Guestopinion

There is No Silver Bullet

by Juval Löwy

On February 13, I witnessed Visual Studio .NET's (VS.NET) formal debut at VSLive! in San Francisco. The launch was a two hour grand event, with thousands of attendees, presided over by Bill Gates himself. With the fanfare and pageantry typical of a Microsoft presentation, it offered two demonstrations of the power of .NET. First, Gates introduced an "Iron Developer" contest (a takeoff on the popular "Iron Chef" television show), then brought out two high school kids to demonstrate the VS.NET app they'd created for a school project.

The high school developers demonstrated a Web-based VS.NET application that tracks homework assignments, grades, and test scores, and allows students and teachers to review these details online or through a wireless device. The application seemed to have a decent set of features and capabilities, and the young programmers proudly announced it took them just five weeks to develop it from start to finish.

In the Iron Developer contest, two Microsoft senior developers were given a basic e-commerce sales management application and instructed to add an impressive set of features. After a frantic hour backstage, they were called back onstage to demonstrate their solutions. One used ASP.NET and the other used Windows Forms, and both achieved a mind boggling set of features, from Web services support and mobile devices connectivity, to graphics reports and fancy data grids. The crowd was duly impressed, and it appeared as if VS.NET was the cure for the issues that continually plague the software industry such as tight deadlines, low developer skill sets, restrictive budgets, and poor quality.

In my opinion, the Microsoft presentation (as well as its ongoing marketing campaign) is merely half the truth and even misleading. Don't get me wrong: I think VS.NET with its application frameworks (ASP.NET, ADO.NET, Windows Forms), native Web services support, integrated visual development environment, and rich set of class libraries is a superb development platform—the best I've ever come across. The problem is, developer productivity is only partially related to technology, because it's also heavily dependent on the development process and management maturity.

Consider a well-managed project, with repeatable and mature processes, tasked with producing a 1.0 version of an application, starting from scratch. In such a project, coding is only about 30 percent of the time and effort. The rest is spent on requirement gathering and analysis, architecture and top-level design, detailed design, unit testing, system testing and integration, not to mention quality control and assurance, build environment, source control, configuration management, and documentation. Without any of these key process areas, you're unlikely to be able to implement all your features and produce a quality product on budget or schedule.

Assuming .NET development is twice as productive as current tools (even after factoring out learning curves), you get only 15 percent overall periodicity improvement, because you still need to do all the other non-technology related activities. This, of course, is nothing new. In his classic book *The Mythical Man-Month*, Frederick P. Brooks, Jr. states there is no silver bullet—no rapid development tool can replace sound process and overall good development skills. Brooks wrote this in the '70s, but his conclusion is as valid today as it ever was. I believe rapid development tools such as VS.NET offer only the potential for—not a guarantee of—productivity improvements. Rapid development tools will make bad developers worse, because they allow them to produce bad code faster. Rapid development tools leave average developers where they are, but they allow good developers to excel, because they aren't slowed down by plumbing or mundane code. It's important to approach VS.NET with the right perspective: A mature process (coupled with a tool such as VS.NET) is the only way to productive development. A mature process is what enables developers to be productive, taking advantage of the best technology.

About the Author

Juval Löwy is a software architect and the principal of IDesign, a consulting and training company focused on .NET design and migration. Juval is a frequent speaker at international software development conferences and a regular contributor to *Visual Studio Magazine* and *.NET Magazine.* He's now working on his second .NET book regarding .NET component development. Juval chairs the program committee of the .NET California Bay Area User Group. Contact him at www.idesign.net.