

Eclipse is an open source community focused on building an open development platform comprising extensible frameworks, tools, and runtimes for building, deploying, and managing software across the life cycle. Now is a great time for developers not yet using Eclipse in their software development processes to research how Eclipse can aid in product development. Developers should take time to consider the ways in which an Eclipse-based test environment can make projects more productive.

With its focus on providing an open platform of extensible frameworks, tools, and runtimes for building and managing software, Eclipse has proven to be popular with independent developers and companies that want to create applications with a broader reach than traditional programs. Using Eclipse, developers can create powerful applications that run on a variety of Operating Systems (OSs) and integrate easily with existing systems and future applications. At the same time, Eclipse remains on the bleeding edge for companies creating exciting, useful commercial products.

The need for applications to work seamlessly in different computing environments is especially important in the product testing process. Testing software built on Eclipse lends itself well to this requirement since testing processes are distributed across multiple quality assurance teams and among different companies.

Eclipse and plug-ins from the open source community have contributed to the need for including testing in the development process. Thus, it is a bit ironic that a company would build a commercial application exclusively for testers that leverages the open source framework of Eclipse. Developers using Eclipse can now leverage another Eclipse tool to further speed development. Testers can gain access to the power and benefits of programs associated with open environments. As a result, testers can devote their attention to developing or running tests and understanding test results instead of building special test applications.

Eclipse-based developments today are changing the software world in ways that no one could have predicted. As an open development platform for building software, Eclipse was originally developed by IBM and then handed over in 2004 to the nonprofit Eclipse Foundation. The cornucopia of new applications since then has, in essence, helped place Eclipse squarely in the forefront of software development.

### Suited for multiple environments

Companies are increasingly choosing Eclipse to assist in product development for several reasons. Most prominently, Eclipse performs well on many platforms, which is critical because companies that use test automation software rely on multiple computing environments. When developing Eclipse, IBM

concentrated on several widely used platforms representing the spectrum of OSs. Today, Eclipse development supports common environments, including Windows, Linux, Solaris, Mac OS X, and various Real-Time Operating Systems (RTOSs). Any company that develops an application on Eclipse can run that application on those environments. Figure 1 depicts a graphical representation of the five most valuable platform attributes relative to a project.

Because Eclipse has its own platform-dependent user interface component, it can closely approximate the look and feel of the native OS. In other words, since Eclipse has its own Windows component built on top of Windows controls, an Eclipse-based application will look like a native Windows application in that environment. The same logic applies to running applications on Linux, Solaris, Mac OS, and various RTOSs. The native look and feel of the application is a major improvement compared to the way Java applications often appear on different OSs. For instance, Java applications can look foreign on a Mac.

### Saving time, resources

Programmers can reuse plug-ins contributed by the open source community to the Eclipse platform. This speeds up development time because programmers do not have to rewrite what has already been accomplished. For example, Eclipse provides built-in

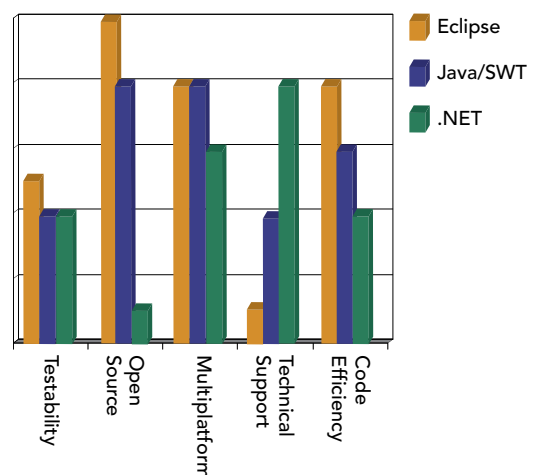


Figure 1

support for automatic software upgrades, hot fixes, and patches. This means that when someone develops a new product based on Eclipse, they do not have to spend programming resources to develop these features.

Since Eclipse contains the basic building blocks of any platform, it can save development teams a tremendous amount of time. For example, using a propriety development platform, developing a product can take several years to complete because developers need to write everything from scratch. Instead, using Eclipse, products often can be developed in less than a year.

Of course, this does not mean that Eclipse development is always easy. The lack of comprehensive Eclipse documentation means that developers may have to dig deep into the code base, combing through millions of lines of code. The biggest challenge occurs during the early phase of switching to Eclipse, but once developers grasp how Eclipse works, they can quickly become familiar with the code and libraries.

In addition, developers must remember that Eclipse might not perform a function the way they had intended. Consequently, developers may have to rebuild their code and then make some compromises, whereas custom building the code may have prevented such a pitfall. Also, because Eclipse is an open source environment, the code can change, and features or functions that developers had built may not work in the exact same way in the next version of Eclipse. Figure 2 provides a representation of the challenges faced by companies relying on open source code and the risk it introduces during the life cycle. However, if acknowledged, most companies can neutralize this risk with proper diligence and testing.

Another essential factor that prompts many companies to use Eclipse is that it is free. Using proprietary environments is always more expensive, both for the company developing the software and ultimately for customers purchasing the software. With Eclipse, application development can begin right away and modifications can be made within a short period of time. Companies can then increase quickness and efficiency, which translates into lower overall development costs.

### A customized experience

Companies can opt to use Eclipse to develop applications in plug-in mode and Rich Client Platform (RCP) mode. In plug-in mode, users can incorporate new applications into their existing Eclipse installations, which may contain applications from other vendors. This means users can benefit from many of these applications, providing a common look, feel, and usage model. If written well, these applications can integrate with each other and offer users even better value.

All Eclipse applications, whether in plug-in or RCP mode, look like native applications. Companies that want their customers to have a

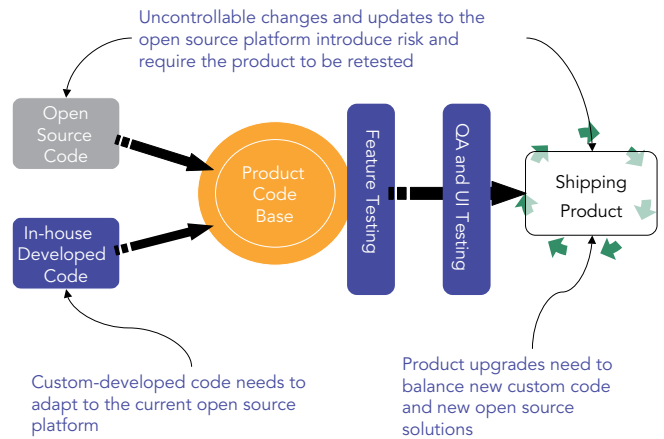


Figure 2

completely customized experience benefit from RCP. For example, a full-fledged Eclipse environment can be daunting to beginning Eclipse users. RCP allows companies to tailor their products to their customers' needs, ultimately simplifying the application.

### Automated testing under control

Many software products based on Eclipse are becoming popular with customers. Network equipment manufacturers and service providers, for example, use Fanfare's iTest to automatically test their applications, while telecommunications companies use the product to run the acceptance process of their hardware.

After capturing data during a usage cycle, testers can convert the data into a test and then run the test over and over again. This means that testers do not need to know anything about programming to write an automated test; they can rely on Eclipse-based products to do the automated testing. Eclipse also accommodates multithreading technology, which allows customers to perform complex system tests by executing simultaneous commands to a number of different devices in their network to simulate real-life network tests.

Currently, Eclipse development at Fanfare focuses on testing software, but future development efforts will likely include Eclipse to extend solutions into other areas. The reason is simple: Eclipse is versatile and proven enough to support a range of development efforts. With its extensible frameworks, tools, and libraries, Eclipse can help deliver powerful solutions faster than ever thought possible. ▲ UUT



*Pawan Singh is the director of software development at Fanfare. He holds BS and MS degrees in Computer Science from the Indian Institute of Technology and Queen's University in Kingston, Ontario, respectively.*

**Fanfare**  
650-776-3958  
psingh@fnfr.com  
www.fanfaresoftware.com