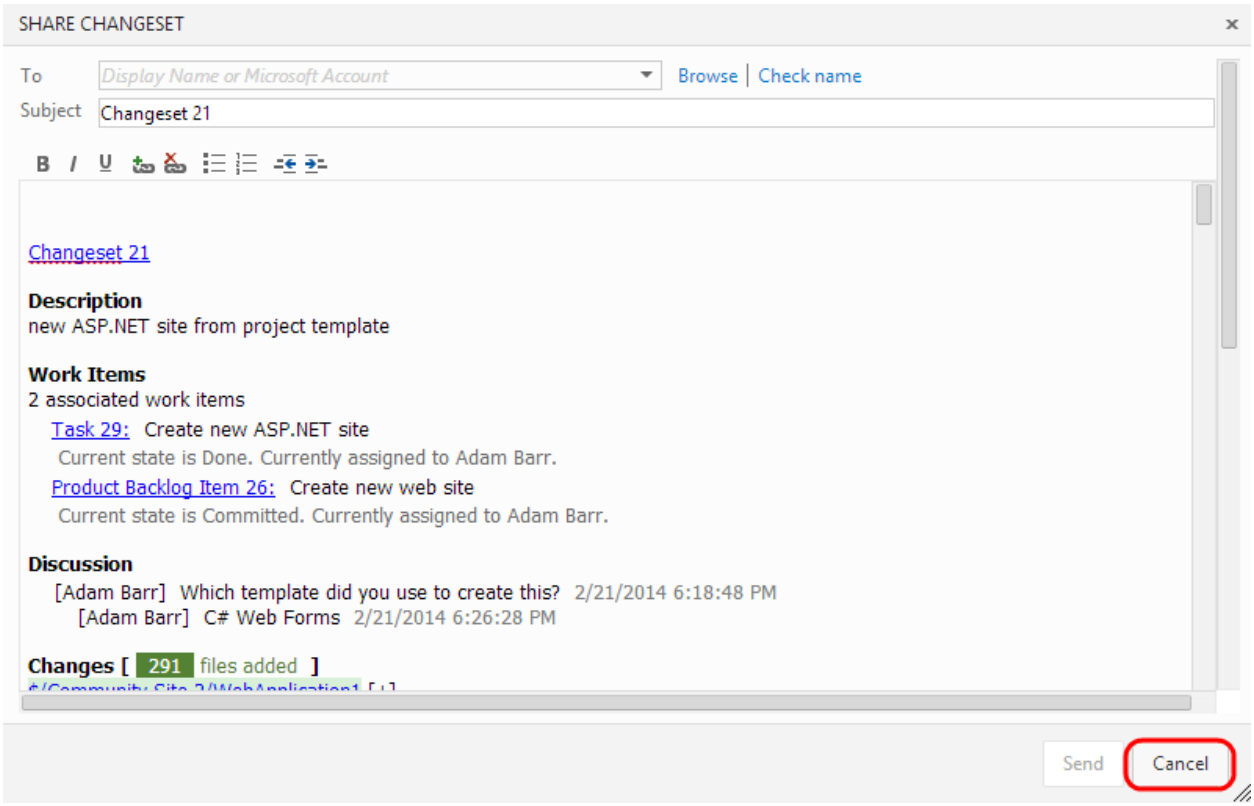
8. Enter a comment and then press **Enter**.



9. All team members can view the code comments using just a browser. To notify another team member about a code comment, you can send mail with a link to the changeset. Select the **Share Changeset** button.



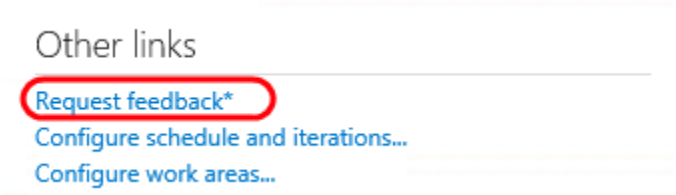
10. Note the description, associated work items, and discussion are sent as part of the mail. Once you are done viewing what would be sent to the recipient, click the **Cancel** button to discard the email.



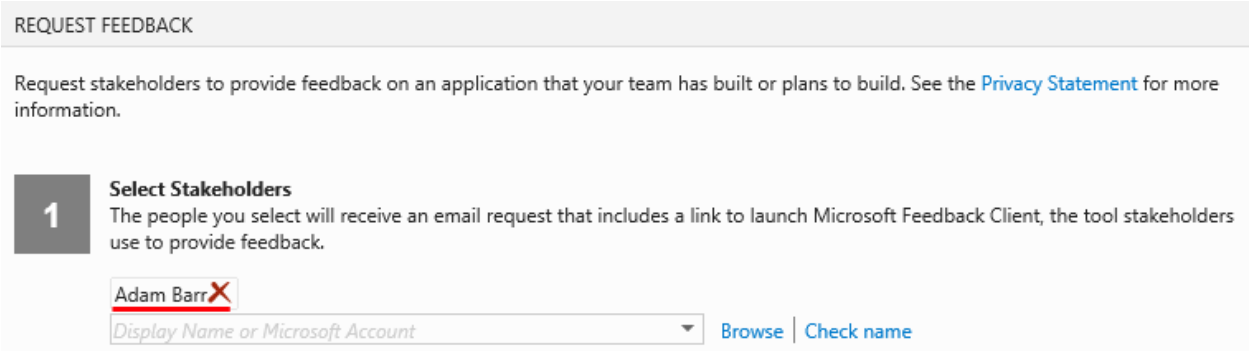
Task 3: Requesting Feedback

You can also request feedback from project stakeholders and have that information recorded in the project.

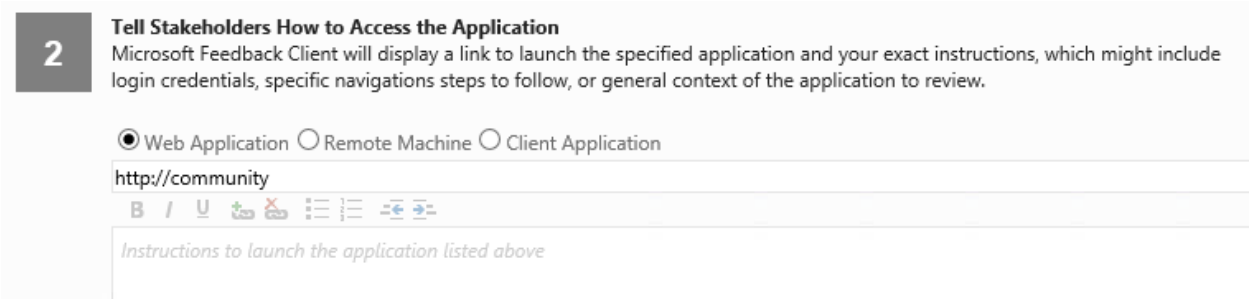
1. Navigate to the Community Site project overview page and then select the **Request Feedback** link.



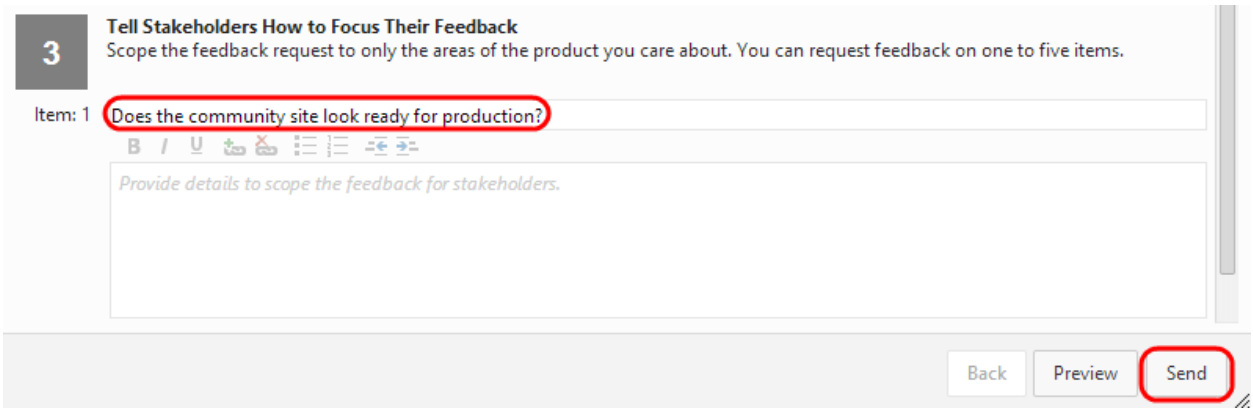
2. In the Request Feedback window, select yourself for demonstration purposes. Note that you could add additional stakeholders to request feedback from.



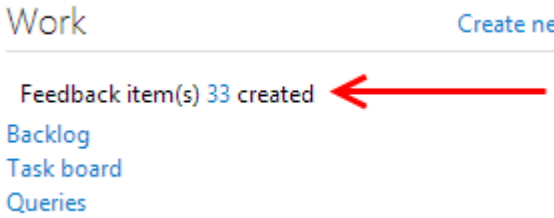
3. For step **#2**, use the default option to specify that this is a **Web Application**. Enter a fake URL such as http://community to represent an intranet test site. You could add in additional instructions if needed.



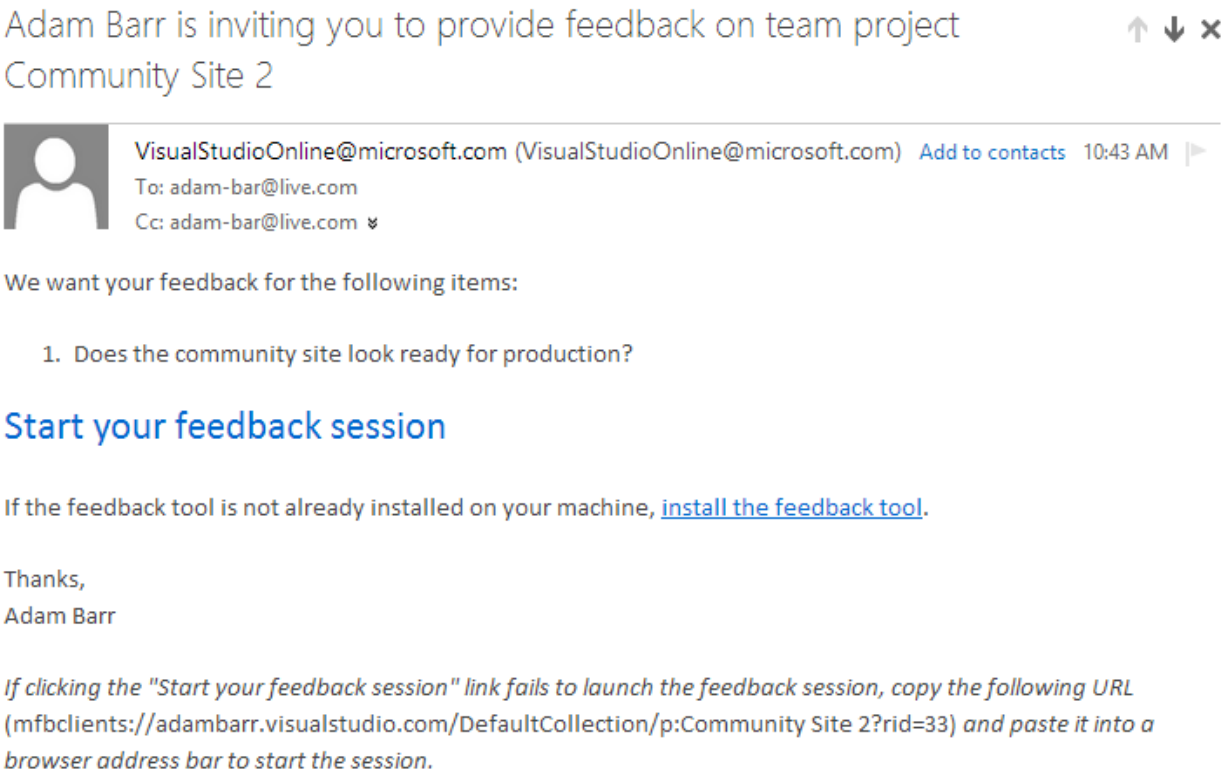
4. For step **#3**, enter a single feedback item of "**Does the community site look ready for production?**" Click the **Send** button.



5. After the feedback request is sent, you should see a note in the Work section stating that a feedback item was created. Your work item number may be different.

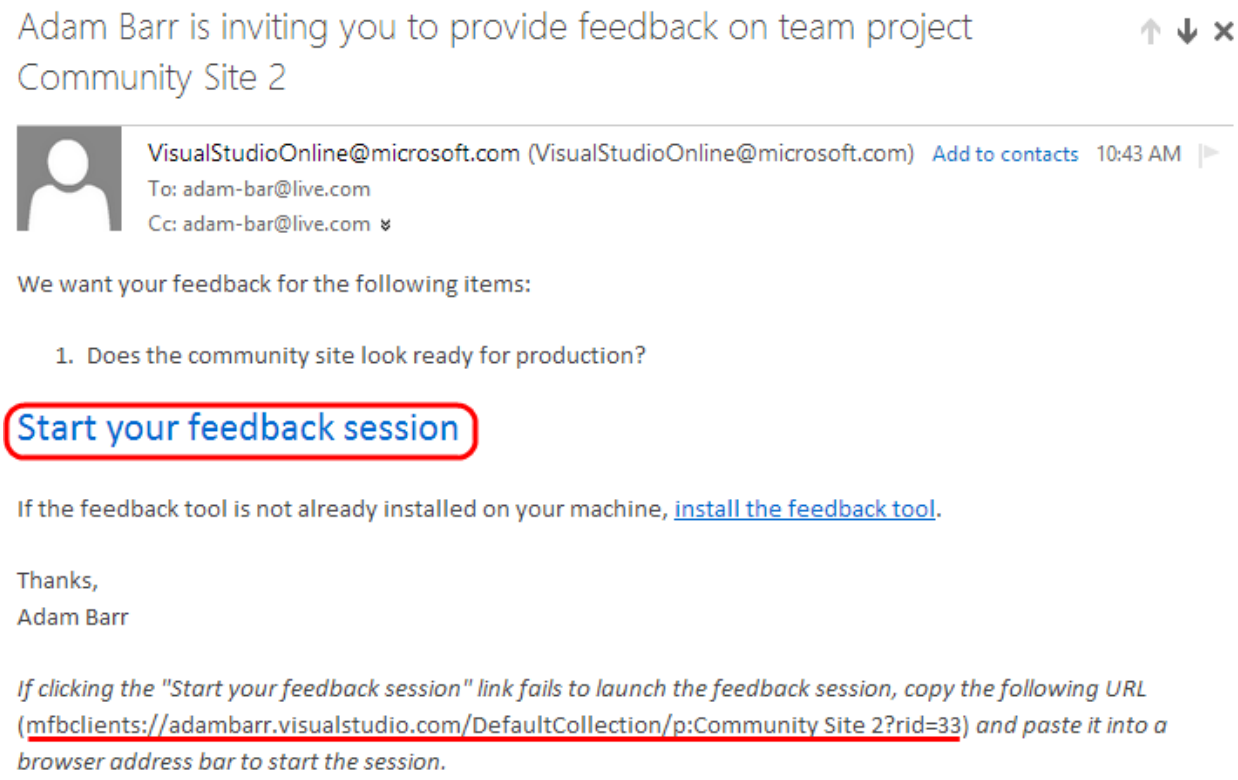


6. Open and view your mail sent from Visual Studio Online.

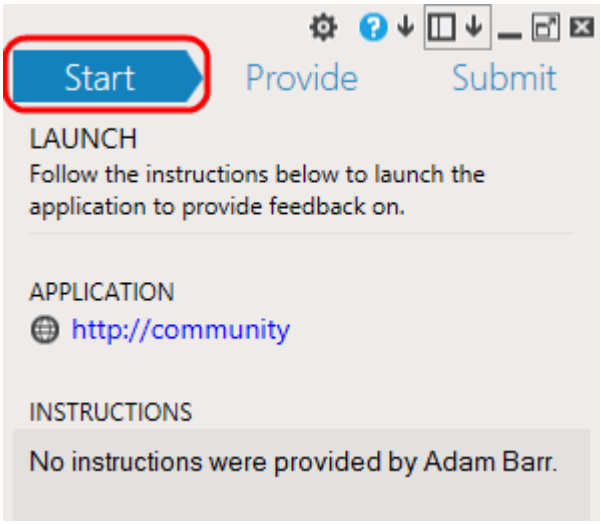


7. Select the "**Start your feedback session**" link. If that does not work, copy and paste the URL at the bottom of the mail into a browser address bar to start the feedback session.

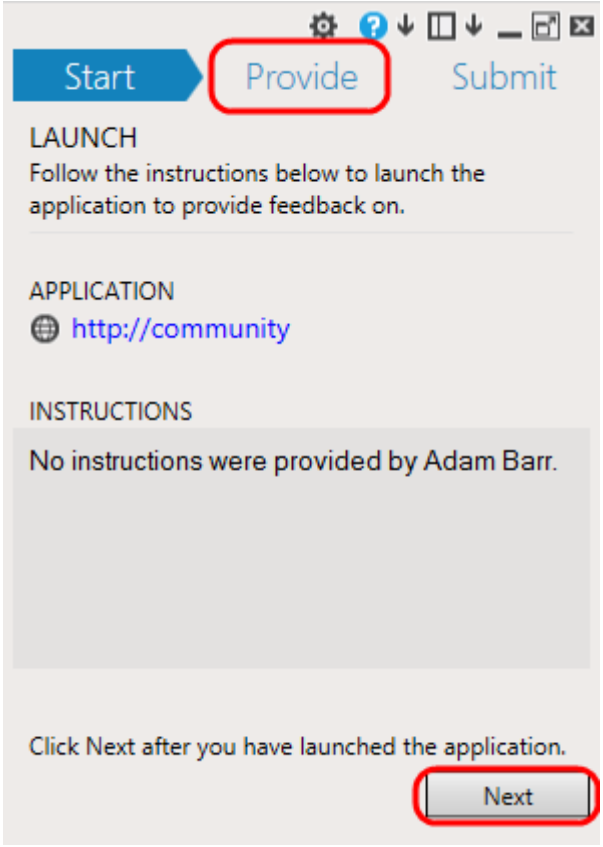
   Note: the mail also contains a link to install the feedback tool in case the recipient does not already have it installed.



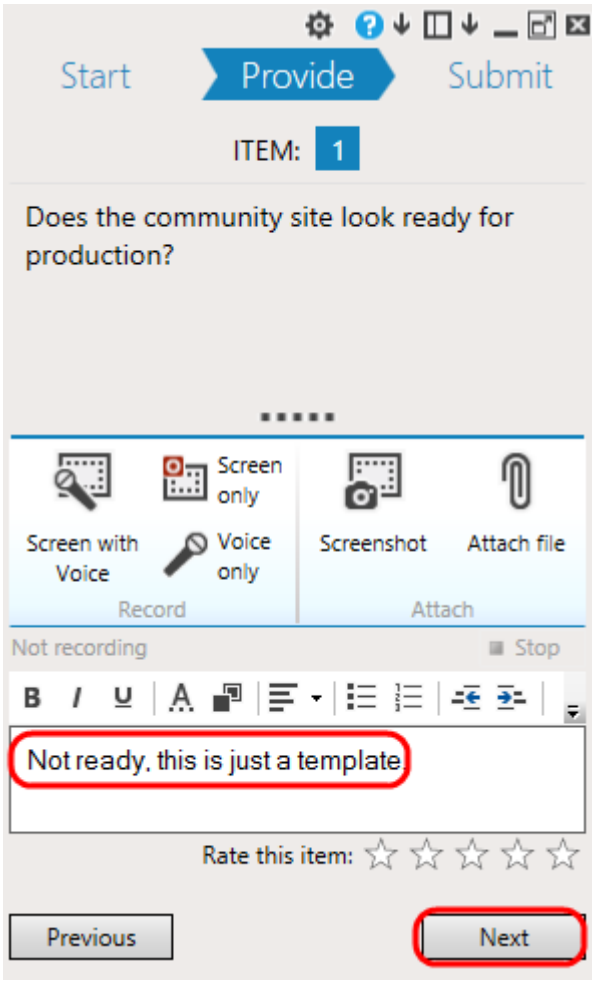
8. The feedback tool window should now be open and currently set to the **Start** tab.



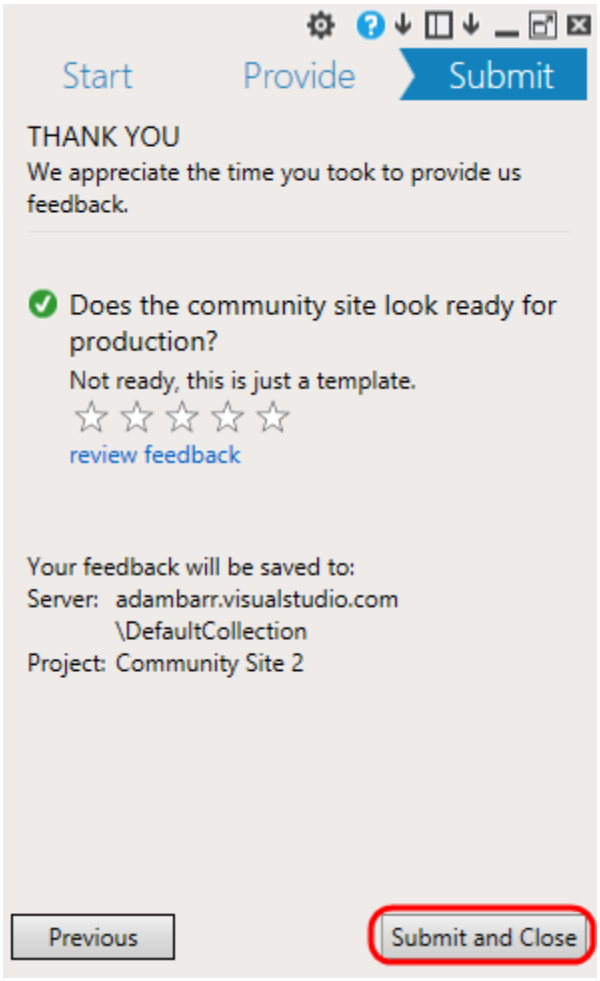
9. The link to the fictional application site is shown, which is what you would click on to launch a browser window to the site if it really existed. Let's go ahead and provide some feedback by selecting either the **Provide** tab or the **Next** button at the bottom.



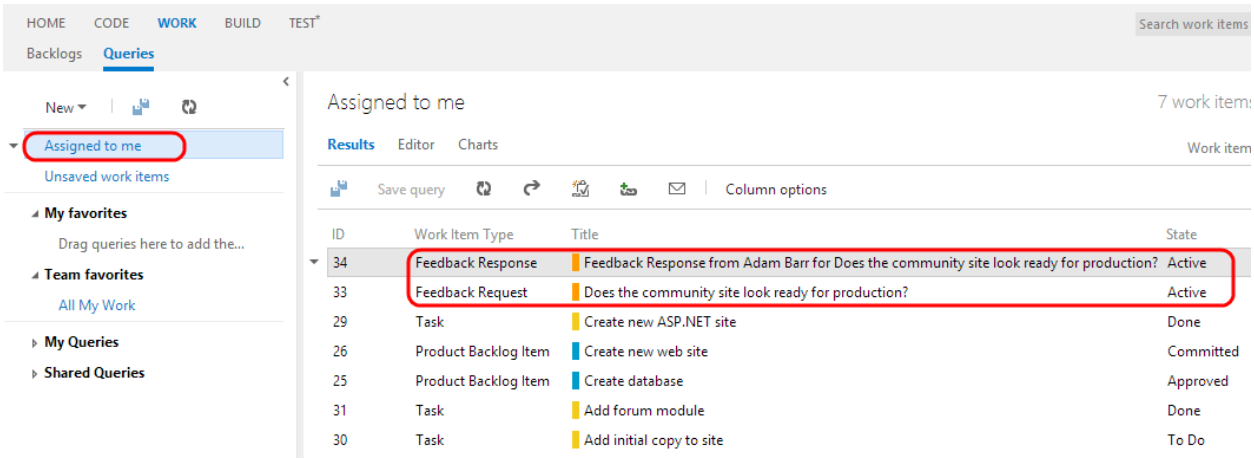
10. The **Provide** tab allows the reviewer to provide the requested feedback. Note that there are a number of different attachments that can be included in the feedback including screen and voice recording, screenshot, and other miscellaneous files. For this purposes of this lab, however, just add a comment of "**Not ready, this is just a template**" and then click the **Next** button.



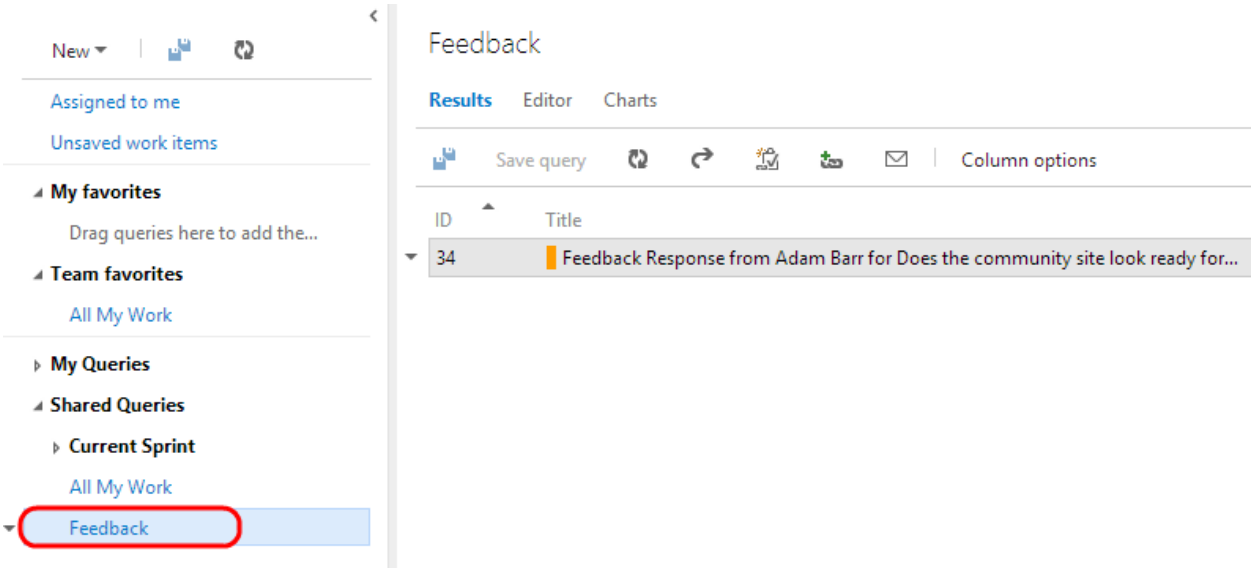
11. Select the **Submit and Close** button.



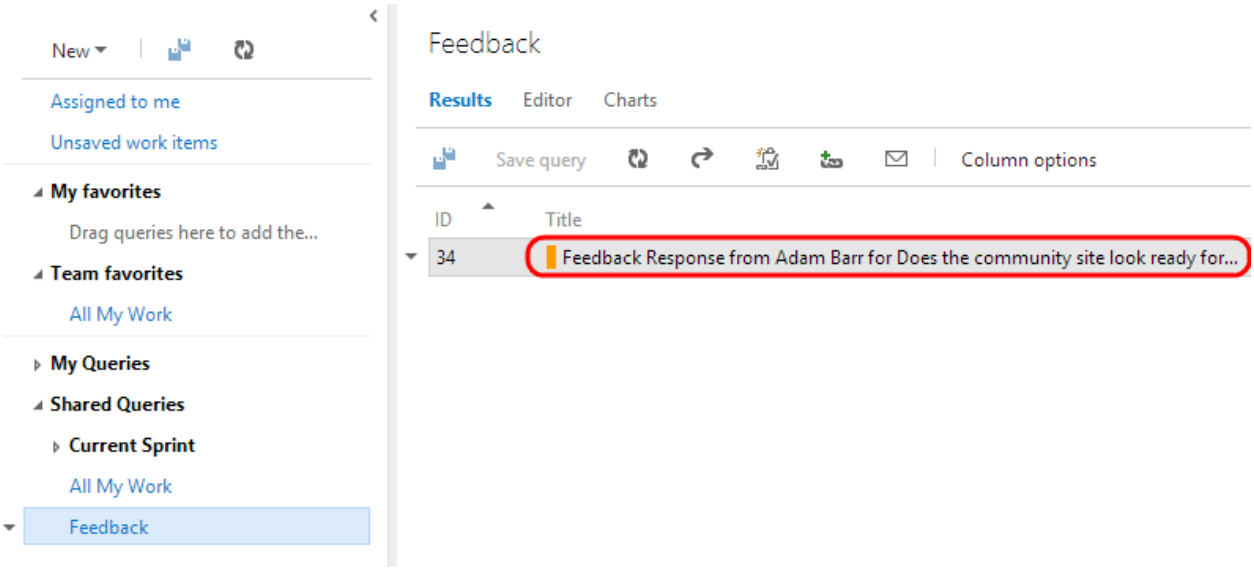
12. Navigate to the **Work | Queries** page for your project in Visual Studio Online. By default, you should now be looking at the "**Assigned to me**" query, which shows two feedback related work items.



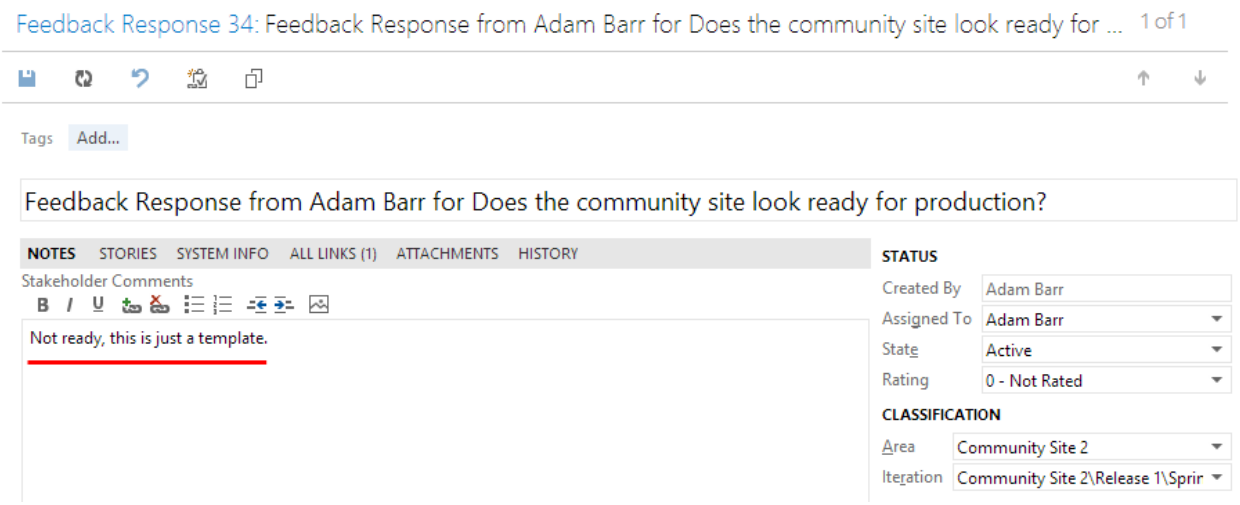
13. You can also check for just feedback responses by running the "**Feedback**" query from the **Shared Queries** group.



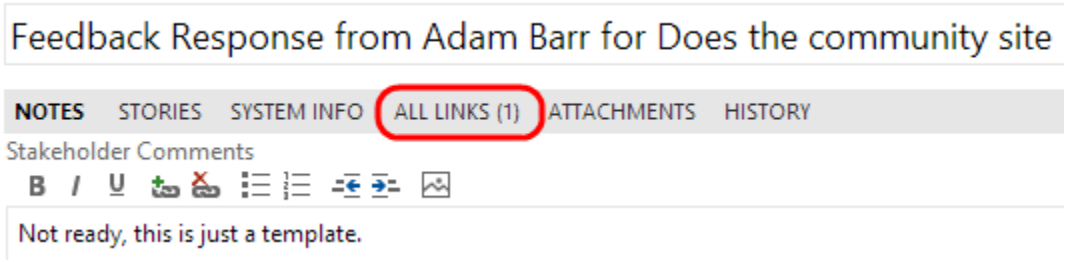
14. Double-click on the feedback response.



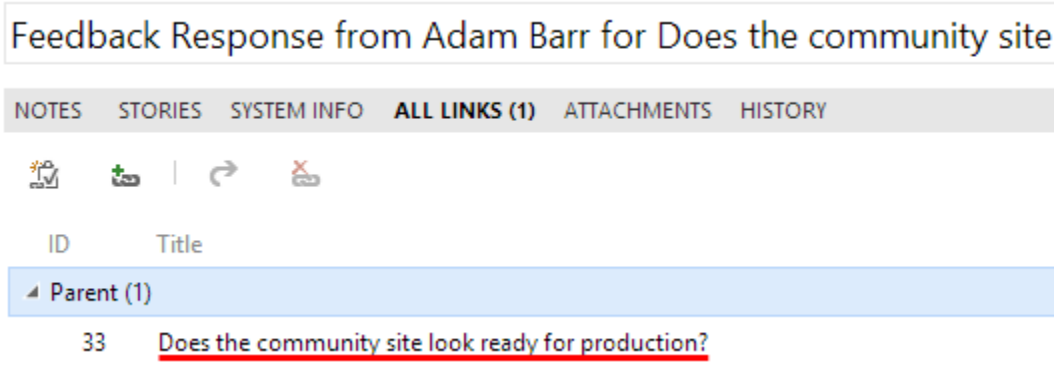
15. Here you can see the feedback notes.



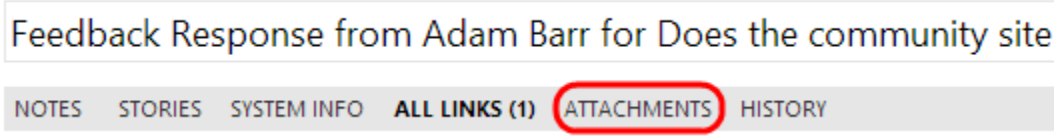
16. Select the **All Links** tab.



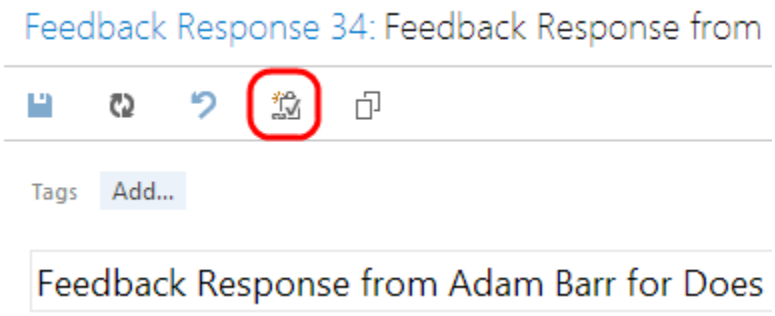
17. When feedback is sent back from stakeholders, it is automatically linked to the original request.



18. Select the **Attachments** tab. If the reviewer had created screenshots, video recordings, or other attachments they would show up here.



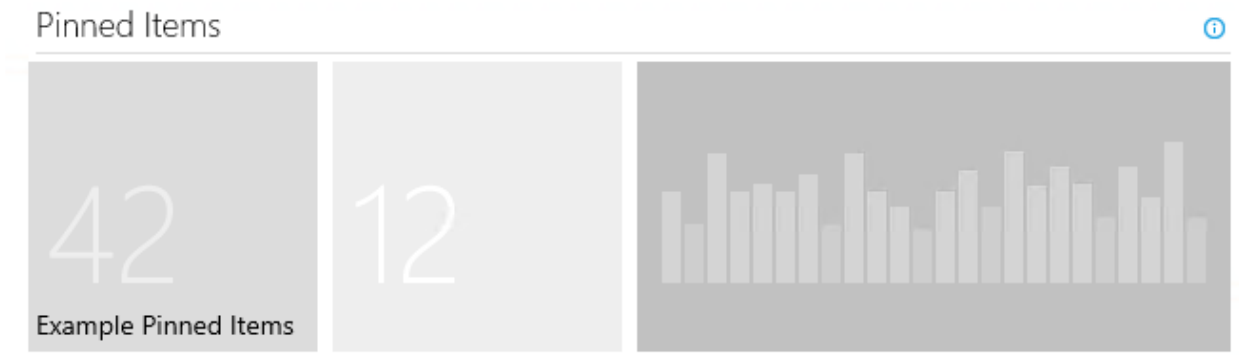
19. If it is determined that the feedback response requires some additional work by the team, you could click the **New linked work item** button to create a new task, bug, or other work item that is a child of this feedback.



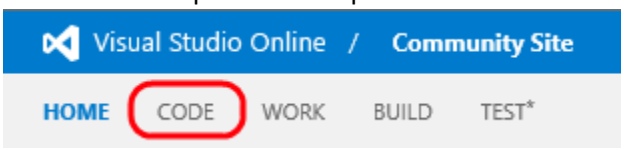
Task 4: Pinning Tiles to the Project Overview Page

There are a few different project metrics, links, and visualizations that can be customized and pinned to the project overview page.

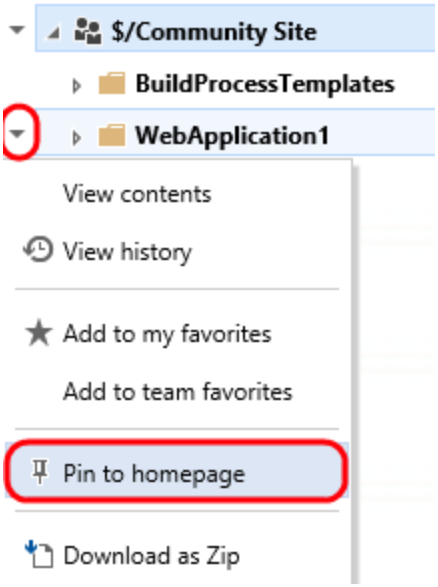
1. Navigate to the Community Site project overview page to take a look at the current state for the **Pinned Items** section. The tiles that are shown here at the moment are simply placeholders and do not represent real data.



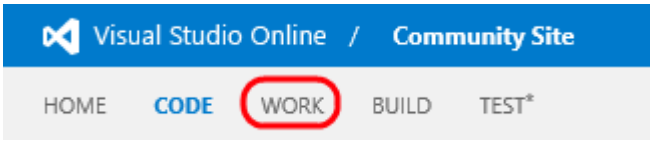
2. Version control paths can be pinned. Select the **Code** link to navigate to the code explorer view.



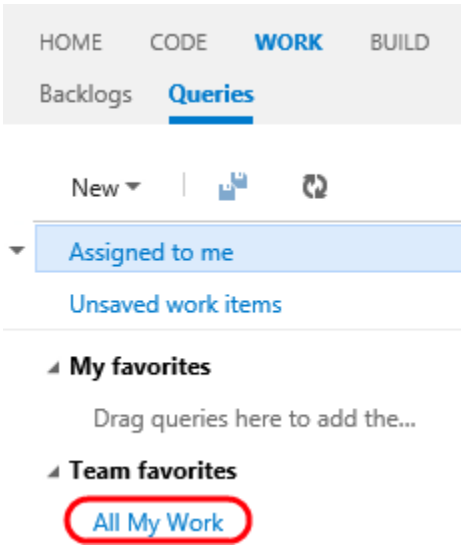
3. Select the **down arrow** that appears to the left of the **WebApplication1** node and then select the "**Pin to homepage**" option.



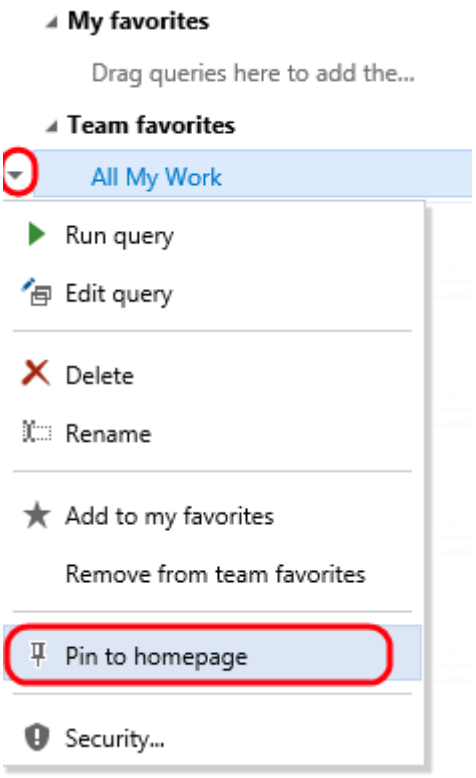
4. Work item queries can also be pinned. Select the **Work** link to navigate to the Queries view.



5. Select the "**All My Work**" query link from the **Team Favorites** section that you created earlier in the lab.

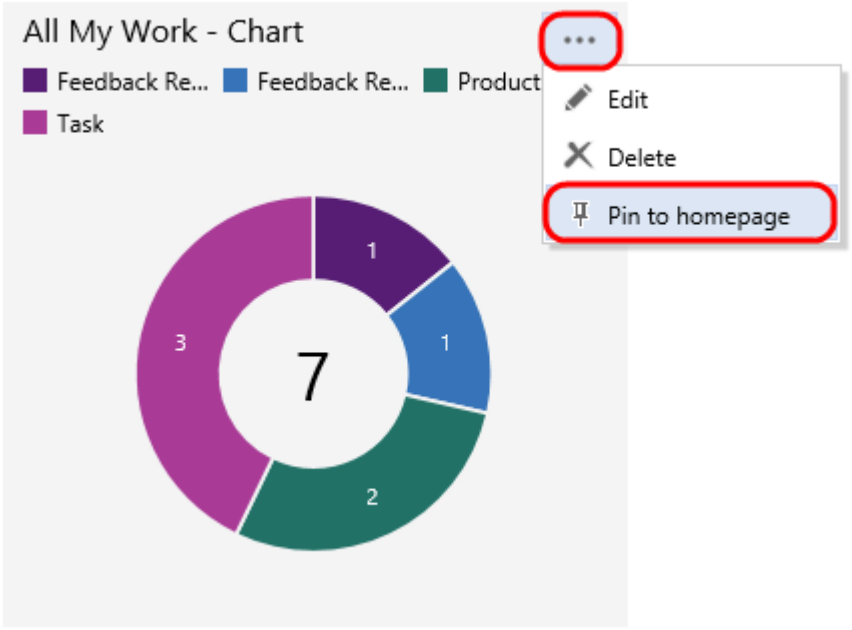


6. Select the **down arrow** to the left of the "**All My Work**" query link and then select the "**Pin to homepage**" option.



7. While still viewing the "All My Work" query, select the **Charts** link.

8. With the mouse cursor placed over the top-right corner of the chart that you created earlier in the lab, click the **ellipses** that appear and then select the "**Pin to homepage**" option.



9. Although we will not do so in this lab, it is also possible to pin specific builds.
10. Return to the project overview page and admire your new pinned tiles. This gives the team an at-a-glance view into selected aspects of the project and also provides a direct link to the pinned source control path, the pinned work item query, and so on.

Sprint 1

April 15 - April 21

5 of 25 h

Backlog items: 1 not started, 1 in progress

Pinned Items

WebApplication1

1

Recent changes

All My Work

7

Work items

All My Work - Chart

Feedback Re... Feedback Re... Product Bac... Task