**Summary**

Great work. In this module, you and the team added unit tests and code coverage to the build pipeline. You set up dashboard widgets so that others can easily track your improvements. You also fixed a build break before it could reach anyone else.

Defining build tasks locally first helps you understand and verify the process before you add build tasks to your pipeline.

Remember, the process you followed was specific to .NET Core applications. The tools and tasks that you use depend on the programming language and frameworks that you use to build your applications.

Let's check in with the team one last time. They're having their weekly team meeting, and Mara wants to show everyone the latest updates.

**Tim:** I'm glad to see you're making good progress stabilizing our builds. But that sure looks like a lot of code!

**Andy:** Yes, it's a fair amount. But it's all things we do anyway. And now that parts are automated, we only need to define them one time. And because it's code, we have a common vocabulary for defining our build process and a consistent place to keep it all.

**Mara:** Thank you all for giving it a try. Now, what's next on the list?

**Learn more**

In this module, you used the DotNetCoreCLI@2 task to run unit tests through the dotnet test command. If you use [Visual Studio](https://docs.microsoft.com/en-us/visualstudio/test/getting-started-with-unit-testing?view=vs-2019) to run your tests, you can use the [Visual Studio Test task](https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/test/vstest?view=azure-devops) in your build pipeline.

If you're interested in unit testing .NET Core applications, here are some additional resources:

* [Unit test your code](https://docs.microsoft.com/en-us/visualstudio/test/unit-test-your-code?view=vs-2019)
* [Unit testing C# with NUnit and .NET Core](https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-nunit)
* [Build, test, and deploy .NET Core apps in Azure Pipelines](https://docs.microsoft.com/en-us/azure/devops/pipelines/languages/dotnet-core?view=azure-devops&tabs=yaml)

Here's more information about how to analyze your test results:

* [Configure the Test Results Trend (Advanced) widget](https://docs.microsoft.com/en-us/azure/devops/report/dashboards/configure-test-results-trend?view=azure-devops)
* [Analyze test results](https://docs.microsoft.com/en-us/azure/devops/pipelines/test/test-analytics?view=azure-devops)