

Good Collaboration: Bringing Developers and Testers Closer

Jim Holmes

Testing Tools Evangelist
Telerik

I fully admit it: I'm extraordinarily biased to having developers and testers co-located in the same room, or even at the same desk. I've seen the great collaboration that ensues as

testers and developers bring their different skillsets, experience, and views together while working together on project tasks. While I'm biased in that direction, I've also led and been a part of teams working in different states and on different sides of the globe—and we were highly successful at our efforts thanks to working hard at establishing a great collaborative environment.

Collaboration isn't just a buzzword, it's a central pillar to a great project. Without it, every action becomes more painful—you suffer large amounts of rework due to a lack of good flowing communication and numerous misunderstandings. With it you find yourself speeding through your work and delivering software that's much more in tune with what your customers expect.

So much of collaboration is around simplicity and basic fundamentals: get process and tools out of your team's way; let them figure out the best way they can solve problems and deliver value. Your team needs to be free to use the frameworks and tools they find most productive. They need to be able to sit together and share ideas on how work items should be designed, executed, and tested. Your teams need to ensure they're able to keep their work queues short and easily dive in to help others where needed.

Test Studio can be a great part of helping boost collaboration between your testers and developers—we add great value to the points I just listed above.

Use the tools and frameworks your team finds most productive

We don't get in the way of your source control system, for example. Since Test Studio is completely file-based, it's a snap to add testing projects to your Github or Subversion repository. Test Studio's UI, as well as our Visual Studio plugin, enables you to integrate directly with TFS if that's your system of choice.

If you're currently a TeamPulse user you can tie Test Studio straight in with your existing process flow—and if you're not we stay out of your way and let you use whatever process works best for your team. Test Studio doesn't force you into a set process. Is your team using some form of Acceptance Test Driven Development? Great! Test Studio will fit right in that. Are you in a shop where the majority of your testing happens in a black-box environment after initial development is complete? We'll help you out in that case, too.

In the same fashion, Test Studio lets you work with whatever testing toolset you have in place already. Some of our customers are using our free Testing Framework with NUnit, others with MbUnit, and some with MSTest. Our automation framework will happily execute its tests inside any of those testing tools—which furthermore means you're free to run those tests like Team City, Hudson, Cruise Control, Team Foundation Build, or any number of other widely used automation and integration servers.

Let your developers and testers write tests however works best for them

Every team approaches testing and development in different ways. If your team is used to writing acceptance tests with Fitnesse, MSpec, or some other tool then we'll bring Test Studio's great advantages to those environments by enabling your testers and developers to write WebAii tests right inside those tools.

If there's a formal test drafting process that starts out with a group of people writing manual test cases for later execution then you can bring those test cases straight in to Test Studio where you can now easily track your manual cases' execution history right alongside your automated ones. Those huge Excel files holding all your manual test cases you've written up over the months of your project's lifecycle? We'll help you bring them in to Test Studio where the whole team can easily work with them!

Keep the "ready to test" work queue as short as possible – and enable the entire team to help

Test Studio helps your entire team keep up with test coverage and execution. Anyone's able to write coded tests or record new tests for your project, regardless of whether they're using the Test Studio or Visual Studio UI. Similar-

ly, anyone on the team is able to run tests, automated or manual, at any time, with a few simple clicks. (Your team will still need to do the brainwork of figuring out exactly what to test!)

Simplicity for executing tests, visibility into your quality status

If much of collaboration is about getting roadblocks out of the way, then one of the biggest to remove is friction around running your tests or seeing their status. You absolutely need a simple, painless way to run tests you want immediately, schedule automated runs, and access a complete history of all runs down to granular test results.

Test Studio gives you all this in spades. It's simple for a developer and tester pair to work through a user story, and then run all the tests in the project to ensure there have been no regressions.

Your entire team can take advantage of Test Studio's Scheduling and Remote Execution servers to split out long-running suites into smaller pieces that can run concurrently. This helps shorten the feedback cycle tremendously.

Finally, anyone on the team can get a historical view of the test suite's quality by using the test run reporting views. Your entire team has quick access to understand what's passing, what's failing, and what tests need to be investigated.

The Bottom Line: Help Where Needed, Stay Out of the Way Where Not Needed!

Test Studio can fit right in to your existing projects and processes without injecting its own set of requirements. You don't need to spend time adapting your environment to fit the tool. We'll adapt to fit yours. We give you a number of ways to solve common problems and leave it up to your team to figure out what's best in your specific environment.

Get started with Test Studio today.
[Download a free 30-day trial with priority support.](#)

[Jim Holmes](#)
Testing Tools Evangelist
Telerik