**Table of Contents**

[**Lab: Module 3 – VSTS Testing 2**](#_Toc492158295)

[Duration: 2 hours (10-30 mins per exercise) 2](#_Toc492158296)

[Prerequisite 2](#_Toc492158297)

[Exercise 1: Setup VSTS Account and Projects (skip if already setup) 2](#_Toc492158298)

Exercise 2: Manual Testing: Test Plans, Suites, Adding Tests, Tagging 9

[Exercise 3: Manual Testing: Microsoft Test Manager (MTM) Web Browser Based 18](#_Toc492158299)

[Exercise 4: Manual Testing: Configuration, Templates, Parameters, Sharing Steps, Re-ordering, Filtering 23](#_Toc492158300)

[Exercise 5: Manual Testing: Excel / Grid View 37](#_Toc492158301)

Exercise 6: Manual Testing: Charts, Test Summaries and adding Widgets to the Dashboard 46

Exercise 7: Automated Testing with Coded UI Testing: Overview (VS Enterprise only) 49

[Exercise 8: Load Testing: Overview (VS Enterprise only) 53](#_Toc492158302)

Final Notes: What's New in Testing? What's on the Roadmap? 67

Lab: Module 3 – VSTS Testing

**Duration: 2 hours**

**Prerequisite**

You must have a VSTS account. You can create a free one using a personal or work email by going here: <https://www.visualstudio.com/team-services/> and clicking the blue “Get Started for Free” button. Everyone will use their own personal VSTS account. And, you need an Internet connection.

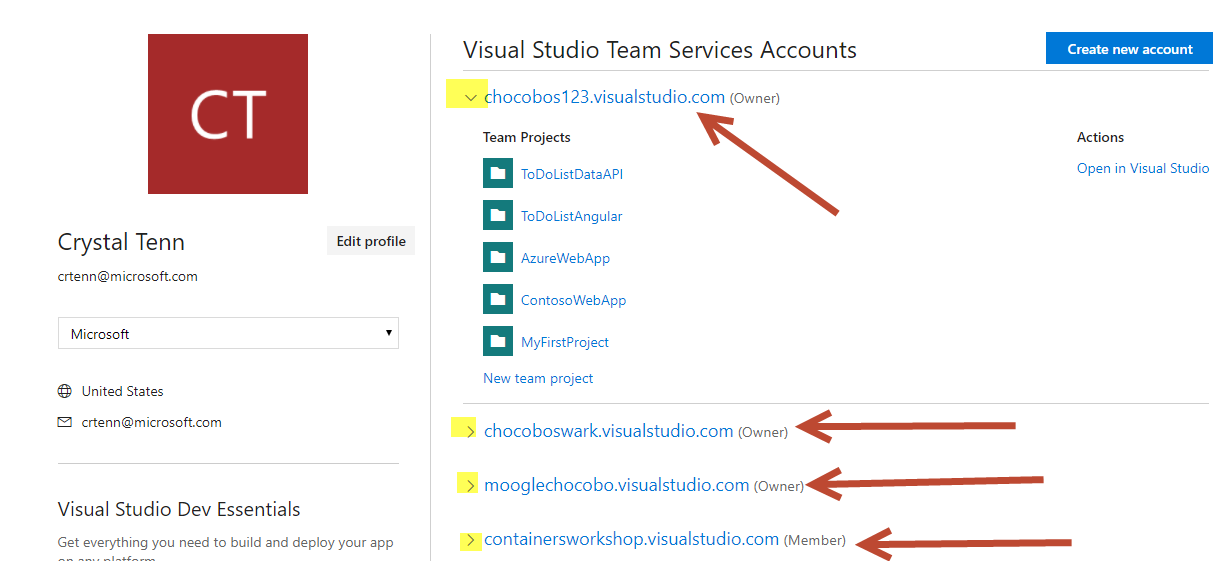
**Exercise 1: Setup Your VSTS Account and a Projects (20 mins)**

If this is the first lab you are completing in the workshop, then you will need to create a VSTS account and team to complete the lab and work in the Test tab of a team. We are following a scenario with a restaurant business called Pho Sizzle Vietnamese Cuisine and you will create a VSTS account to help their IT department manage their workload!

If you have completed another lab in this workshop, you should already have a VSTS team account setup. If this is the case, please skip exercise 1 and move on to exercise 2. Please login at <https://www.visualstudio.com/team-services/> and you should see your project.

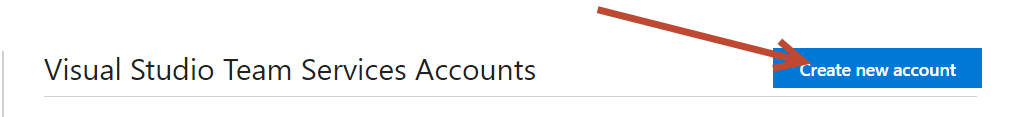
**Tasks**

1. **Setup VSTS**
2. Go to this URL: <https://www.visualstudio.com/team-services/>
3. Click on “Get Started for Free”. If you do not have an account, please Create one now. You can use a work email or a personal Microsoft account. If you have an account already setup (or an MSDN) please login.
   * 1. *Note: VSTS and TFS are basically the same, except VSTS is constantly updated and kept current and to get new changes you need to manually update your TFS. And, TFS on premises and VSTS is just accessed through the web. If you do not see any of the features I talk about here in your TFS, most likely it is in a newer update of TFS.*
4. You will land on a page that looks like this (except no projects like shown in my screenshot with the arrows and highlights!).

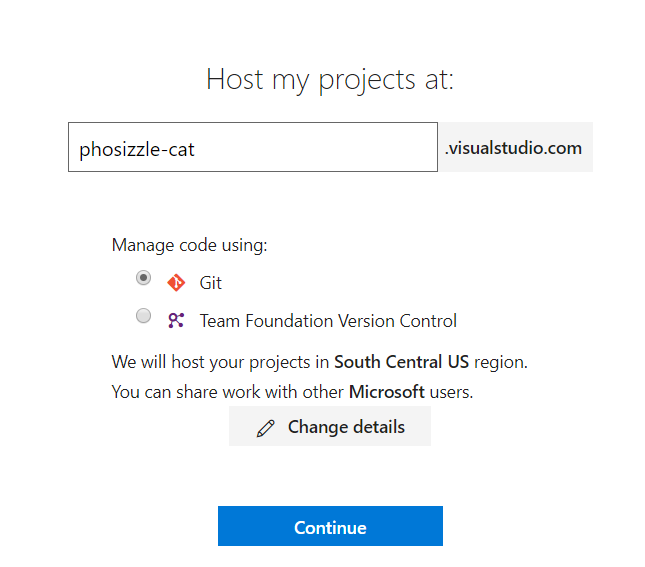


*Note: Each account typically contains multiple team projects. If you have multiple unrelated dev teams, it would be good to make a separate VSTS account for each. For example, let’s say you owned a restaurant business. If you had 3 teams of developers, who worked on 3 separate types of projects: customer-facing restaurant websites, restaurant applications for the employees to manage orders, and an eCommerce online ordering platform. Each of these 3 teams should have their own VSTS page. Also, if different teams have different processes and project settings, they would need to be split up. While you could put all the applications on the same VSTS, this is not the best idea as every person on every team has access to everything even though they shouldn’t. It is also harder to find However, you may want to place all the projects in one VSTS account if it is a small company and all the team members touch all the different applications.*

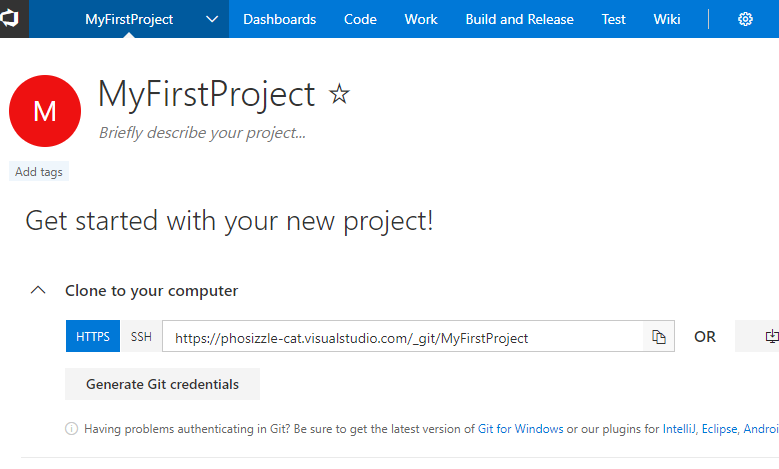
1. Hit Create New Account on the top right (blue button) to make your own VSTS account.



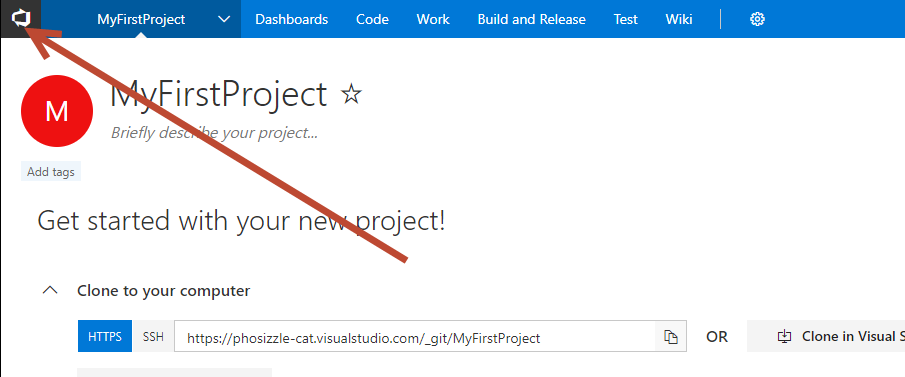
1. The name has to be unique across all existing VSTS accounts. You can name it “PhoSizzle-YOUR-INITIALS” for example. Mine here is called “phosizzle-cat”. Leave the code management as Git. Hit Continue and give it a few minutes to create your acount.



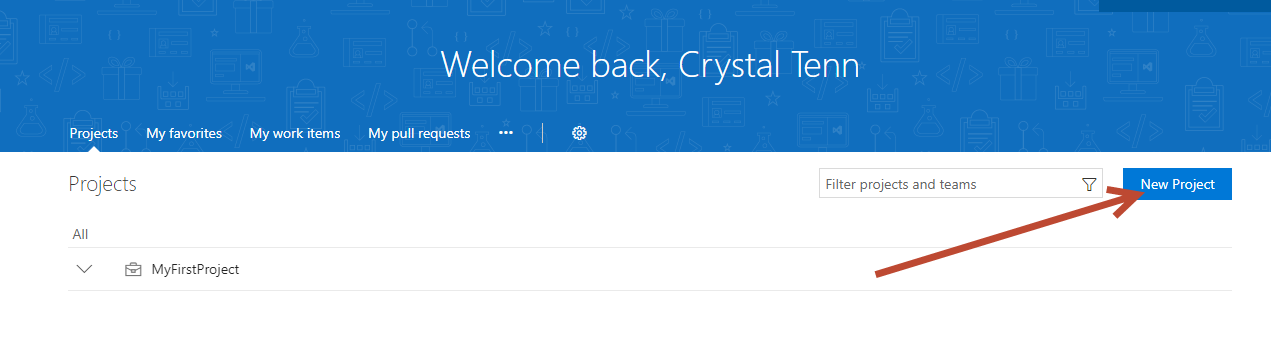
1. You will land on a page like this:



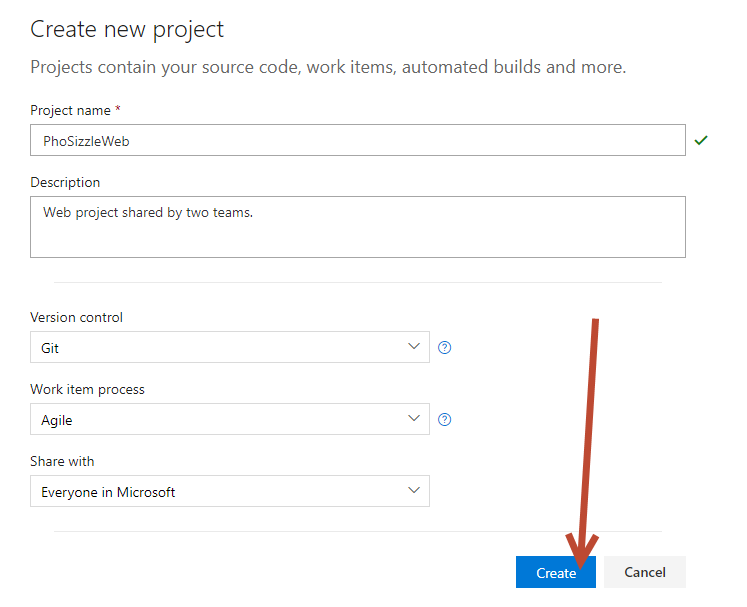
1. Click on the main VSTS logo on the top right.



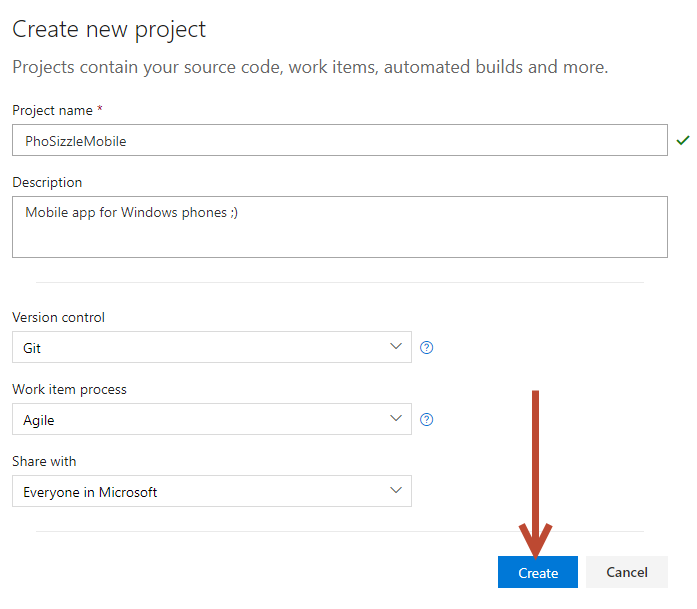
1. You will land on a page like below, click on New Project.



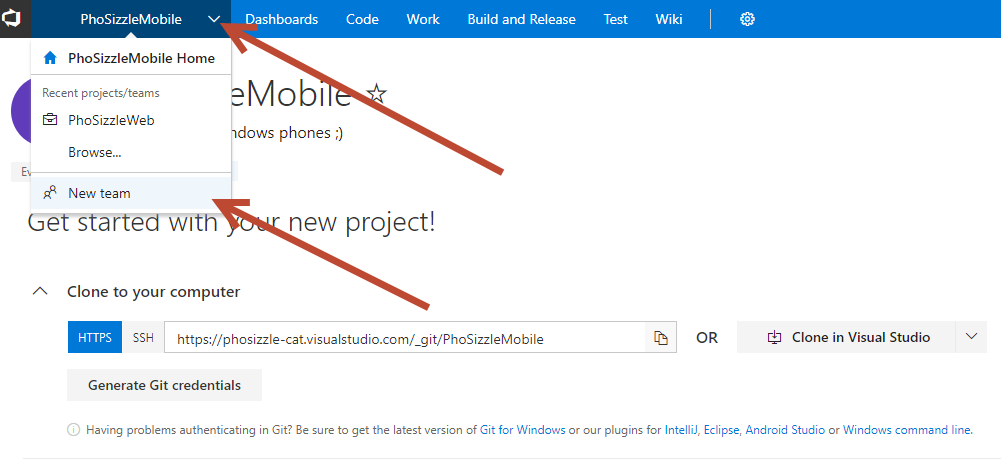
1. I’ll keep it simple and call one PhoSizzleWeb and hit Create (put anything you like in the description):



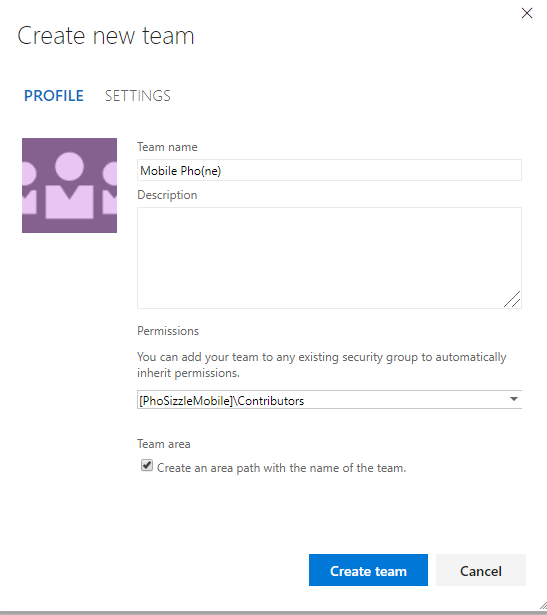
1. When you are done, Click the VSTS logo again. Then, click New Project again to create the Mobile project (put anything you like in the description) and hit Create:



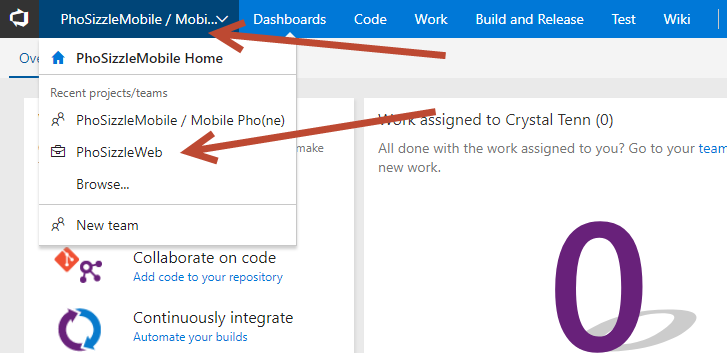
1. Let’s stay on this Mobile project for now. We need to create a team. Click on the name of the PhoSizzleMobile and click New Team.



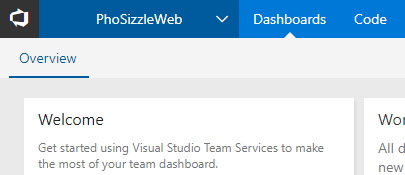
1. Add your Mobile Pho(ne) team. Permissions will stay as Contributor. And you can leave the box checked to create an area path. Hit Create Team when you are done.



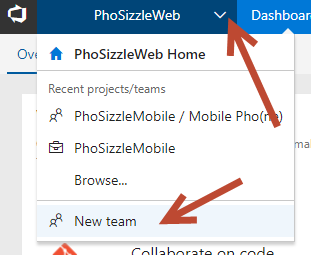
1. Click on the PhoSizzleMobile button on the top to swap to the other project for the PhoSizzleWeb:



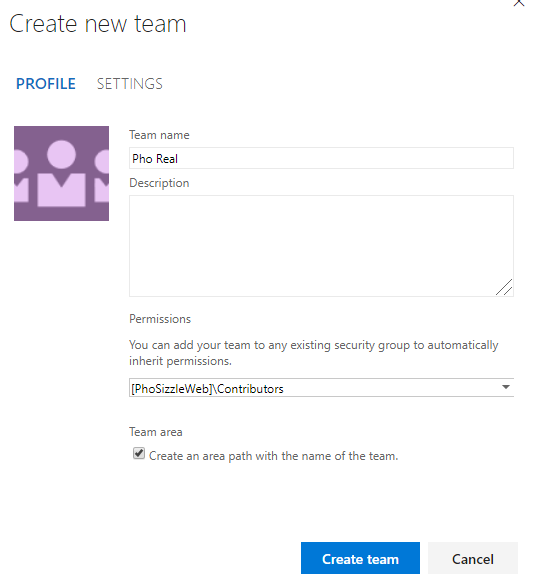
1. You should see PhoSizzle web when you are done swapping over:



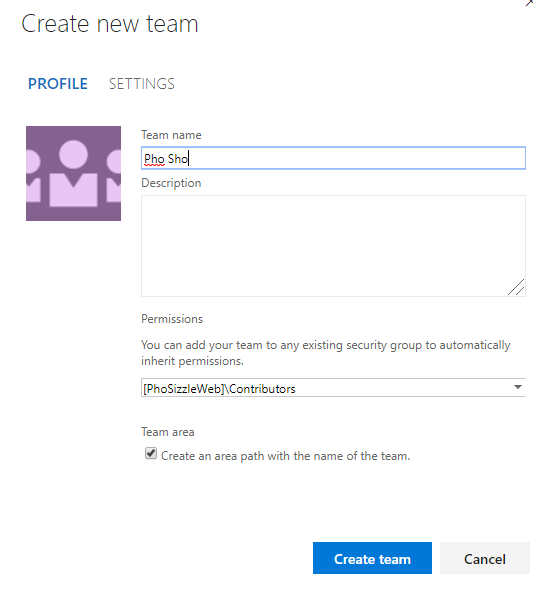
1. Click the dropdown again, and hit New Team:



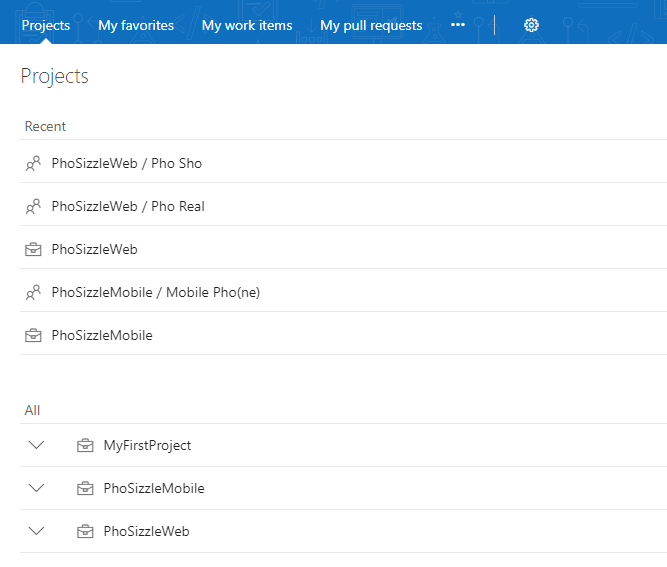
1. Create one Team for Pho Real.



1. Create one Team for Pho Sho.



1. Click on the VSTS logo and you should see the following as your Projects!



**Exercise 2: Manual Testing: Test Plans, Suites, Adding Tests, Tagging, and Filtering (15 mins)**

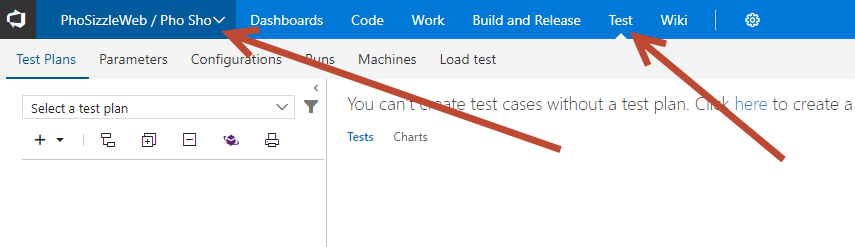
The Pho Sho QA Testers are ready to get started on making some tests to test out that the menu works properly for the restaurant. Let’s help them setup some test plans, suites, and tests! They will also be making a smoke test and regression tests suites (we will just get them started on it! We don’t have to fill in all the tests now!). We will also teach them how to use tags and how to filter their tests, so they can easily find their work.

Here is an overview of basic testing / VSTS testing / TFS testing info below. If you are advanced you can skip this section and start working on the lab exercise.

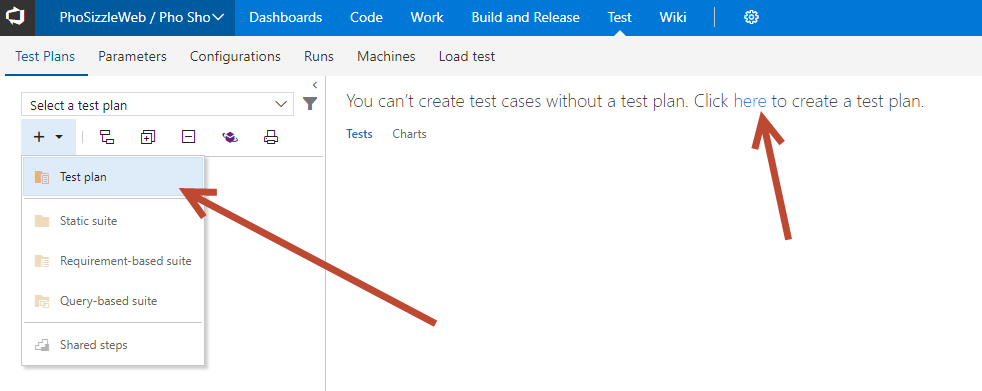
* Create test plans to track manual testing for sprints or milestones. That way, you can see when the testing for a specific sprint or milestone is complete.
* **Test plans** are used to group together test suites and individual test cases. This includes static test suites, requirement-based suites, and query-based suites. You can add individual test cases to a test plan without creating a test suite if you wish, but using a test suite provides a way to group test cases for separate testing scenarios within a single test plan.
* At the start of a sprint, find out what you need to test. Discussing test cases is a great way to help the team understand the detail of what your users need. Tests planned in this way provide a clear target for the development team.
* **Test suites** are folders inside a test plan that help you organize tests. When you run tests, you can choose to run all the tests in a suite, one after another. These operations don’t affect the test case itself. Suites contain links to test cases, and it’s the links that you’re moving or copying. For the same reason, removing a test case from a suite doesn’t delete the test case from VSYS.
* Test suites A can contain overlapping tests with another test case, Test suite B, for example. They are not exclusive. You can have a test that belongs in each suite because it is very important and in a very common testable area. You could also have a test case that is only in one suite, or no suites.

**Tasks**

1. **Setup a Test Plan, Test Suite, and Test Case**
2. Go to the Pho Sho Team and their Test hub (sometimes the Test tab of VSTS is called Test hub). Make sure you see PhoSizzleWeb / Pho Sho on the top left navigation bar.



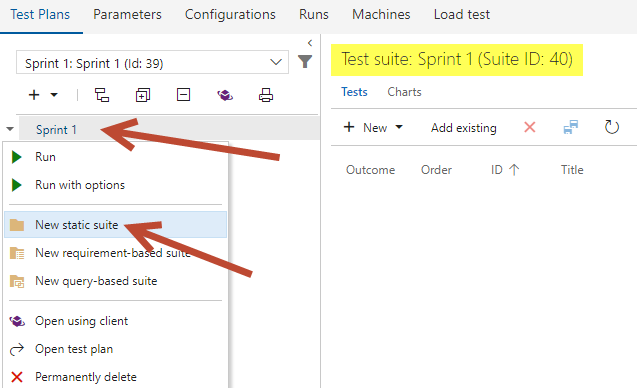
1. Let’s create Test Plan. You can either click the ‘+’ sign on the left side and select Test plan, or if it’s your first Test plan there will be a message when you land in the Test hub.



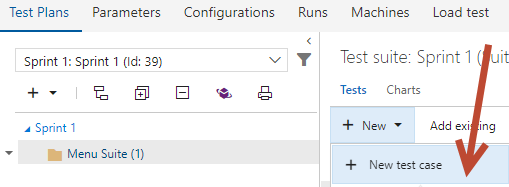
1. Name your Test Plan ‘Sprint 1’ and choose the Area path to be Pho Sho. If you have a sprint setup you can choose it for the Iteration. Click on the Create button.



1. Right click on Sprint 1 on the left, and choose to add a New Static suite:



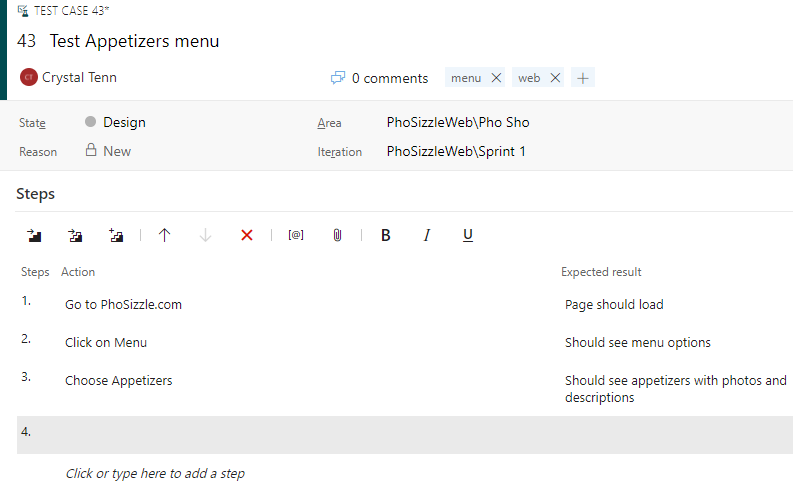
1. Add a new Test case



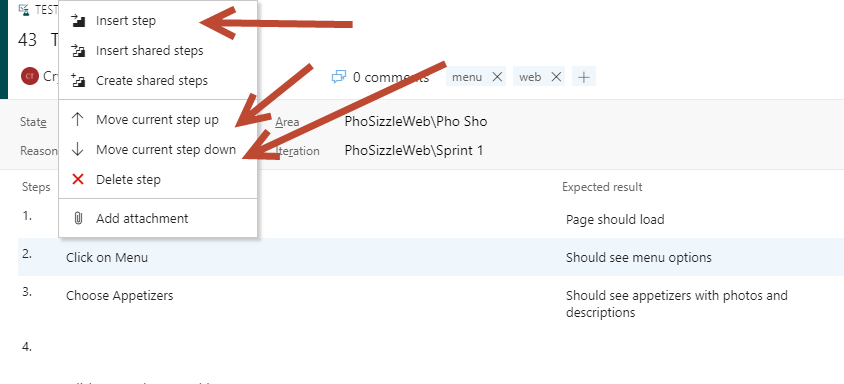
1. Add a name for the Test Case: ‘Test Appetizers menu’ and add two tags ‘menu’ and ‘web’ by clicking the “+” sign. Keep the state as Design, since you are currently designing the test case.



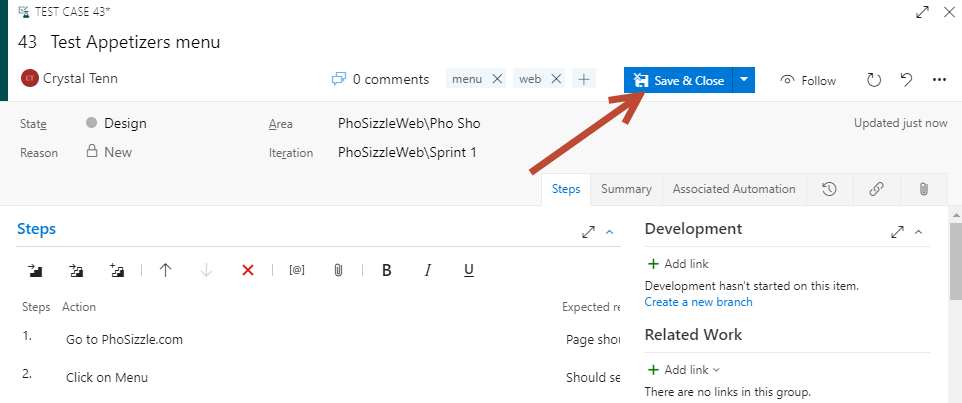
1. Add the following Actions and Expected Results. If you are in the Action, when you finish typing you can hit ‘Tab’ to go straight to Expected Result. When you are in the Expected Result, and when you are done, you can hit Enter to go straight to the next step.



1. If you need to re-order a test step, or insert a test step in between existing steps—Right click on a step and select one of the following options:



1. Click Save and Close



**Exercise 3: Manual Testing: Microsoft Test Manager (MTM) Web Browser Based (25 mins)**

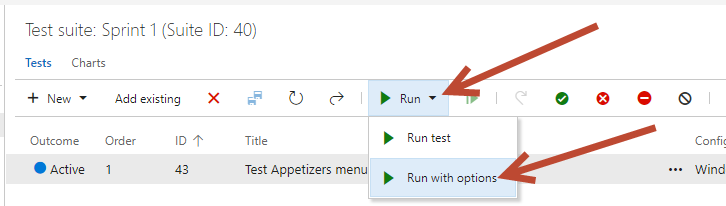
The Pho Sho QA Testers are ready to get started with testing their brand-new Appetizer test! However, they don’t have Microsoft Test Manager installed or any version of Professional or Enterprise Visual Studio (hint! Those are requirements to get Microsoft Test Manager automatically on your computer!). Luckily for us, there is a web based Microsoft Test Manager.

With this version of Microsoft Test Manager, you do need to use the **Chrome browser** in order to utilize all of its features successfully.

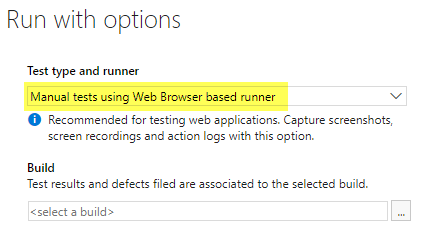
We are going to be able to run tests, take screenshots, take video captures of tests, and create bugs all while testing!

**Tasks**

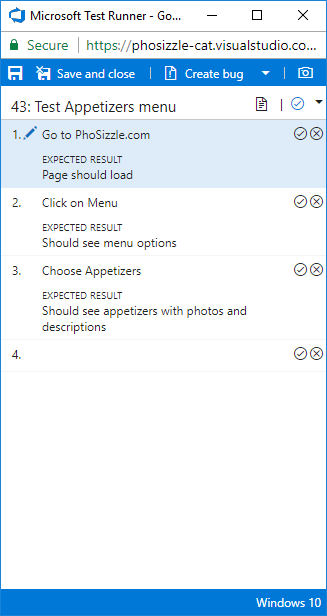
1. Run a test
2. Click the Run button, it will dropdown two options, choose Run with Options.



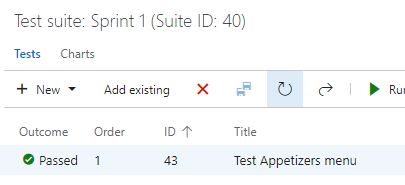
1. Make sure you select “Manual tests using Web Browser based runner”. Also make sure you are using Chrome as your browser. Then, hit OK.



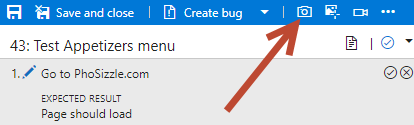
1. You will see a window like this pop-up. If it does not, make sure you do not have an Ad Blocker.



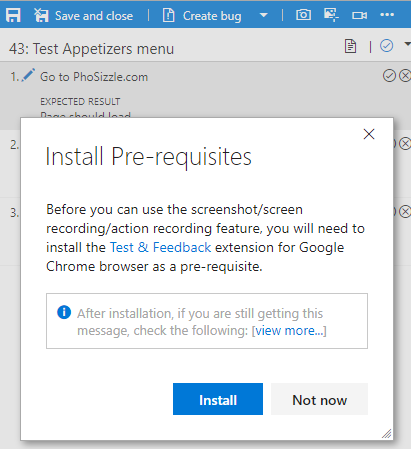
1. Click the check mark to pass a step, or click an “X” mark to fail a step. Click on the check marks one step at a time to pass all the steps as passed. Then click Save and Close at the top to end the testing session.
2. You should see the outcome of your test as Passed on your testing page.



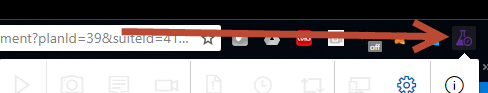
1. Run a test with Screenshots
2. Click on the Run the test option, you should be able to the normal Run Test option without options.
3. When the pop-up shows up, click on the camera icon.



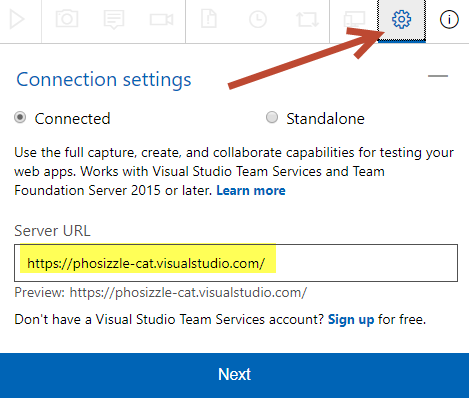
1. Hit Install if it prompts you like below. You will need to add the extension to chrome. You will need to relaunch the test runner after you are done installing it.



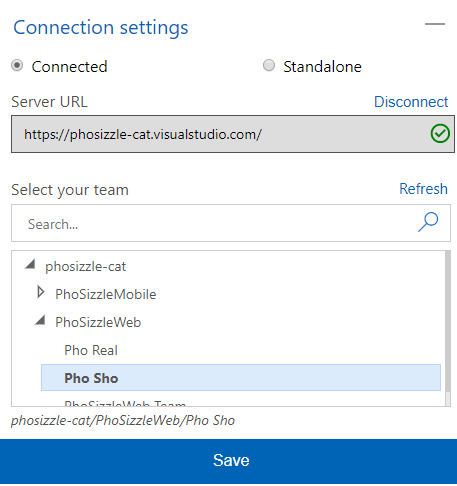
1. Click on the purple Test icon in Chrome:



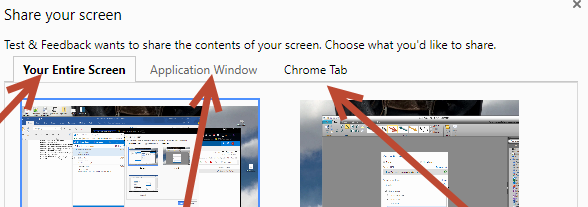
1. Click on the Settings Gear. Add your VSTS url into the Server url input box. Hit next.



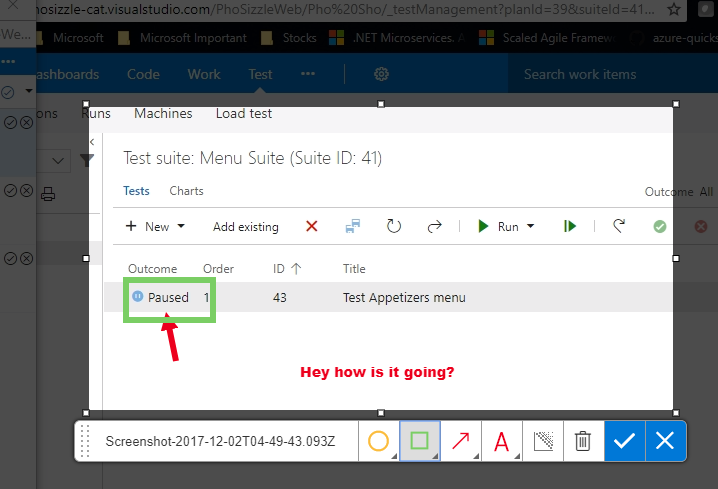
1. Select the correct PhoSizzleWeb project and Pho Sho team. Hit Save.



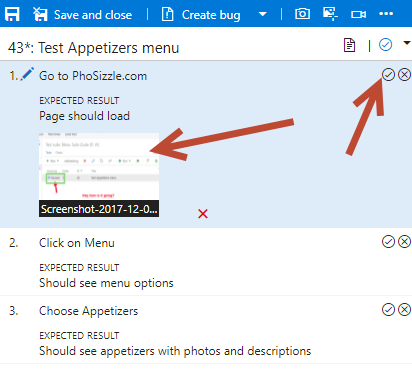
1. You can choose any of the following to take a screenshot from. Choose “Your Entire Screen”



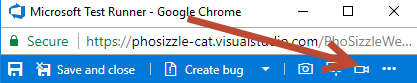
1. It will allow you to edit the screenshot and crop in what you want to see like this (try using the different options and adding text / circles / squares!). Hit the blue checkmark on the right when you are done.



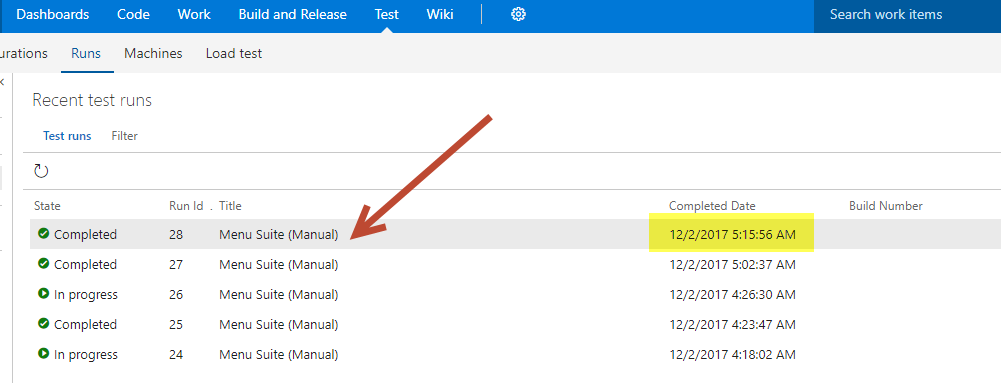
1. You will see the screenshot show up in the step. Now you can mark the step as passed or failed. Typically, you will want screenshots if you need to show steps for reproduction, or if a step fails. Pass the other steps, and hit Save and Close.



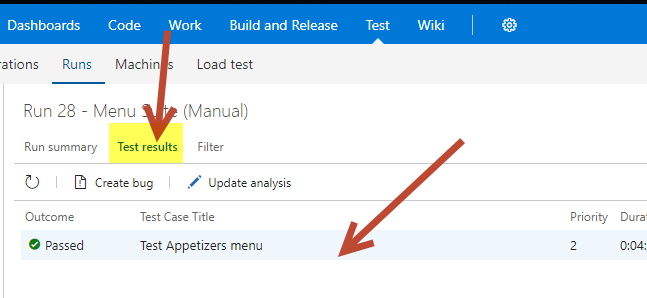
1. Run a test with Video Capture
2. Click on the Run the test option, you should be able to the normal Run Test option without options.
3. When the pop-up shows up, click on the video camera icon.



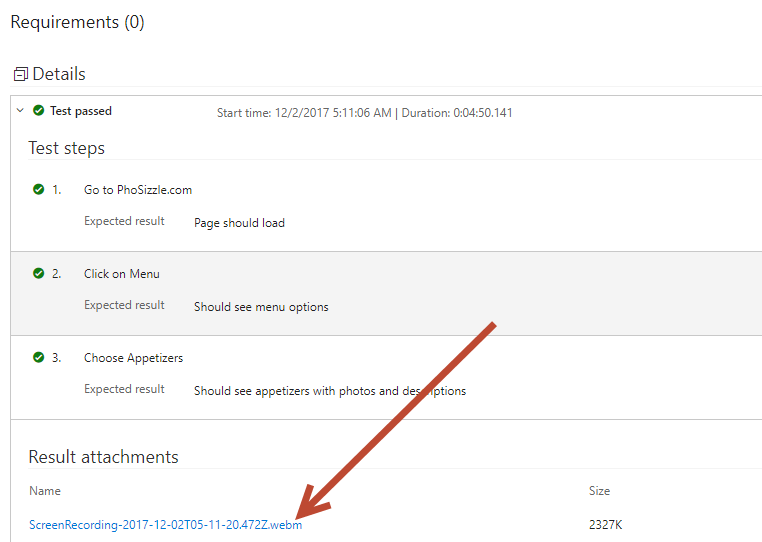
1. A video screen capture will begin. We will mock out fake test steps to go through to see how it shows up in the screen capture.
2. Open another tab in Chrome, go to Google.com.
3. Click the first step as passed.
4. On google.com, search for menu.
5. Click the second step as passed.
6. Click the Images tab in Google.
7. Click the last step as passed. Click Stop on the screen recording.
8. Your file is saved in this test, you can click the paperclip on the bottom left and select the file to download it.
9. Optionally, you can always access the file by doing this: Hit Save and Close.
10. Click on the Runs tab. Choose your most recent test and **double click** on that entry.



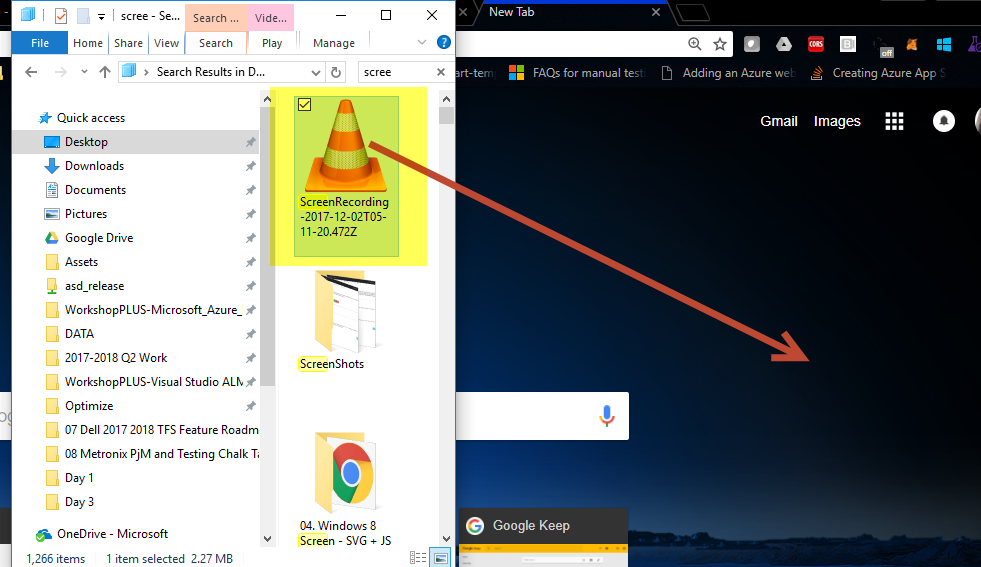
1. Click on the Test Results tab. Then double click on your Test Appetizers test entry.



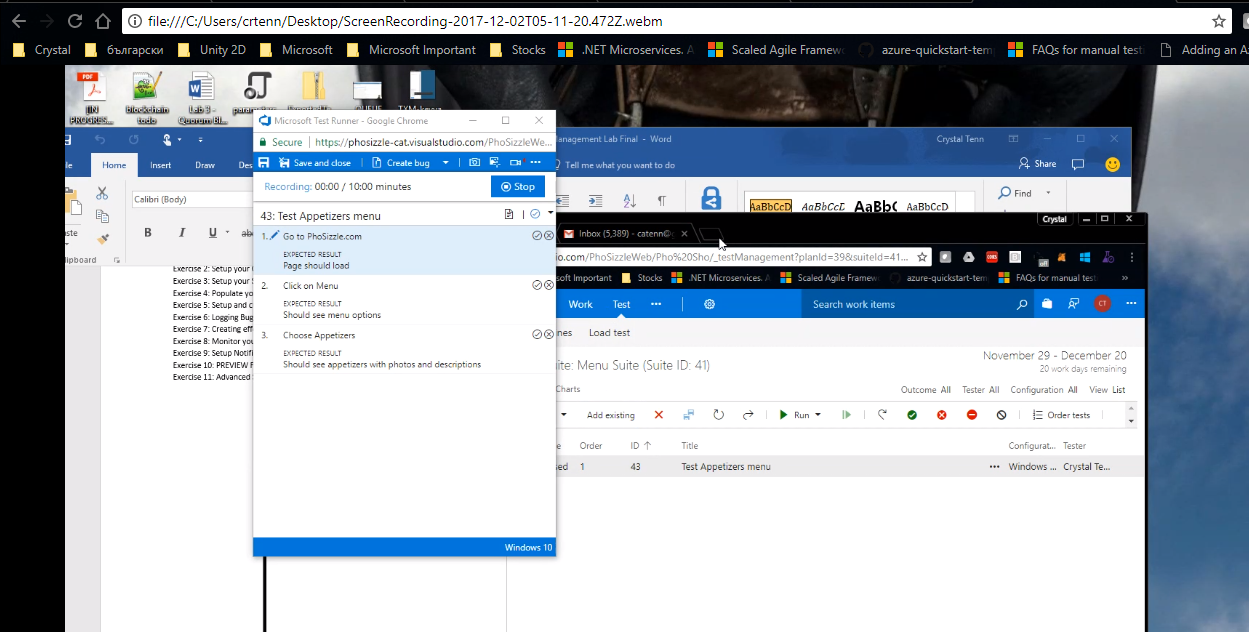
1. On the bottom of this page, you will see the attachment. Download it.



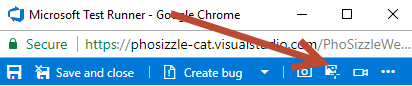
1. Now that it is downloaded. Open a new empty tab in Chrome.
2. Click and drag your webm file that you downloaded into your browser. This type of file can be played directly in the browser. It most likely won’t work in Windows Media Center. VLC play will play it this format also, optionally.



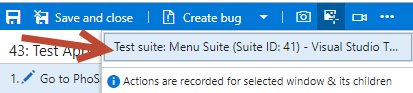
1. You should see your screenshare video playing in Chrome. Check out your video! Press the spacebar to pause / unpause the video. This is the most useful way to show steps for reproduction.



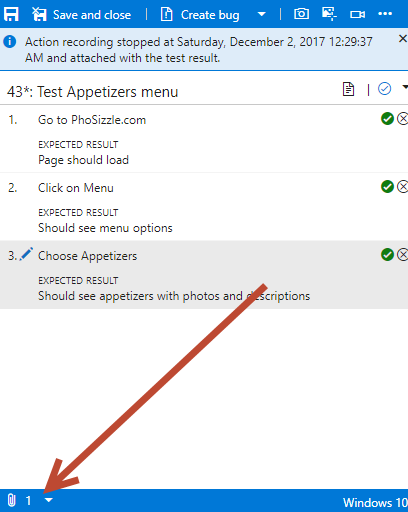
1. Run a test with Actions Captured
2. Go back to the main test page that shows your test cases for your new Suite (Click the test plans tab).
3. Click on the Run the test option, you should be able to the normal Run Test option without options.
4. Click the Photo icon with the crosshair cursor:



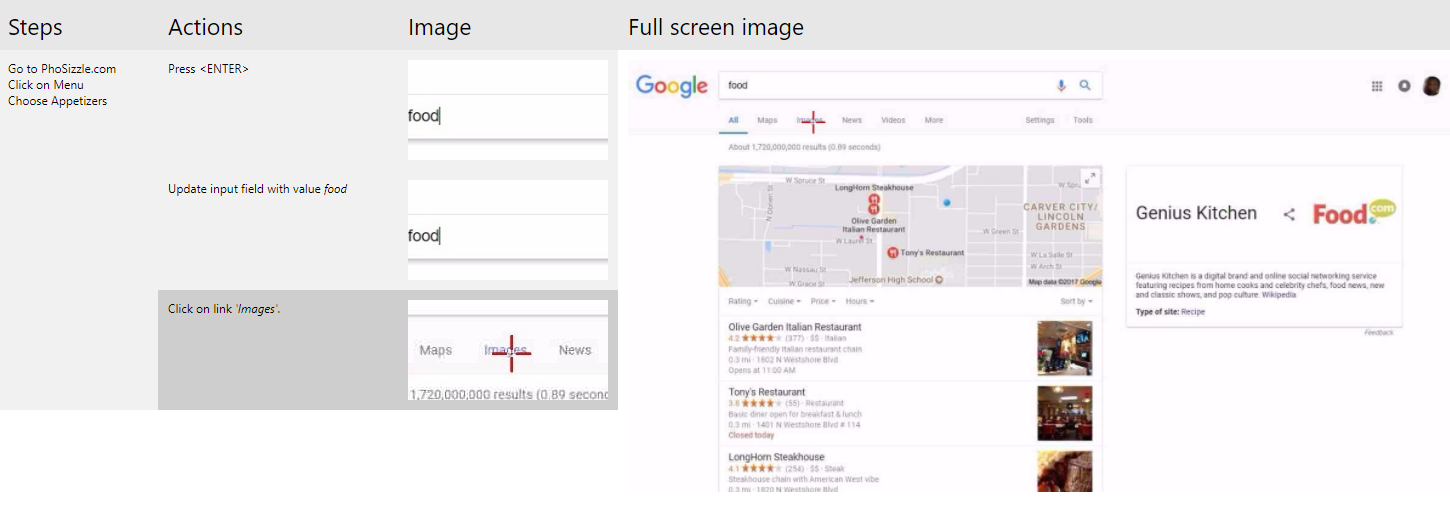
1. Click your test suite:



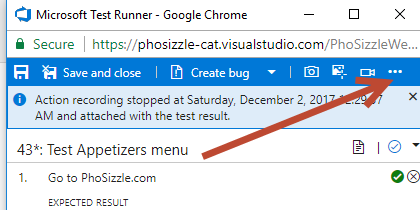
1. Open a new tab, go to Google.com. search for food, click on Images.
2. Click pass step for all steps.
3. Click stop to stop the recording.
4. Click on the file button on the bottom, choose your file, and download it:



1. Find your downloaded file. Open a new empty Chrome tab. Drag the file into your new Chrome empty tab.
2. You should see a screen like this with all your steps recorded with images:



1. Mouseover all your steps and you can see the full-screen image to the right.
2. When you go back to your Test runner, click on the “…” options on the top:



1. Try adding a comment! Note the other options you can use! Hit save and close when you are done. As a note, action recordings are not always supported, check here to find out the requirements: <https://docs.microsoft.com/en-us/visualstudio/test/supported-configurations-and-platforms-for-coded-ui-tests-and-action-recordings>

**Exercise 4: Configuration, Templates, Sharing Parameters, Sharing Steps, Re-ordering, Filtering (25 mins)**

The Pho Sho QA Testers have learned a lot and feel like they could run a basic test now! However, they want to make sure they are being efficient. How can they make a template of a test? They want to know what parameters, shared parameters, and shared steps are! They want to be able to reorder their tests, filter them, and find them easily.

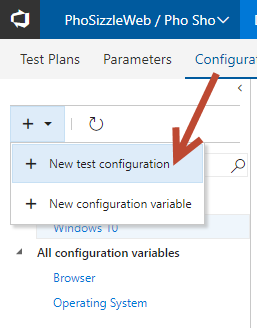
Let’s create a few more test cases and show them how to make tests efficiently!

**Tasks**

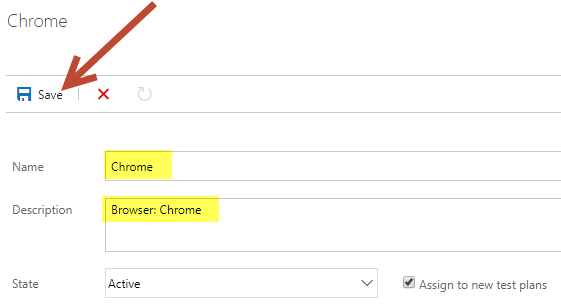
1. Configuration and Template
2. Click on the Configurations tab.



1. Add a new test configuration. Note a test configuration variable is something like: Browser, Operating System, Version of iPhone, Android Operating System, etc…



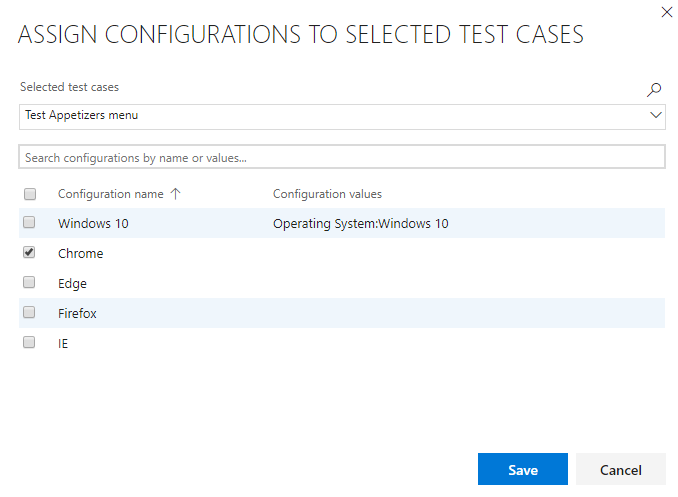
1. Create a new one Named Chrome and with a Description. Hit save when you are done.



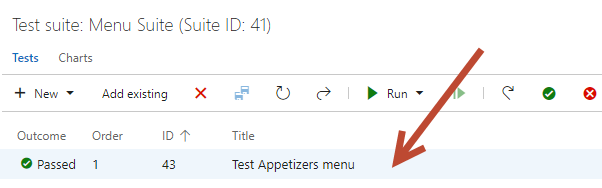
1. Add a couple more configurations like: Firefox, IE, and Edge. And we’re done adding configs!
2. Go to the Pho Sho Team and their Test hub main page.
3. Right click on your test case and choose Assign configurations.



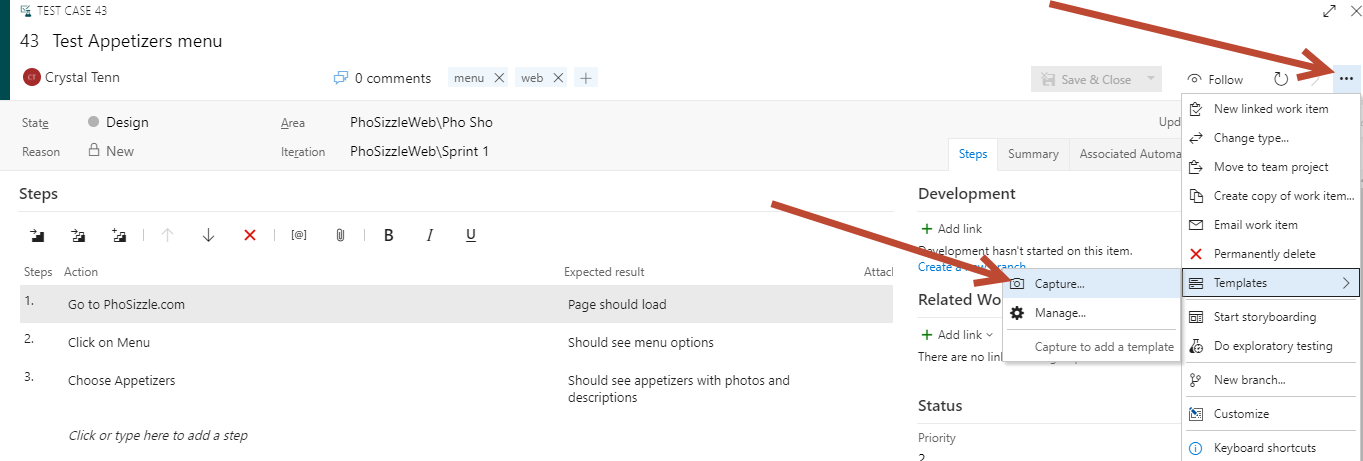
1. You can check different configuration(s), then hit Save.



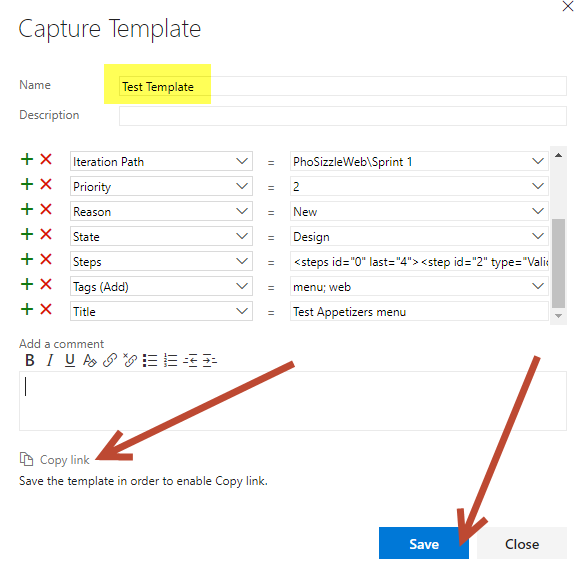
1. Double click on your test item to open the work item view.



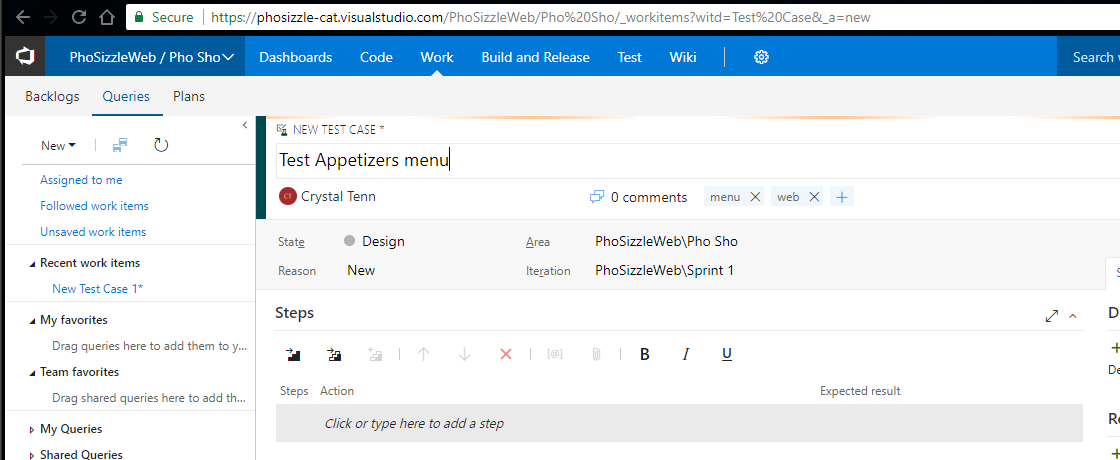
1. Click the “…” button, go to Templates, then Capture.



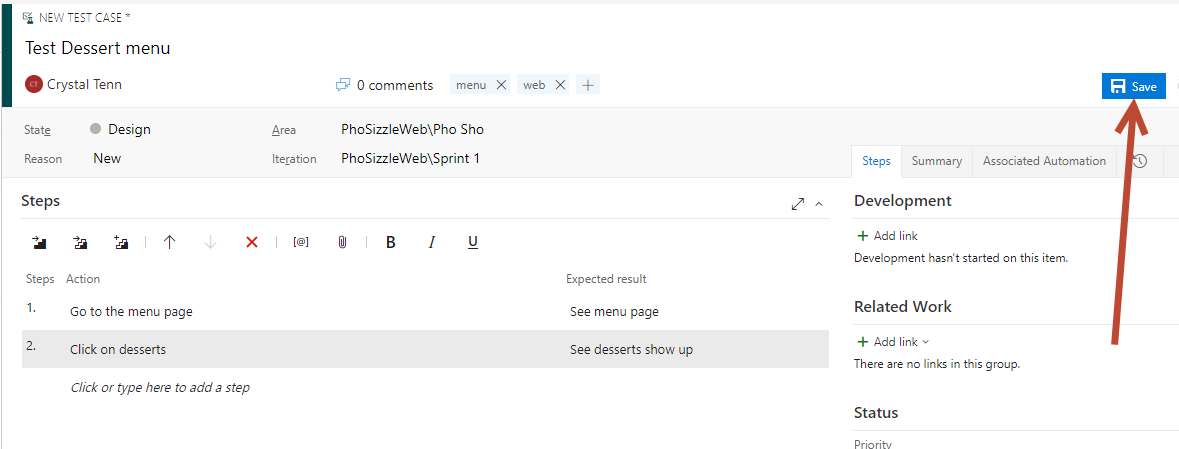
1. Give it a name. You can edit any variables you like, such as Title, tags, or description now to try it out. Click Save. After you hit Save, then you can hit Copy link.



1. Close the template modal.
2. Open a new tab in Chrome and paste the link in.
3. You will be able to work on this base template now, it will have the same information as before.



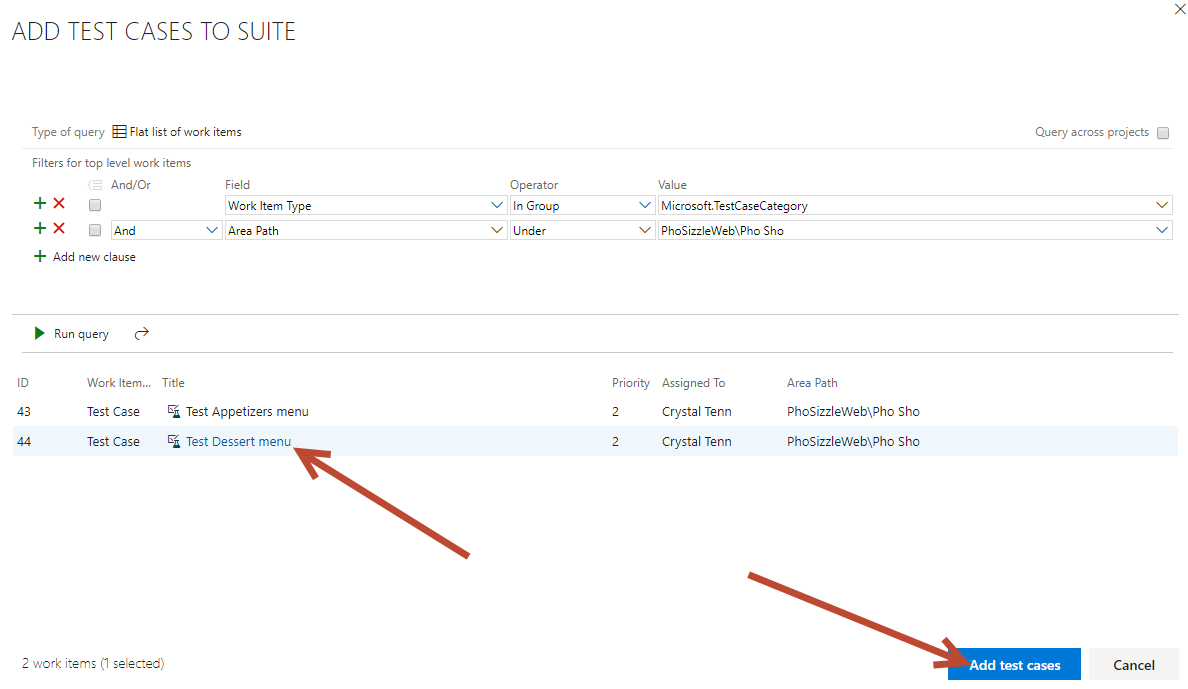
1. Change this one to Dessert and add a couple of steps. Then hit Save.



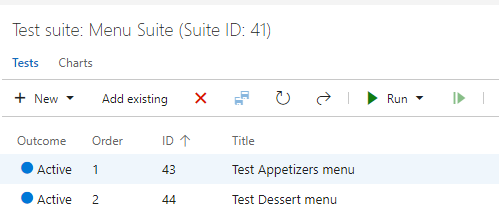
1. We need to add this to our suite. Go back to the Test Plans tab. Click Add existing.



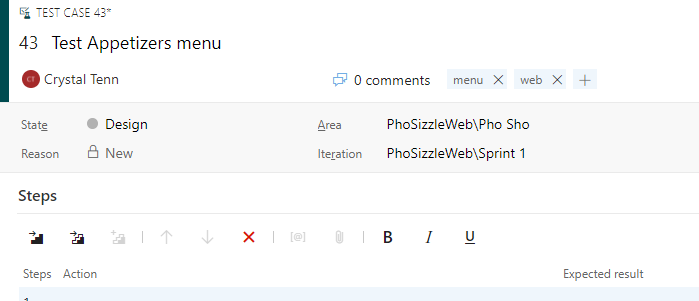
1. Run the default query. You should see both of your test cases show up. Select your new “Dessert” one. Click Add test case.



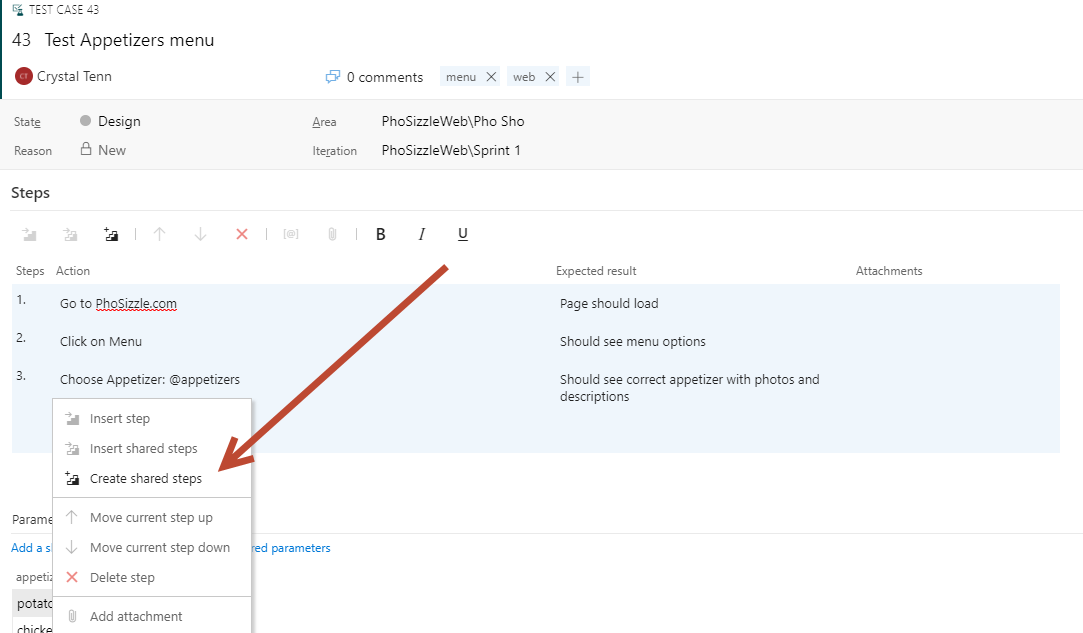
1. Both should show up now in your suite:



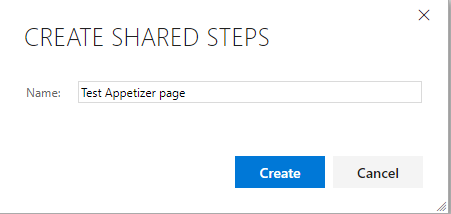
1. Sharing Steps
2. Click into your Test Appetizer work item view.



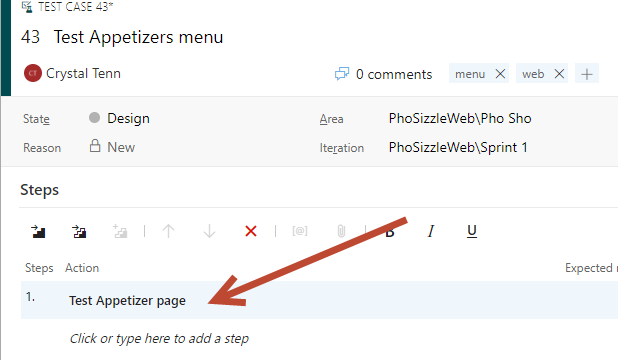
1. Click on the first step to select it.
2. Hold shift down, and then click on the last (3rd) step. You should see that all the steps are lightly highlighted.
3. Right click on your steps.
4. Choose Create Shared Steps.



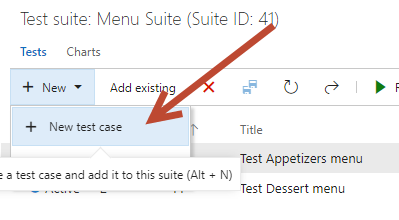
1. Call it “Test Appetizer Page”. Hit Create.



1. You will notice that the steps in your work item / test case are now minimized to your new ‘shared step’.



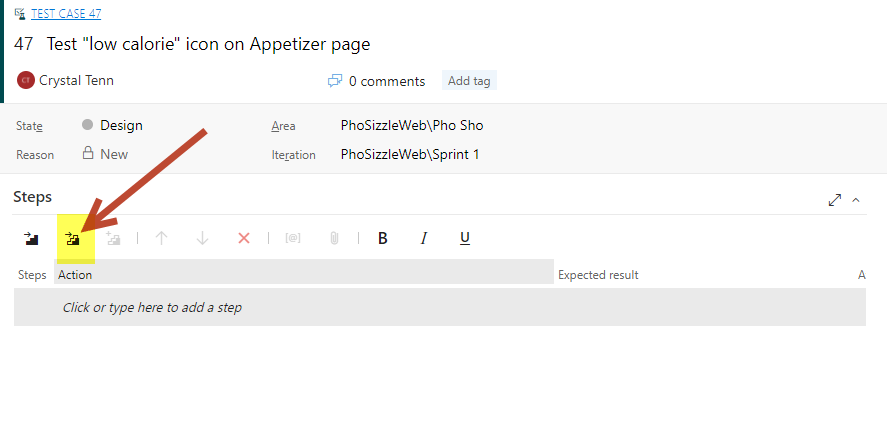
1. Click Save and Close.
2. Let’s create another test case:



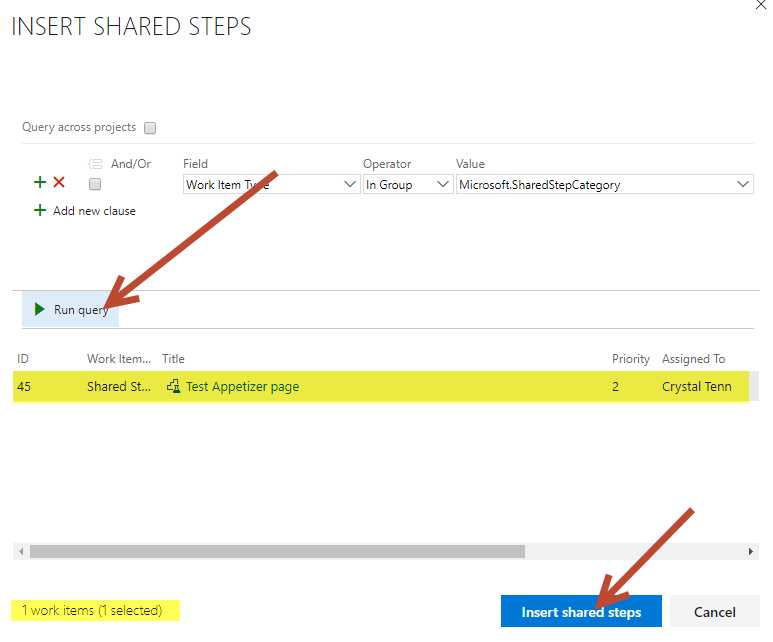
1. Add a title to this page: “Test ‘low calorie’ icon on Appetizer page”



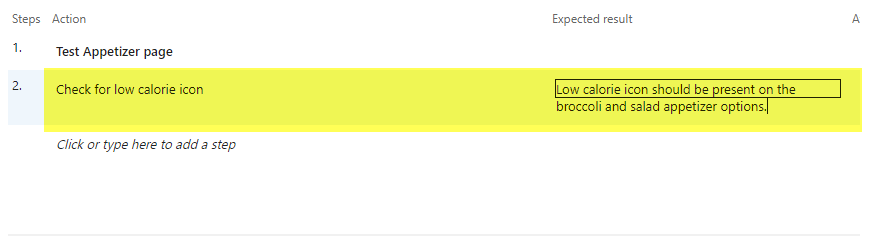
1. Click on the Insert Shared Steps icon.



1. Click on Run query without changing any of the clauses. You should see your Shared Steps show up in the bottom. Select it, then hit Insert Shared Steps.



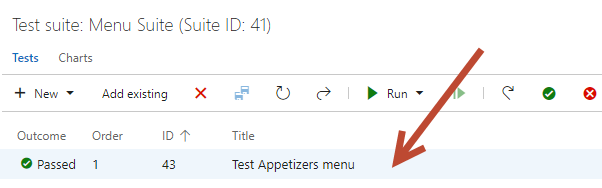
1. Add one more additional step underneath your shared step:



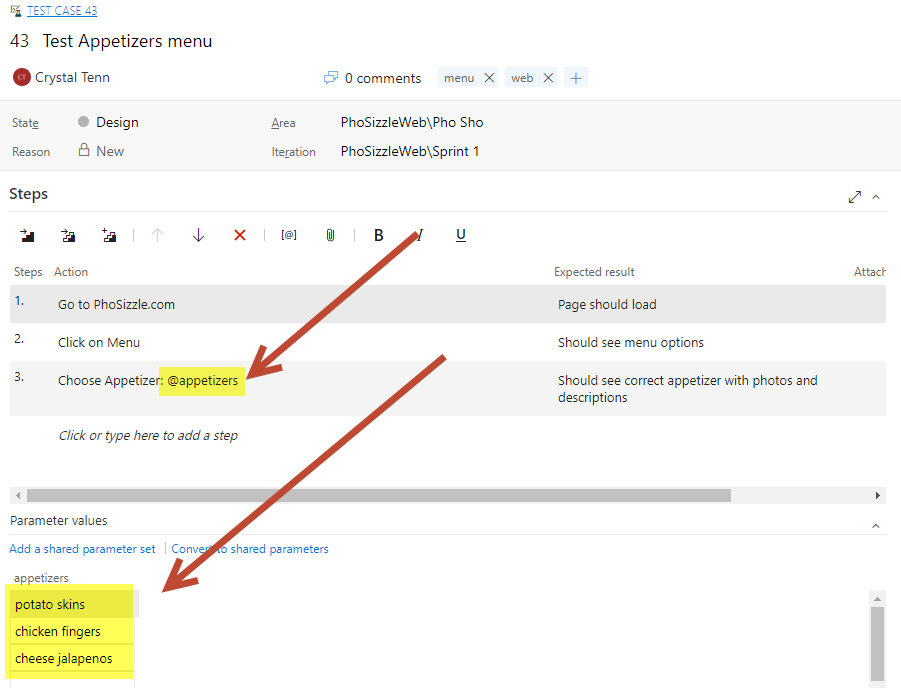
1. Note parameters are empty, we will fix that in the next exercise! Hit Save and Close.
2. Let’s try running our new test now to see how our shared step looks!
3. Select your new “Test low calorie icon on Appetizer Page” test and click Run test.
4. You should see all your shared test numbered as 1.1, 1.2, and 1.3 (different parts of step 1), then you will see your additional step 2.



1. **Parameters and Shared Parameters**
2. Double click on your Appetizer test item to open the work item view.



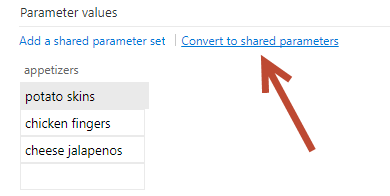
1. You create a parameter by typing a name preceded by "@" in the actions and expected results of your test steps. Let’s create a parameter in Step 3 Action. After you add the @appetizer as in the screenshot below, give it a few seconds for it to register and show up on the bottom of the work item. Once it appears, add 3 random appetizer names or you can use the same ones as in the screenshot.



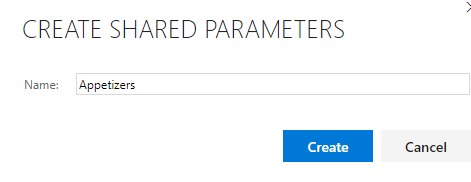
1. Hit Save and Close.
2. Select your Appetizer test.
3. Hit Run.
4. In your web based MTM, you will see different Iterations with each parameter:

|  |  |
| --- | --- |
|  |  |

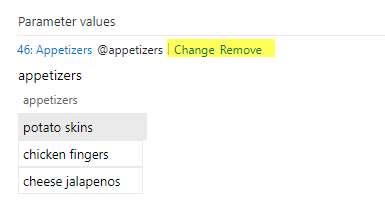
1. Click onto the other 2 iterations and verify the parameter on step 3’s action changes to your other appetizer options.
2. Now let’s learn how to do shared parameters. Close the test window. Open your Appetizer work item again. Look at the bottom of the page where the parameters are located. Click “Convert to shared parameters”.



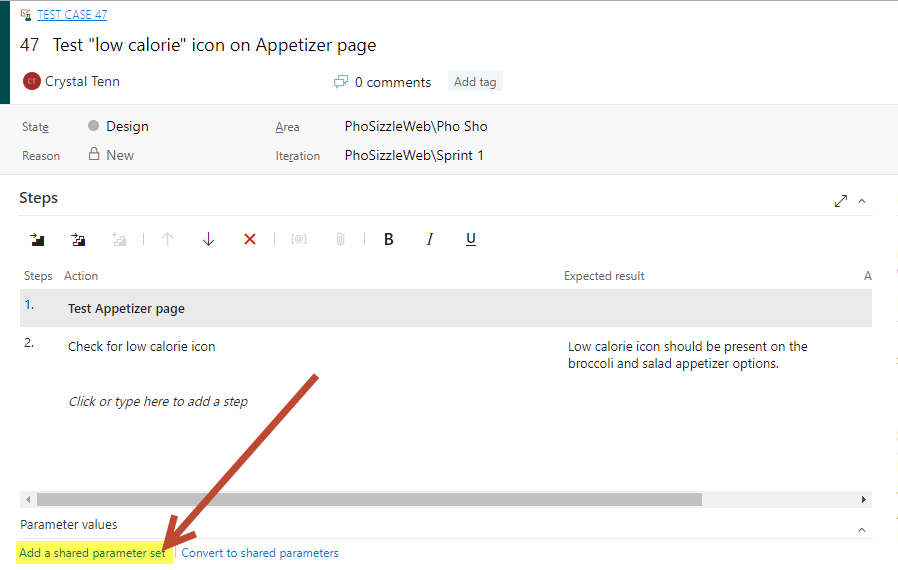
1. Name it Appetizers. Click Create.



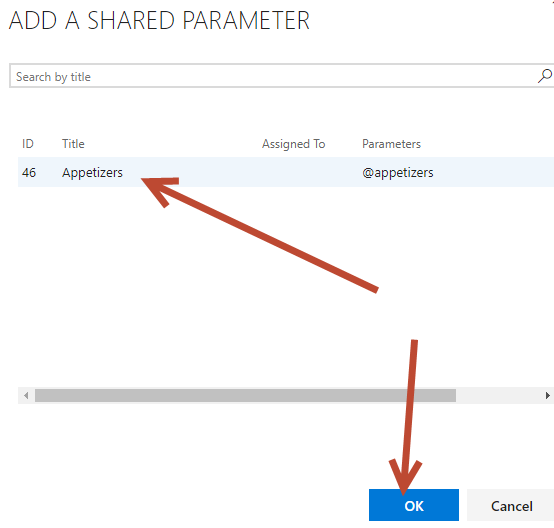
1. Now you will see options to Change or Remove. You don’t need to hit either of those, this is just for your reference to see that it changed how your Parameters work now that they are shared.



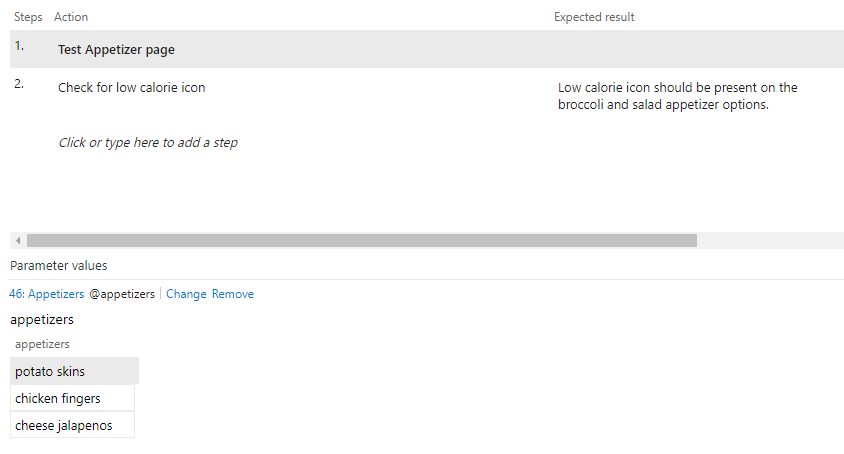
1. Click Save and Close on your test case / work item. Now we will need to add these shared parameters to our new test case.
2. Double click on the “Test low calorie icon on Appetizer page” to open up the modal.
3. Click “Add a shared parameters set” on the bottom of the work item.



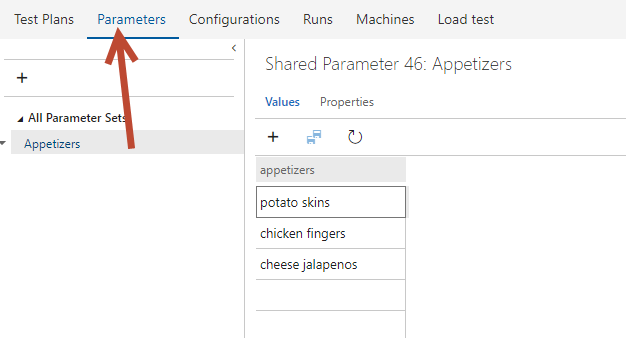
1. Click on the Appetizers parameter and hit OK.



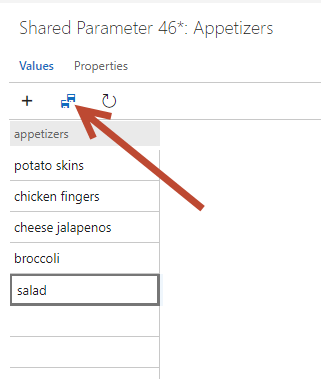
1. Now you should see the appetizers parameter filled in.



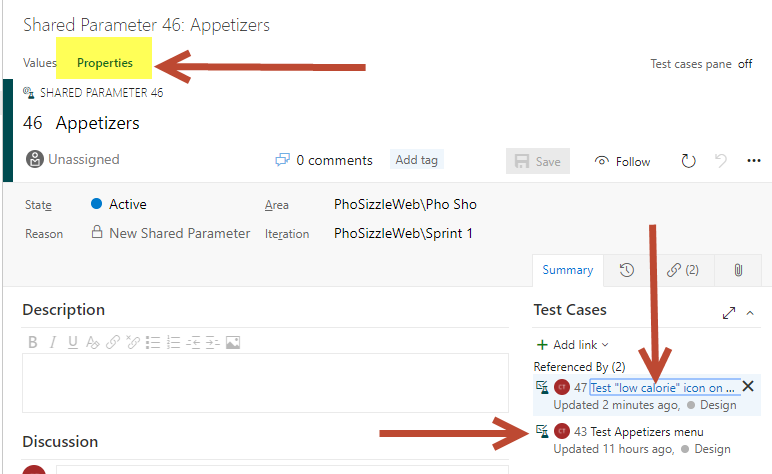
1. We should edit the set of parameters now that we know of two more appetizers from the steps, the broccoli and salad ones! Let’s edit the parameters and ensure that it updates the parameters for both of our Appetizer related work items.
2. Hit Save and Close. Click on the Parameter Tab.



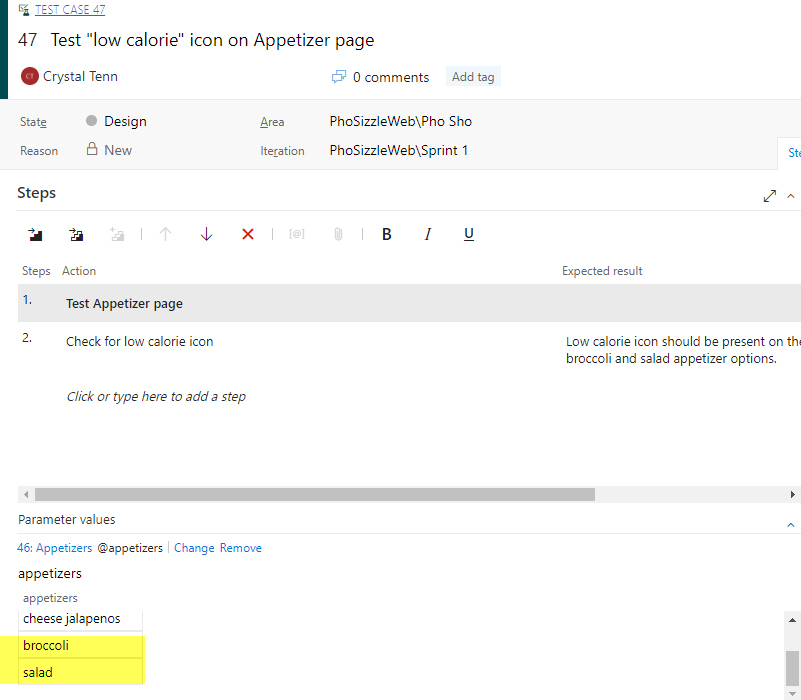
1. Add in ‘broccoli’ and ‘salad’ and hit the Save button.



1. Click Properties and you can see all associated Test Cases with this Parameter set.



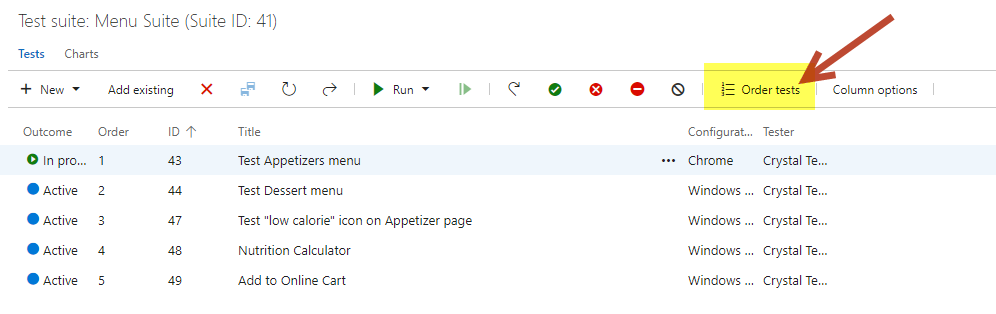
1. Click on one of your work items and verify that the Parameters have changed.



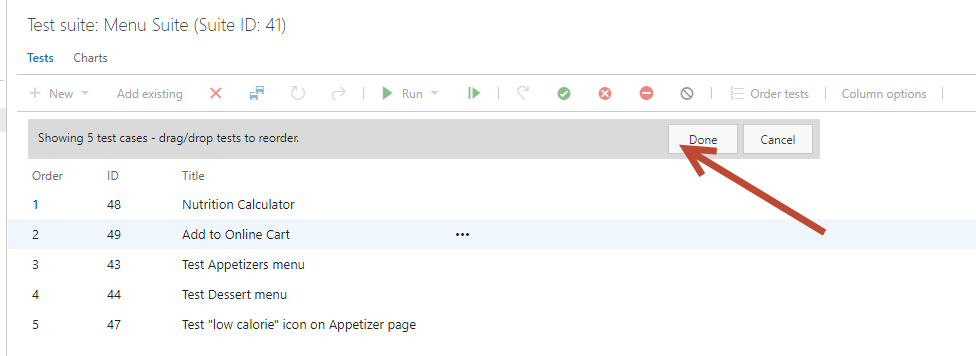
1. Other examples of good parameter use would be:

* A repeating username and password login that is needed for multiple test cases
* A varying set of numbers that should be manually entered into an input box to ensure the page doesn’t break like: -1, -.5, 0, 1, 1000, 99999, “asdf”, “\*?”, “ “. While not all of the choices need to work, the page should gracefully handle error messages and not break with random strings being entered.. or it should not even allow values that aren’t whole numbers potentially.
* Sizes for clothing like “small, medium, large” etc. and making sure that each of these show up in the cart later
* Anything that you feel is like making many repetitive tests, see if you can find one common denominator that you can make into a parameter!
* **Good Practice Note!** We could refactor our tests, so they cover more ground. If your restaurant doesn’t have anything special about their menu sections and each section is the same, you could make it so that the Test ‘Appetizers’ Menu is just ‘Test Menu Section’ and the parameters for the test are: ‘Appetizers’, ‘Lunch’, ‘Dinner’, ‘Dessert’, etc. You might then be able to remove the “low calorie” test case and have a step in the one test case that ensures that any “low calorie” icons are seen on the page, or even specify depending on the page which ones should be labeled “low calorie”. So, instead of 3 tests (as we have now), you would have one test case that covers the whole menu + the “low calorie” icon! If you have extra time, try it out! If not, no worries and keep it in mind for future reference.

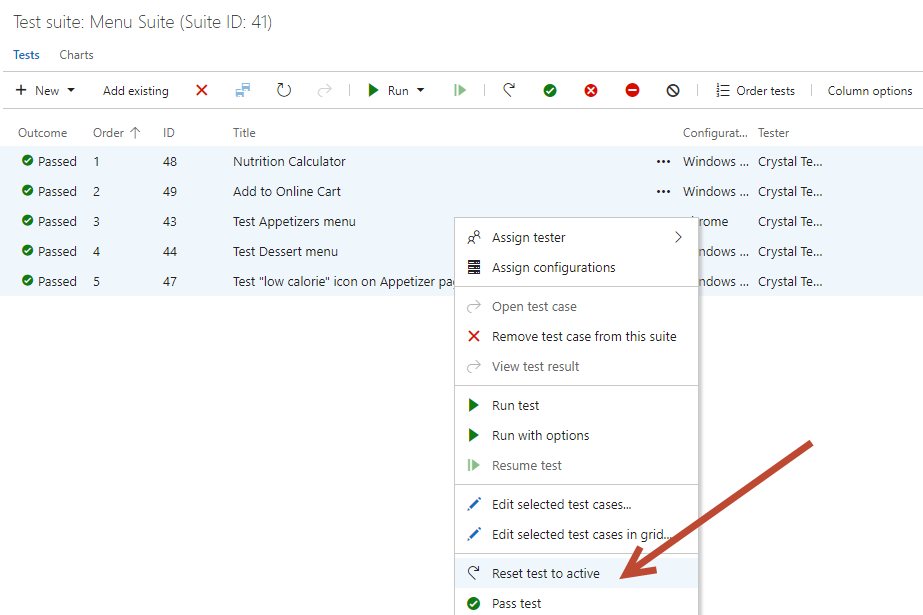
1. Re-ordering tests and Filtering tests
2. Go to the main Test Plan page. Add 2 more new test cases to your page: “Nutrition Calculator” and “Add to Online Cart”, just the title is good for now and don’t worry about the steps / other content.
3. Click on Order Tests.



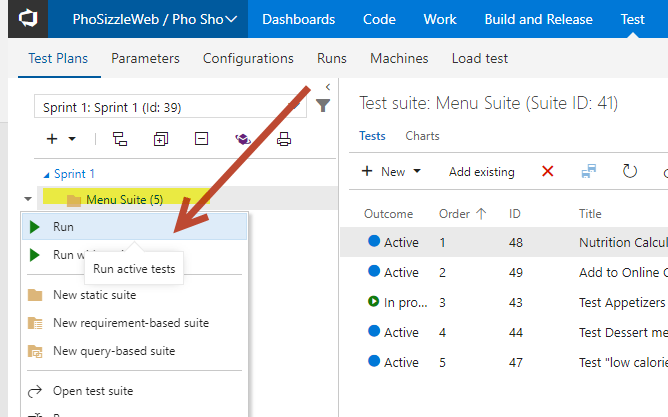
1. Try dragging and dropping the tests to reorder them. Try to put the Nutrition Calculator as the first test, then the Add to Online Cart, then any order for the other test(s). Hit Done to save your changes.



1. First, make sure all your tests are in “Active” mode so you can run the suite. Click on the first test one time to select it. Shift click on the last test, so that all the tests are selected. Right click on them and choose “Reset test to active”.



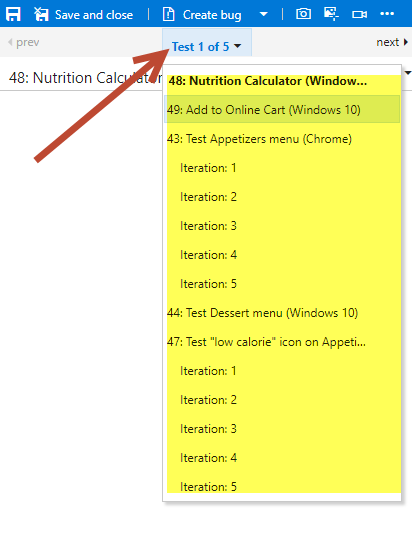
1. Before we had only run one test at a time. Now, let’s try running the whole suite to see the order your tests are being run. On the left menu, right click on your “Menu Suite” and hit Run. Tests need to be in Active in order to be run in the Suite. If any tests are not in Active, they will be skipped over in the Suite run. Just right click a test to set it to active if needed.



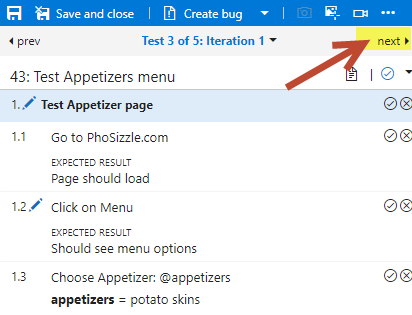
1. You should see multiple tests, in my case I have 5 tests in the suite. As a note, the Nutrition and Online Cart tests will be empty since we did not add any steps to them. Just hit the check button to pass them.



1. If you click on the text that says “Test 1 of 5” then you can see a dropdown of all the tests and iterations (different parameters for the same test) and this can also be used to navigate around to different tests.



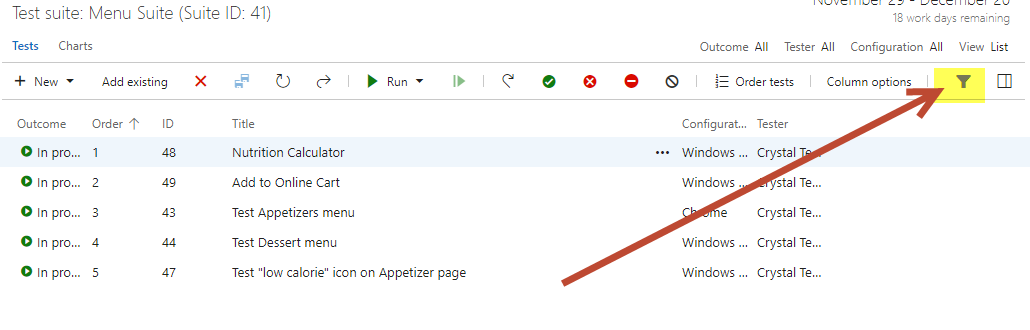
1. Click on the next button and explore the other tests, try passing and failing whole tests or specific steps.



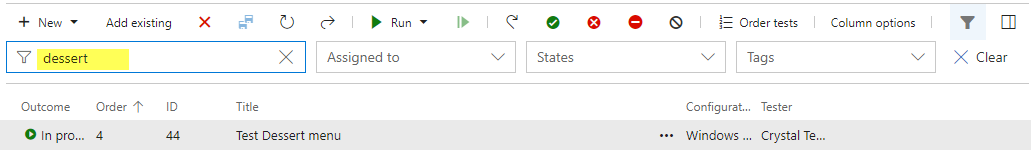
1. When you are done experimenting, hit Save and Close.

*Note: During this exercise you learned how to run test cases by the suite. Typically, you will need to run them by suite more commonly in a work scenario. You will run them by the specific test is if you’re working with the developers and they want a specific test run that they think they fixed or while you are building the tests themselves and just want to try one out to see if it’s built correctly.*

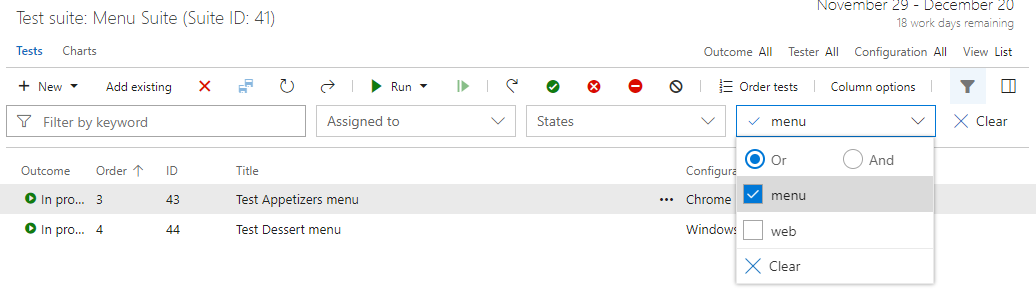
1. If we had a **lot** of tests, it would be really useful to be able to filter them. Let’s try out filtering. Press the filter icon.



1. If you type in “desserts” you will filter down to one test case:



1. Remove your filtered keyword. Or you can filter by other things like Assigned To, States, or Tags. Try out each of these.



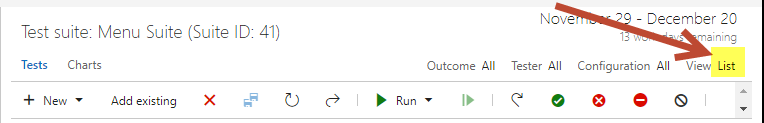
1. When you are done experimenting with filtering, you are done with this exercise. Click the Filter button again if you want to stop filtering.

**Exercise 5: Manual Testing: Excel / Grid View (20 mins)**

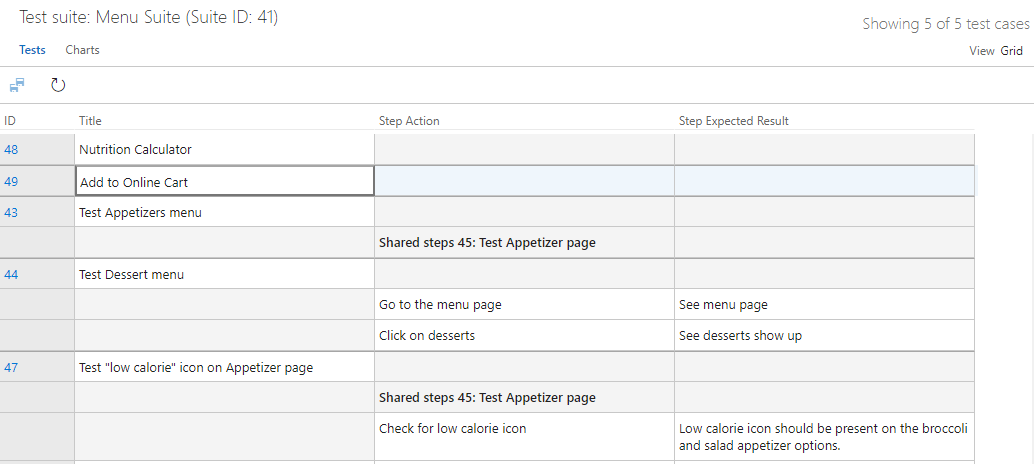
The Pho Sho QA Testers want to know how to use Excel, they love using Excel. They said Excel is much faster and they can make many more tests quickly. They also want to know how to use Excel or Grid view to edit many columns all at once.

**Tasks**

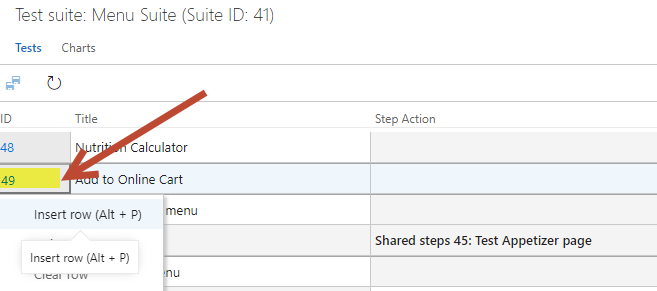
1. Grid Mode and Excel
2. Go to the main testing page (Test Plans page).
3. On the top right of the page, click on the text by View that says “List”.



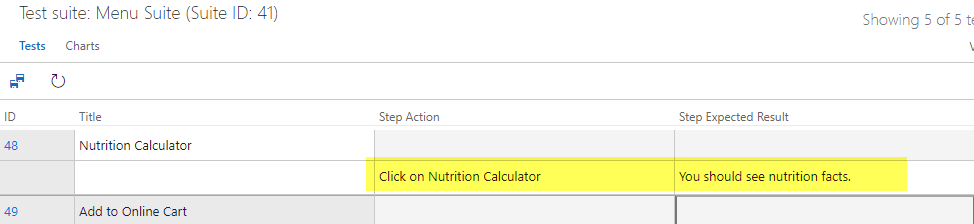
1. You will see a grid view like this:



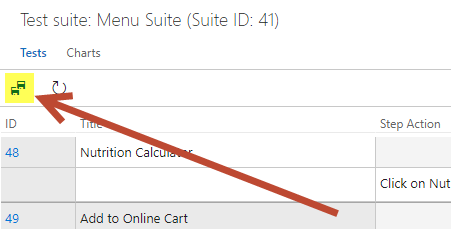
1. Right click on row 49 and choose Insert row. This will insert a row above 49.



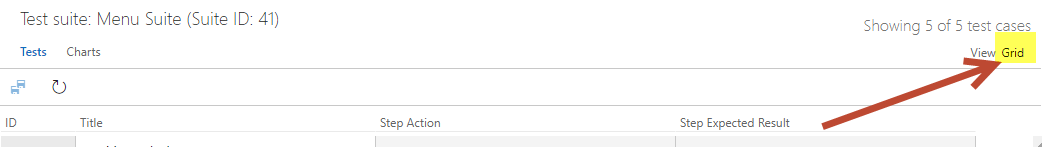
1. For ID 48, add a Step and Expected Result into your new grid row. See highlighted below.



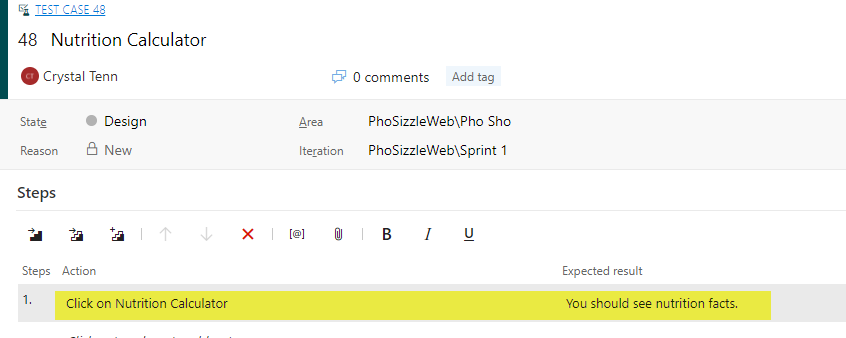
1. Hit the save button.



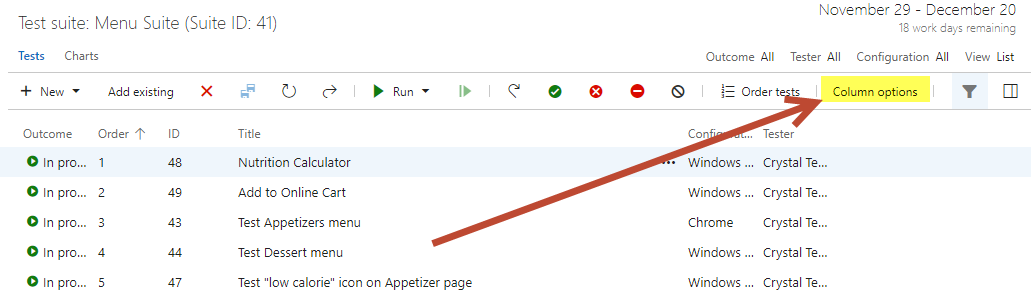
1. Click on the List ‘Grid’ text so you go back to the List view.



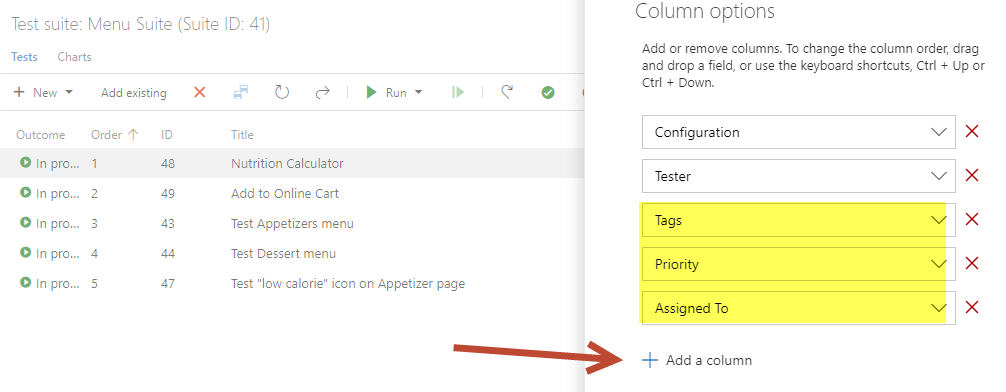
1. Open the Nutrition Calculator test case. You should see your step that you added in the Grid mode.



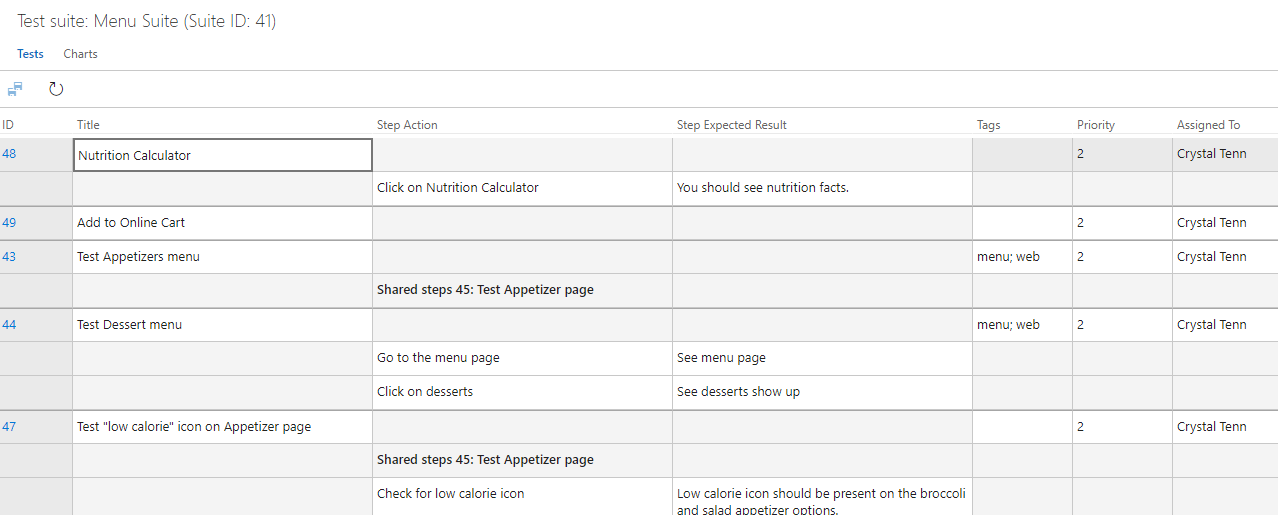
1. Right now, there aren’t a lot of options to edit in the grid option. How do you add more so you can edit Tags, Priority, and Assigned To for example? Click on Column options.



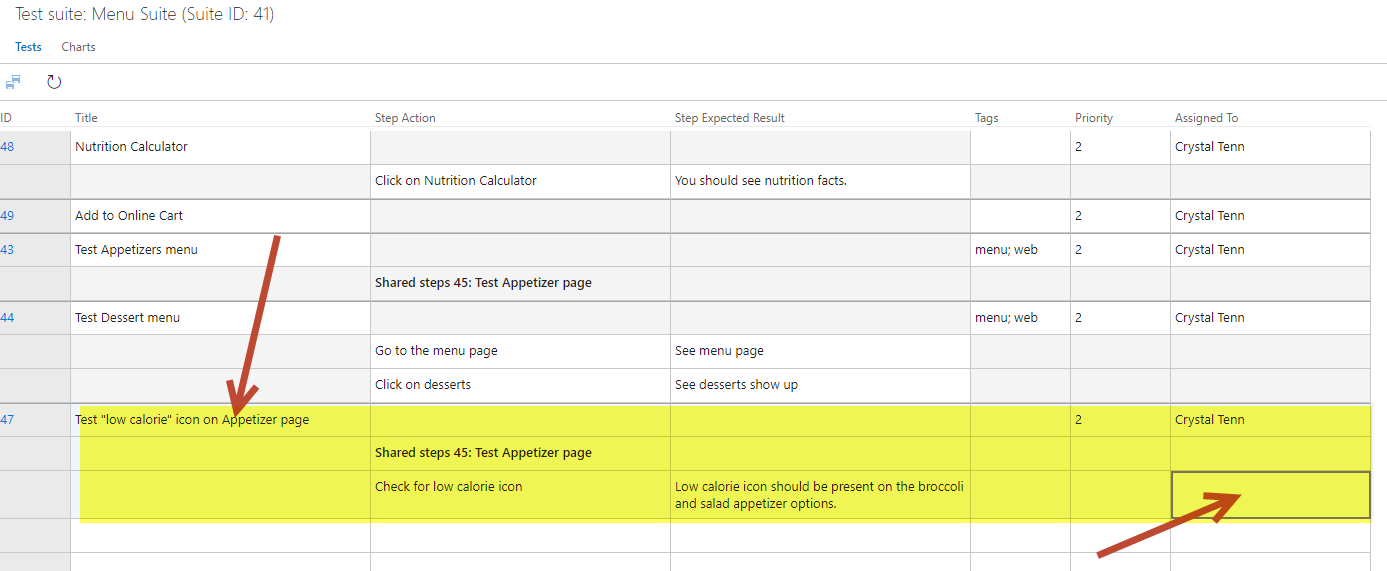
1. On the right popup, click Add a column. Add 3 columns: Tags, Priority, and Assigned To. Hit OK when you are done.



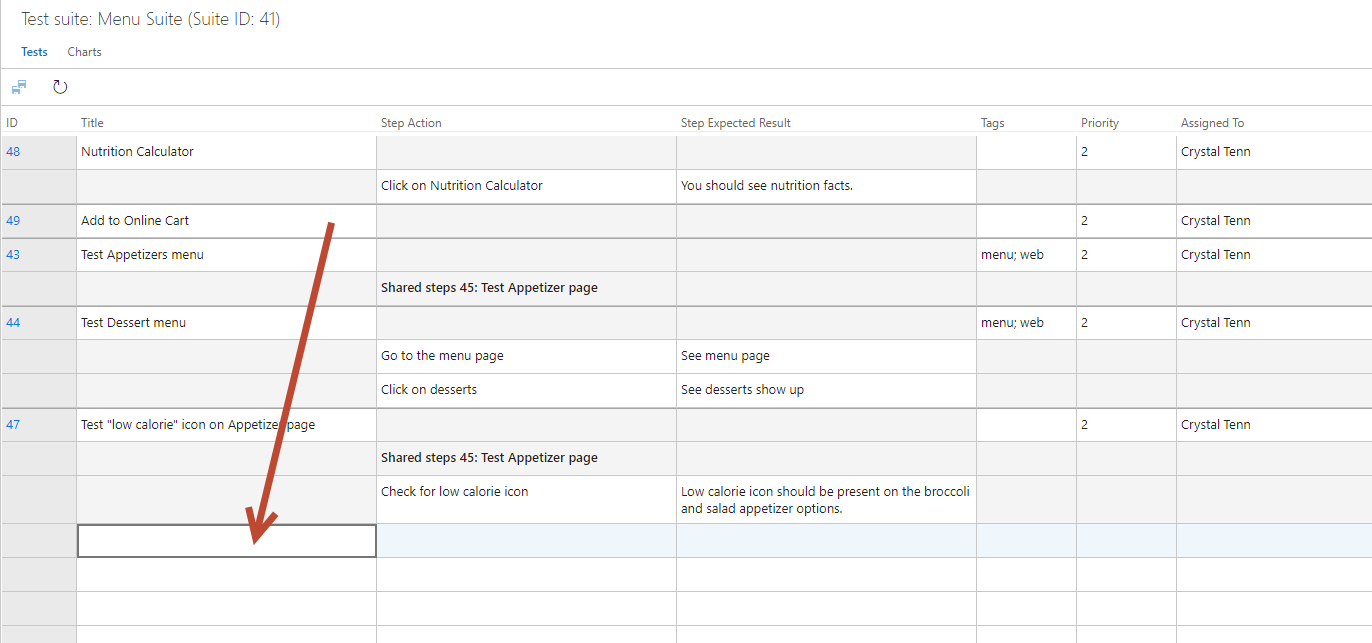
1. All your columns should show up on your list view.
2. Now swap back to Grid view.
3. Now you should see more columns to enter. They are not required. However, you have the option to edit them now that they have been added.



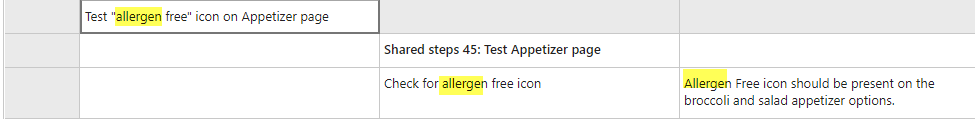
1. Try copying a section from the grid. Click on the Title of the one you want to copy. Then hold shift and click on the bottom right grid of where you want to select. See in the screenshot, the arrows are where you click. Then the highlighted is what gets selected. Note you will not need an ID because it is a read-only view and VSTS will fill in the ID for you after you hit Save.



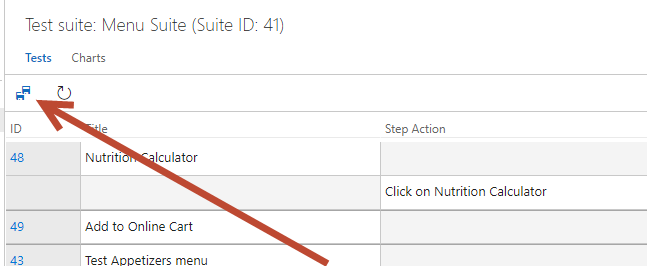
1. Now hit Ctrl + C for copy.
2. Click to select the first completely empty row. Then hit Ctrl + V to paste into that row.



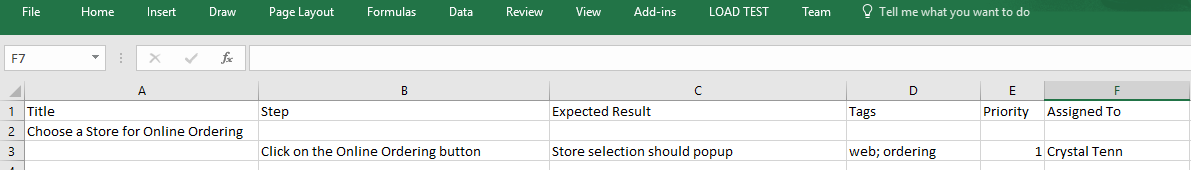
1. Edit the text so it has a different title, change the “low calorie” text to “allergen free”.



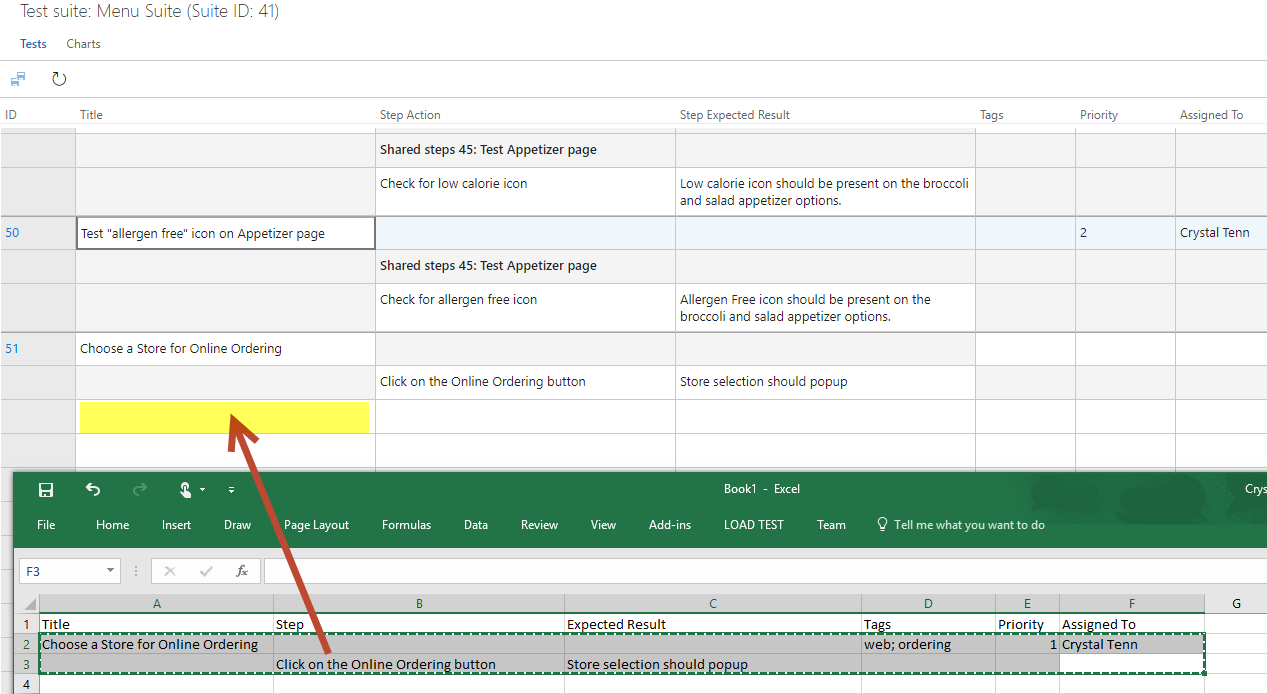
1. Hit Save.



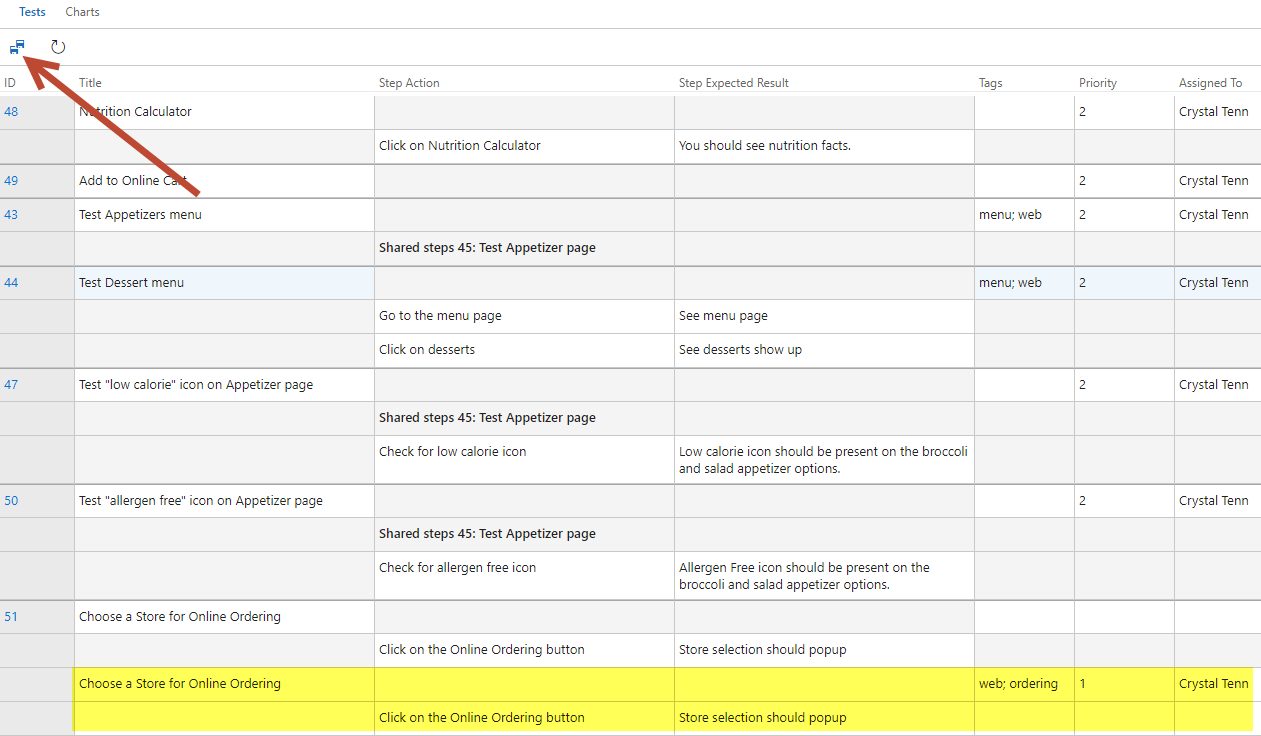
1. You should notice the “allergen free” test case entry will have an ID after you hit Save.
2. Now open Excel. Add the columns shows in the screenshot below. Create an entry with this format.



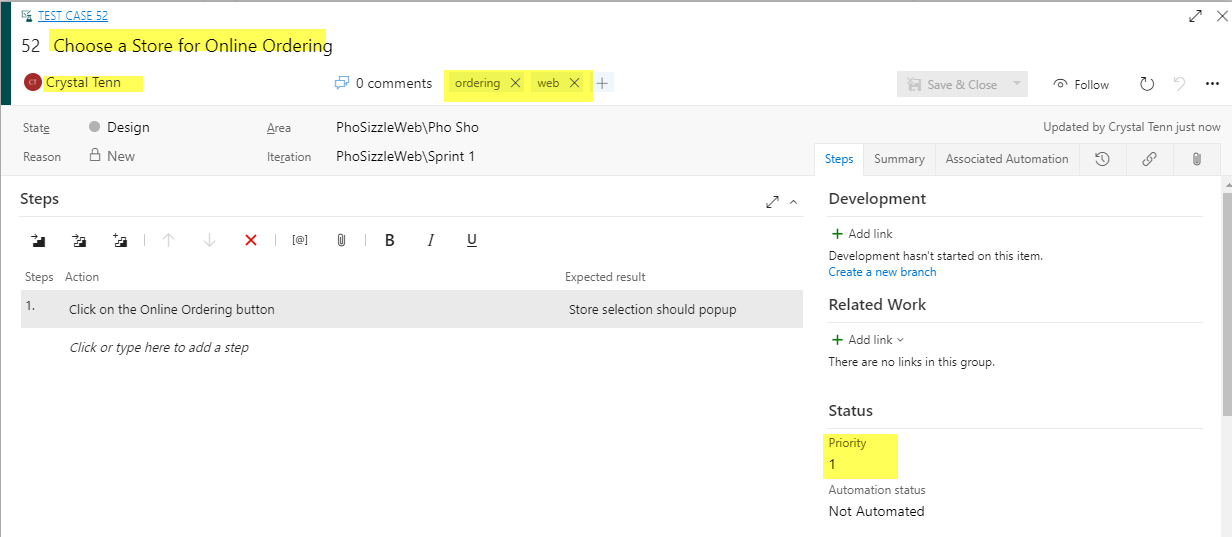
1. Copy and paste it into the Grid view in VSTS.



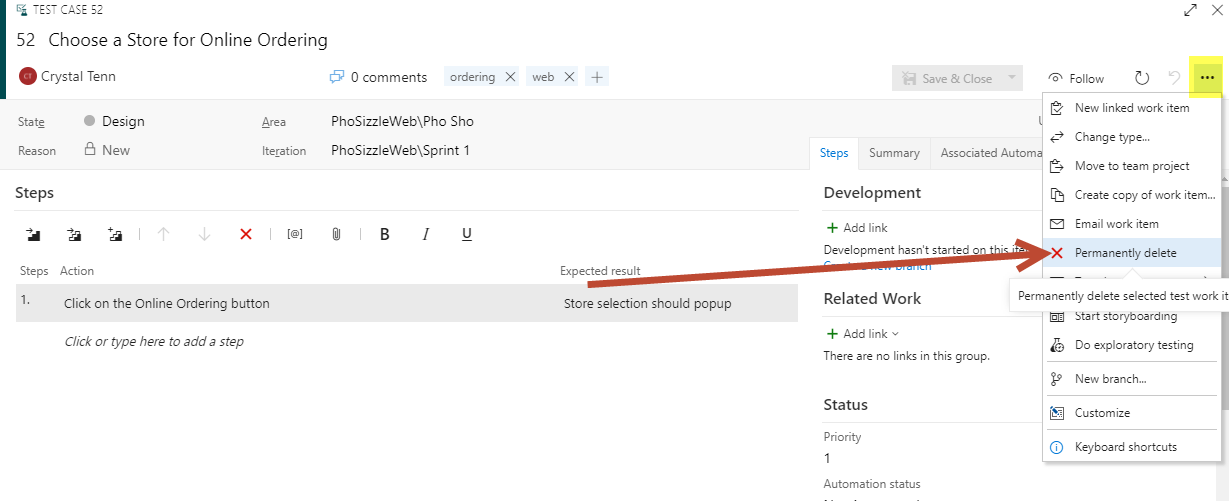
1. Hit the Save button.



1. The ID will get assigned. Now click on List view mode.
2. Open your new work item.



1. Let’s learn how to Delete a work item / test case. In case you accidentally made a test case you don’t want at all anymore, you can delete it. Click on the “…” button.
2. Choose the Permanently delete option.



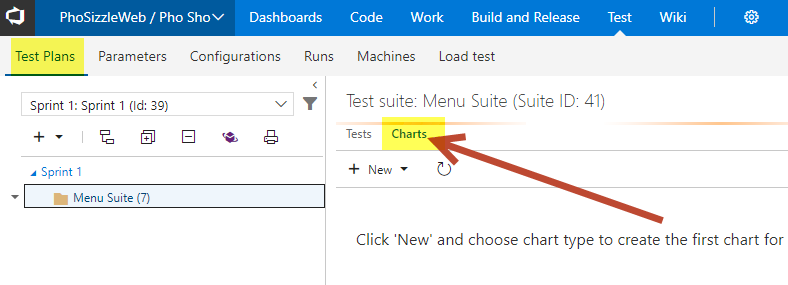
1. This will remove your test case!
2. You are done with this exercise and all Manual Test Case / Work item editing! Feel free to experiment around with this area (test cases, running suites/test cases, etc) with any remaining time and to ask your instructor any questions.
3. Please take 5-10 minutes to scan through common FAQs on Manual Testing in VSTS: <https://docs.microsoft.com/en-us/vsts/manual-test/reference-qa#testplans>

**Exercise 6: Charts, Test Summaries and adding Widgets to the Dashboard (15 mins)**

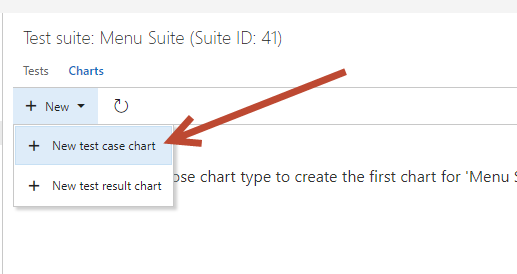
The Pho Sho QA Testers want to check out all the summaries of the tests they have been running while they were learning! Also, they want to create their own Dashboard with widgets to track their testing.

**Tasks**

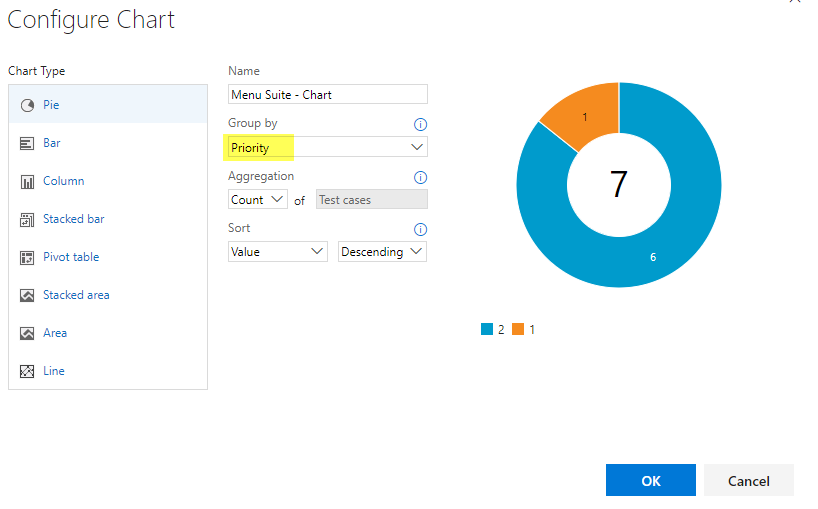
1. Charts
2. From the main Test Plans tab, click on Charts.



1. Add one new Test Case chart.



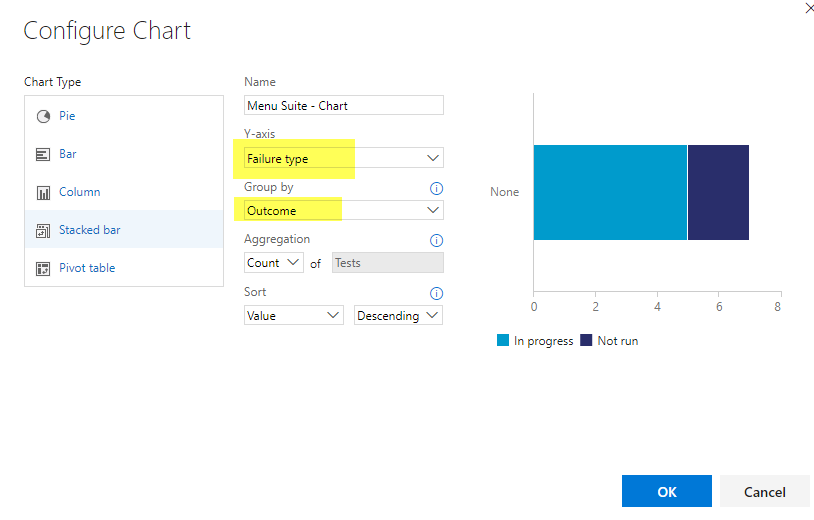
1. Create a Pie Chart grouped by Priority. Click OK.



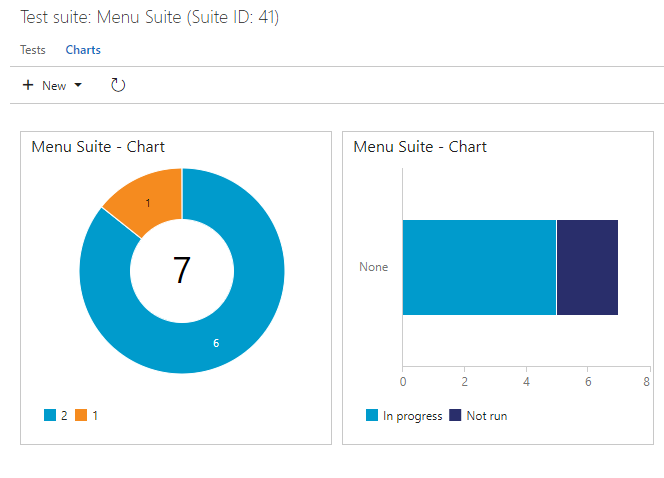
1. You should see your chart show up. Then Add one new Test Result chart.



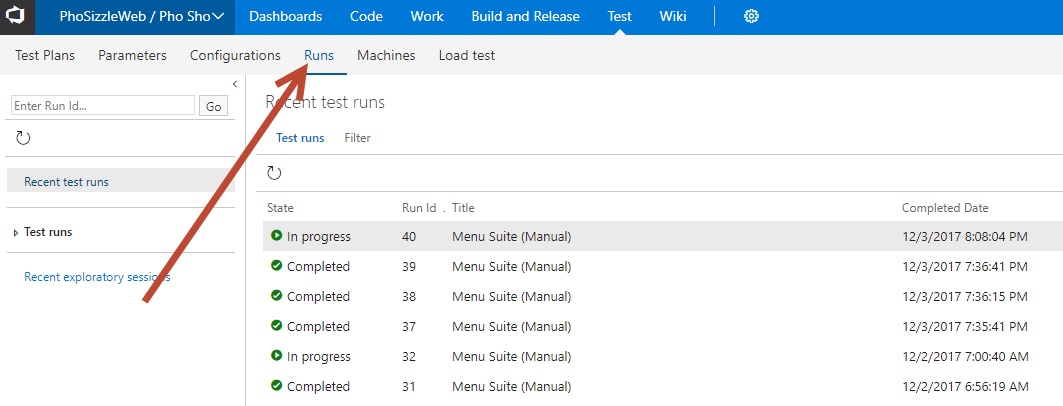
1. Try making a Stacked Bar chart based on Failure Type and Outcome. Click OK.



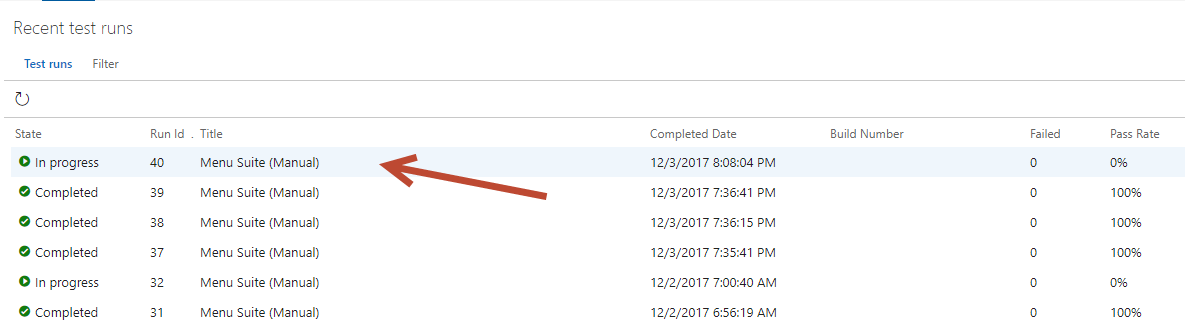
1. You should see your two charts like this:



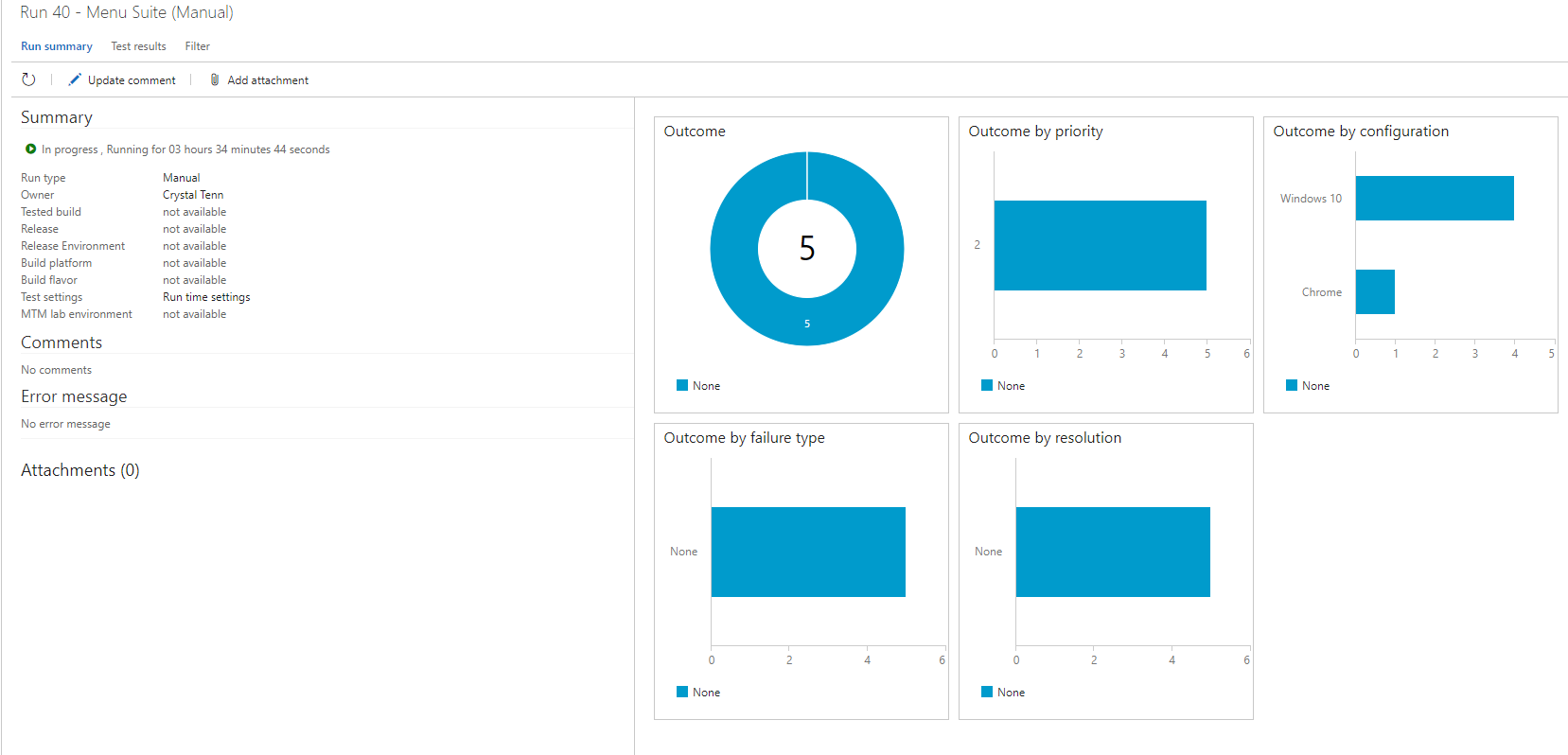
1. Create two more charts of your choosing to practice!
2. Test Summary
3. Click on the Runs tab.



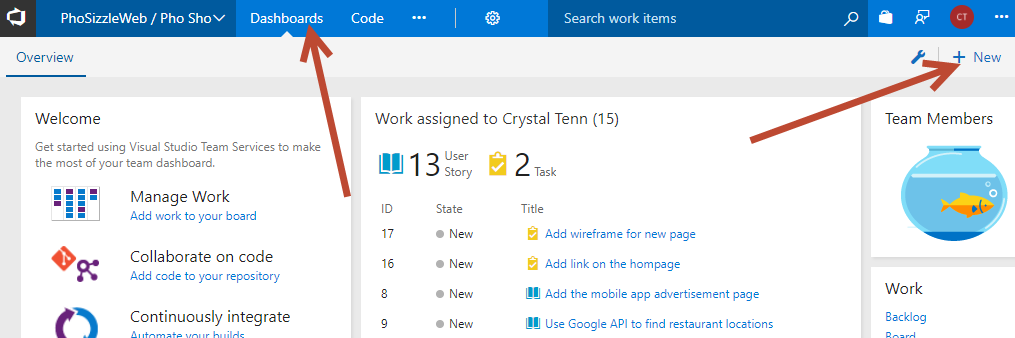
1. Double click on your most recent test run.



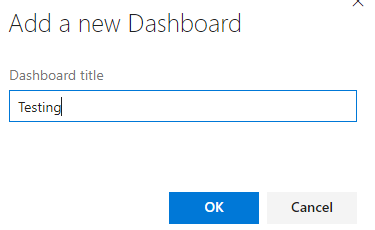
1. Check out the Runs Summary on the right. There are many charts here. Understand what the charts do and try to make the best use out of your future test case building, such as by adding configurations and priority.



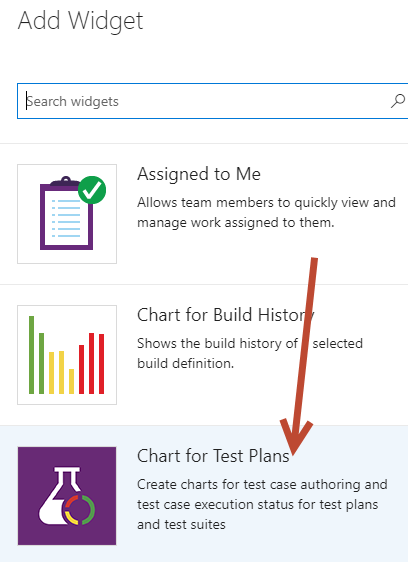
1. Create a Test Dashboard and add Useful Widgets
2. Click on the Dashboards Tab and hit New.



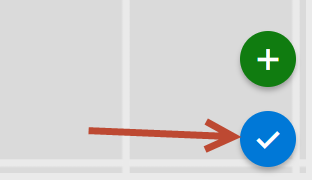
1. Calling it “Testing”



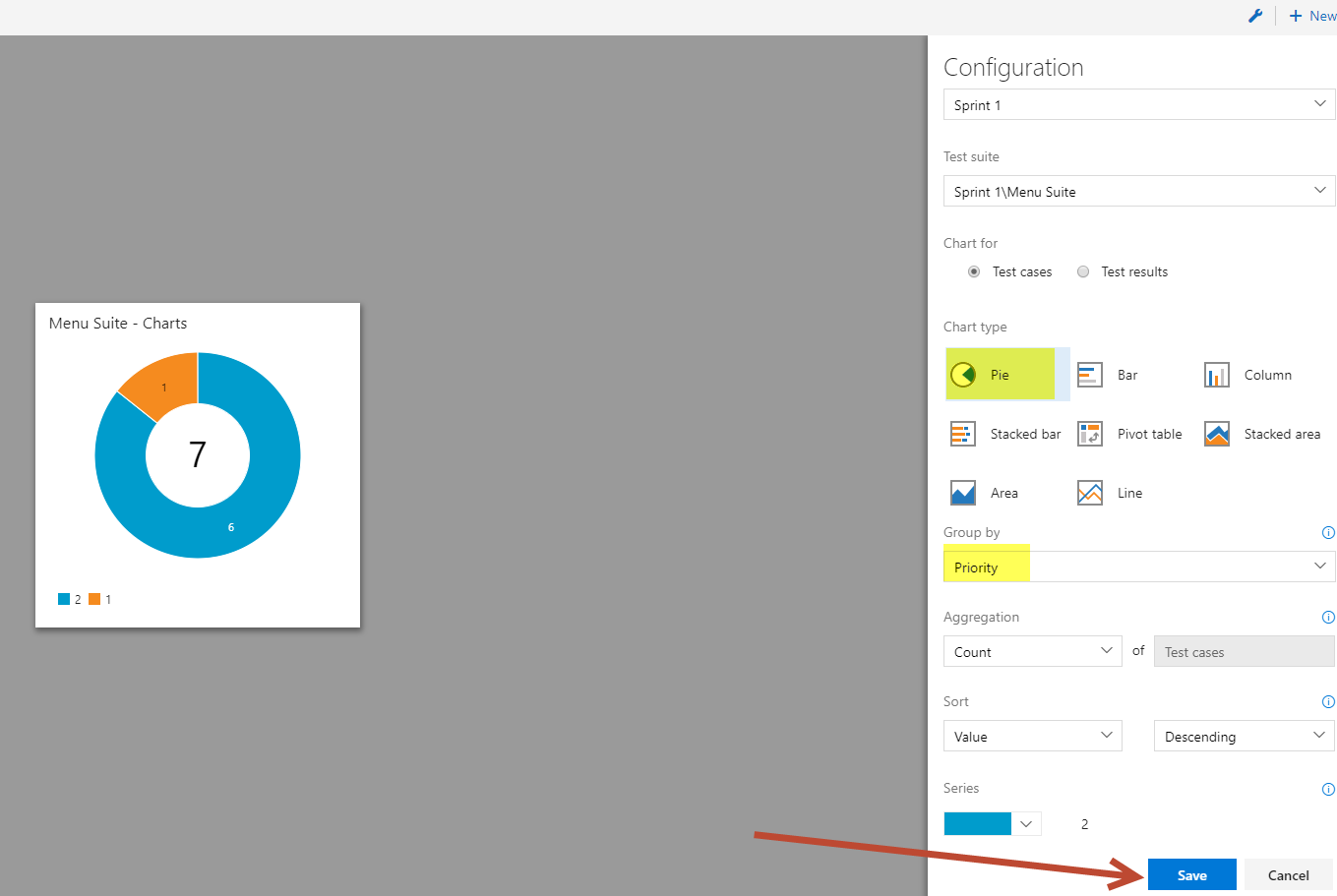
1. Choose the Chart for Test Plans widget.



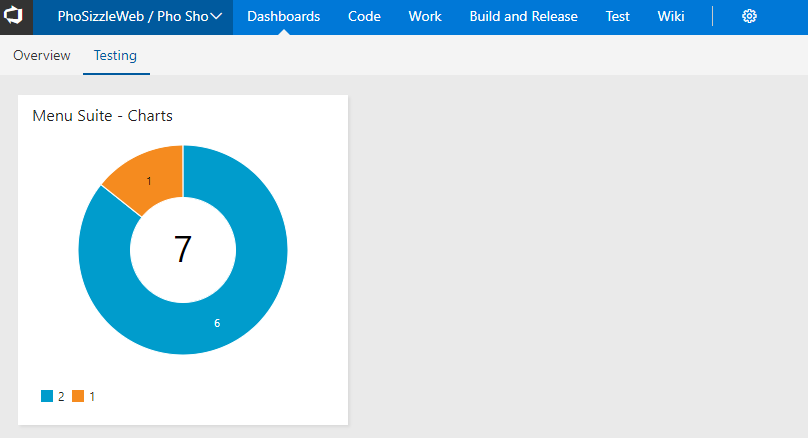
1. Click Add. Click Close on the widget menu to the right on your screen.
2. Click the blue check mark icon on the bottom right of the page.



1. Click on the middle of the Widget to configure it. Group by Priority and use a Pie chart. Hit Save.



1. Your Dashboard should look like this:

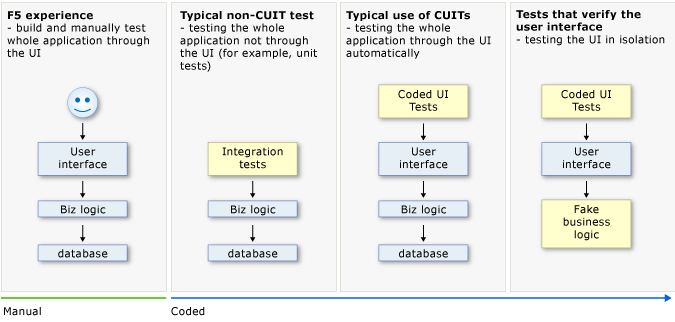


**Exercise 7: Automated Testing with Coded UI Testing: Overview (25 mins)**

\***Prerequisite Note:** You cannot complete this Exercise without Visual Studio **Enterprise**. This cannot be done with Visual Studio Community or Professional.

Automated tests that drive your application through its user interface (UI) are known as coded UI tests (CUITs). These tests include functional testing of the UI controls. They let you verify that the whole application, including its user interface, is functioning correctly. Coded UI Tests are particularly useful where there is validation or other logic in the user interface, for example in a web page. They are also frequently used to automate an existing manual test.

As shown in the following illustration, a typical development experience might be one where, initially, you simply build your application (F5) and click through the UI controls to verify that things are working correctly. You then might decide to create a coded test so that you don't need to continue to test the application manually. Depending on the functionality being tested in your application, you can write code for either a functional test, or for an integration test that might or might not include testing at the UI level. If you simply want to directly access some business logic, you might code a unit test. However, under certain circumstances, it can be beneficial to include testing of the various UI controls in your application. A coded UI test can automate the initial (F5) scenario, verifying that code churn does not impact the functionality of your application.



There are two ways to generate the code for a new coded UI test:

* The first and default option is to use the Coded UI Test Builder, which allows you to generate a test code by manually walking through a test scenario
* The second option to generate the code is using an existing action recording and it will be explored in a subsequent exercise.

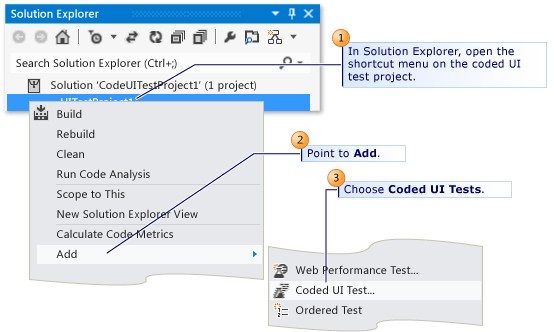
**Tasks**

1. Coded UI Testing
2. Coded UI tests must be contained in a coded UI test project. If you don't already have a coded UI test project, create one. In Solution Explorer, on the shortcut menu of the solution, choose Add, New Project and then select either Visual Basic or Visual C#. Next, choose Test, Coded UI Test.

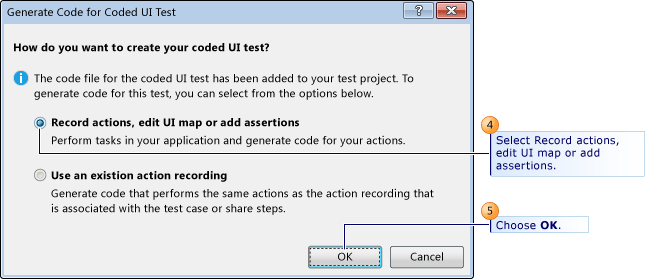
*If you don't see the Coded UI Test project template:*

*You might be using a version of Visual Studio that does not support coded UI tests. To create coded UI tests, you must use Visual Studio Enterprise.*

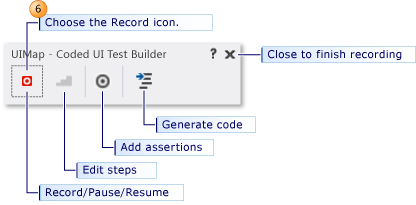
1. If you just created a Coded UI project, the first CUIT file is added automatically. To add another test file, open the shortcut menu on the coded UI test project, point to Add, and then choose Coded UI Test.

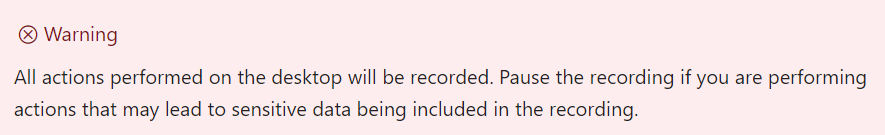


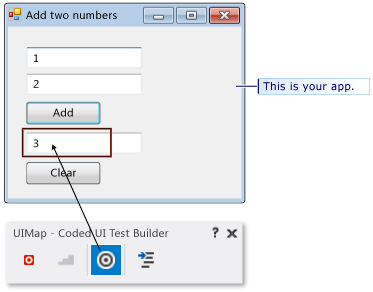
1. In the Generate Code for Coded UI Test dialog box, choose **Record actions, edit UI map or add assertions.**



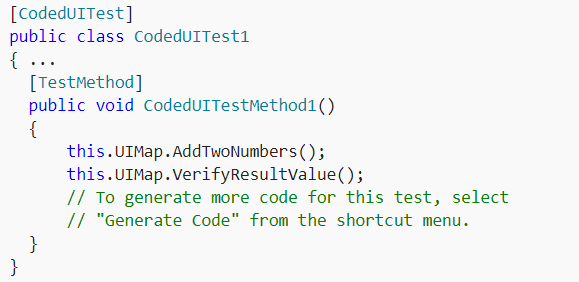
1. The Coded UI Test Builder appears, and Visual Studio is minimized. Choose the “Record Icon”.



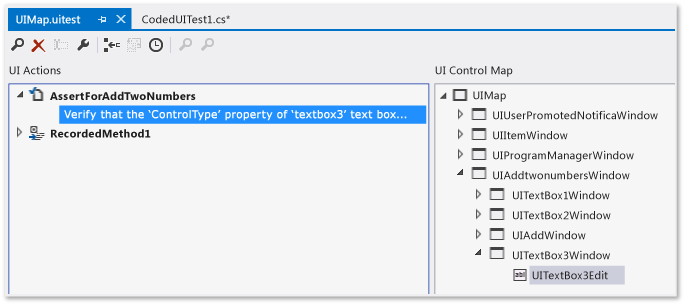
1. To start recording, choose the Record icon. Perform the actions that you want to test in your application, including starting the application if that is required.
   1. For example, if you are testing a web application, you might start a browser, navigate to the web site, and log in to the application.
   2. To pause recording, for example if you have to deal with incoming mail, choose Pause.
   3. To delete actions that you recorded by mistake, choose Edit Actions.
   4. To generate code that will replicate your actions, choose the Generate Code icon and type a name and description for your coded UI test method.
   5. 
2. Choose Add Assertions in the Coded UI Test Builder, and then choose a UI control in your running application. In the list of properties that appears, select a property, for example, Text in a text box. On the shortcut menu, choose Add Assertion. In the dialog box, select the comparison operator, the comparison value, and the error message.
3. Close the assertion window and choose Generate Code.



1. Alternate between recording actions and verifying values. Generate code at the end of each sequence of actions or verifications. If you want, you will be able to insert new actions and verifications later.
2. To view the generated code, close the UI Test Builder window. In the code, you can see the names that you gave to each step. The code is in the CUIT file that you created:



1. Place the cursor at the appropriate point in the test method and then, on the shortcut menu, choose Generate Code for Coded UI Test. New code will be inserted at that point.
2. Open UIMap.uitest. This file opens in the Coded UI Test Editor, where you can edit any sequence of actions that you recorded as well as edit your assertions.



1. Use Test Explorer, or open the shortcut menu in the test method, and then choose Run Tests.

**Exercise 8: Load Testing: Overview (15 mins)**

\***Prerequisite Note:** You cannot complete this Exercise without Visual Studio **Enterprise**. This cannot be done with Visual Studio Community or Professional.

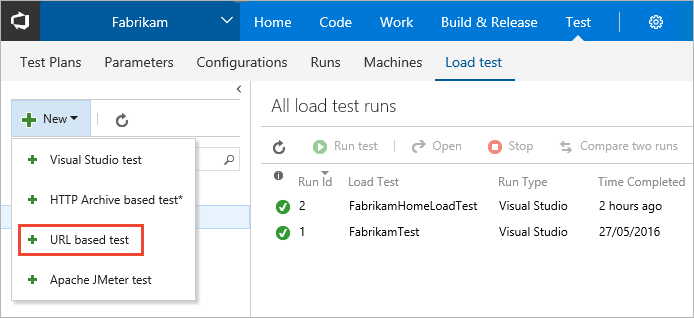
Check your app or web site's performance before you launch it or deploy updates to production. Find problems before your customers do. Start running cloud-based load tests in almost no time with Visual Studio and Visual Studio Team Services (VSTS).

You can run a load test on your web app or site directly using Visual Studio Team Services (VSTS).

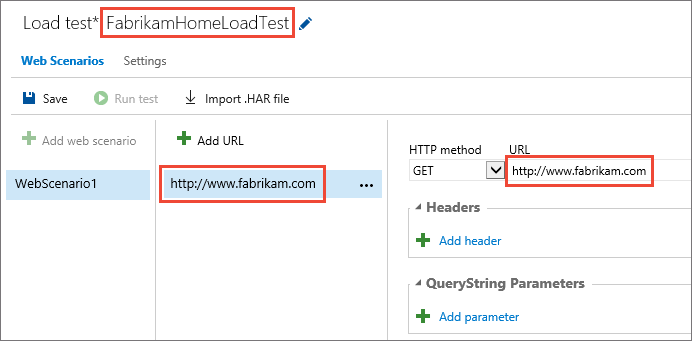
If you want to use Visual Studio Enterprise to do this in the future, please use this URL for reference: <https://docs.microsoft.com/en-us/vsts/load-test/getting-started-with-performance-testing>

**Tasks**

1. Load Testing
2. Go to the Load test hub, open the + New menu and choose URL based test.

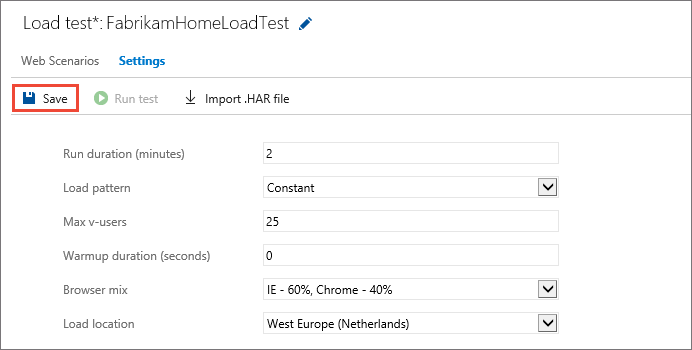


1. Type a name for the load test, and enter the URL you want to test in the center column and in the details pane on the right. For a simple load test, leave the HTTP method set to GET.

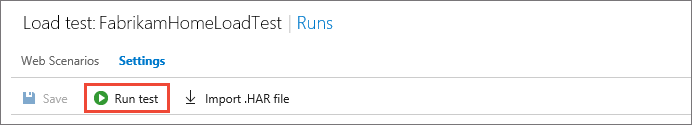


You can add multiple URLs and select the method for each one, such as POST or PUT. You can also add headers and querystring values if you need to send these as part of the request. The URL Load Test accesses each of these URLs multiple times using the parameters you specify, and records the results.

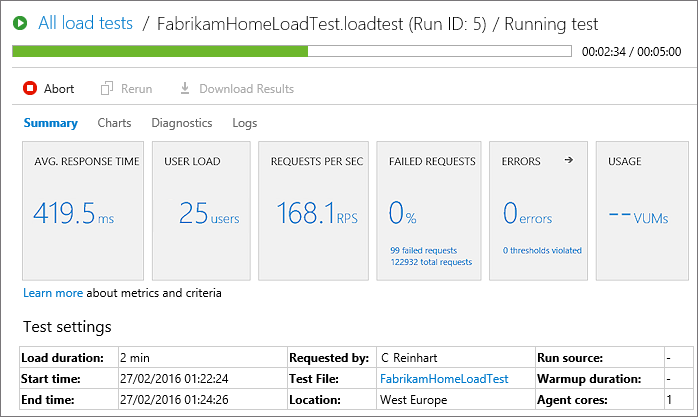
1. Open the Settings tab. Here you can change the parameters of the test such as the duration, load pattern, number of users, and more. To run the test near to your users, select a Load location. Then choose Save.



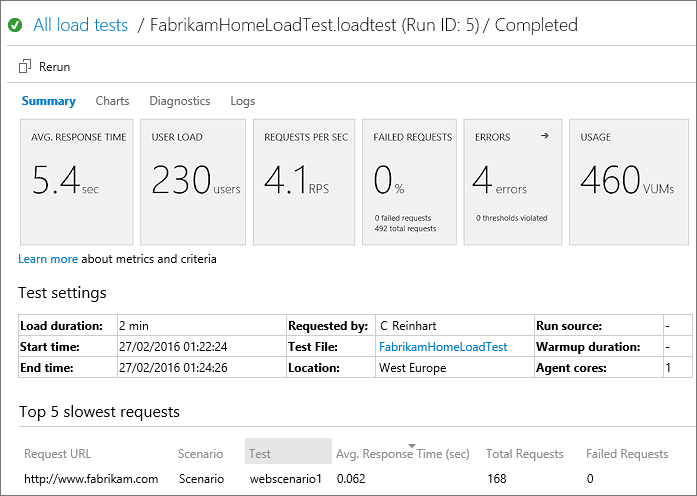
1. When you have set up all the URLs and parameters for your test, start it by choosing Run test.



1. As the test runs, you see live information about the progress of the test. You can stop the test by using the Abort link on the toolbar.

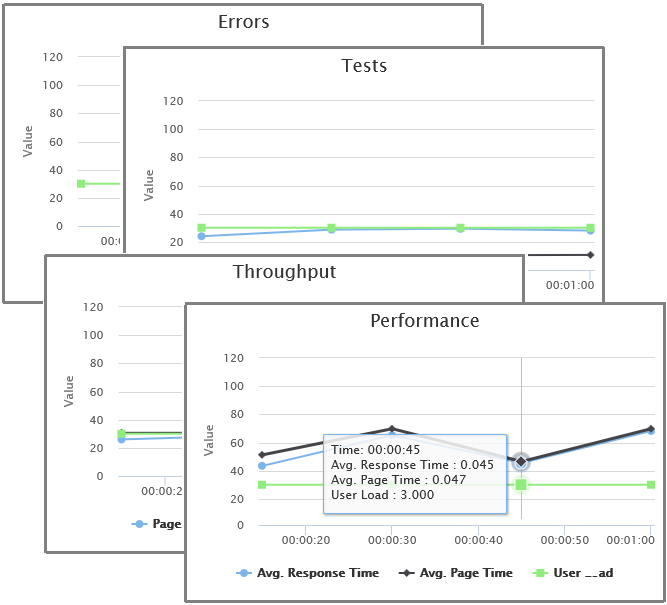


1. When your test is done, look at the results to see how well your app performed. For example, you can see an overview of your app's performance in the Summary tab. This tab shows all of the main metrics such as average response time, user load, requests per second, failed requests, any errors that might have occurred, and test usage.



The lower section of the Summary tab shows the settings used for the test, and details of the five slowest requests during the test. If there are any transaction tests, the tab will also show the five slowest of these. Use the down arrow icon above a column to sort the list based on the contents of that column.

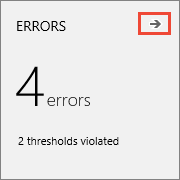
1. Open the Charts tab to see a graphical representation of the test results over time. The charts show the average performance, throughput, errors, and the results of each test request. Hover your mouse pointer over a chart to see more details.



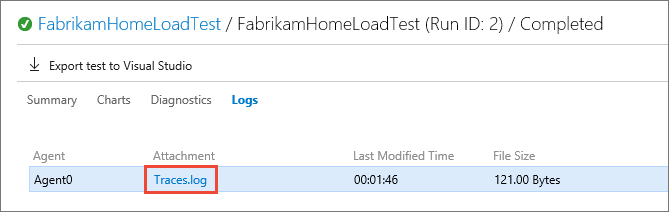
1. Open the Diagnostics tab to see detailed information such as a list of errors and status messages.



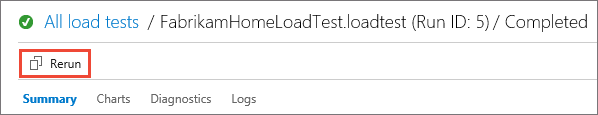
1. You can also use the arrow icon in the Errors section of the Summary tab to go directly to the Diagnostics tab.



1. Open the Logs tab to see a list of test runs. Choose the link in the Attachment column to download the detailed log as a text file.



1. To run the same test again, choose Rerun.



**Final Notes: What's New in Testing? What's on the Roadmap? (5 mins)**

If you wanted to find out what features currently exist to see if it’s implemented, you could go to the current VSTS features: <https://docs.microsoft.com/en-us/vsts/release-notes/#current-features>

The upcoming VSTS roadmap is here: <https://docs.microsoft.com/en-us/vsts/release-notes/>

* If you were to look under the **Area** column, you could Ctrl+F (find) on the page any that were for Test

If they wanted to request new features, they would go here: <https://visualstudio.uservoice.com/forums/330519-visual-studio-team-services>