HTML5 or Silverlight?

What's in Store for the Future and What You Should Do about It

Contents

Introduction

In a rare moment of frustration, Albert Einstein said that perfection of means and confusion of goals seemed to characterize the age. One might think these were the words of a software developer, not a physicist.

As development platforms continue to evolve and improve, the forest is sometimes lost in the trees. Such confusion can be seen in the ongoing debate regarding Microsoft Silverlight and HTML5.

Some think that Microsoft is going to abandon Silverlight as momentum for HTML5 continues to grow. This does not appear to be true. While Microsoft has shifted its strategy for Silverlight, no longer touting it as the vehicle for delivering a cross-platform runtime, it continues to push it as the development platform for Windows Phone as well as for some media and line-of-business applications. Silverlight is not going away. In fact, Silverlight 5 in its final form will be available this year, and the tooling support for which Microsoft is legendary will back it up.

While HTML5 is a draft standard, its ultimate role as the dominant cross-platform solution is a *fait accompli.* Even Microsoft acknowledges this, calling HTML "the only true cross-platform solution for everything."[1]

Basically, Silverlight and HTML5 both have their place and purpose, as a close look at the similarities and differences of the two tools will reveal.

Similarities on the Surface

At first glance, HTML5 and Silverlight are similar in a number of ways. "The similarities are tied to ease of deployment, the richness of the user interface, and the interaction model," says Daniel Jebaraj, the vice president at Syncfusion who leads product development.

One of the concerns with the desktop, especially for Windows developers working within the Windows environment, is deployment. This is an issue for those working in mid-sized to large companies. "The deployment footprint for desktop applications is quite troubling for them because they have to ensure that a certain version of the runtime is available, and they have to get updates to every machine," says Jebaraj. Therefore, deployment has been a pain point for richer desktop applications.

In some cases, the business reasons for sticking to desktop applications will continue: better performance and better use of native hardware—simply getting things done in an easier, more seamless manner. Plus greater programmer knowledge exists for development scenarios on the desktop platform. These reasons can override the need for easy deployment.

For customers to whom deployment is a central priority, pure Web solutions may be preferable, although not all customers will be happy with the richness delivered vis-à-vis what is available in the desktop application. Such concerns must be balanced as alternative solutions are explored.

---

[1] Foley, Mary Jo (October 29, 2010) "Microsoft: Our strategy with Silverlight has shifted," http://www.zdnet.com/blog/microsoft/microsoft-our-strategy-with-silverlight-has-shifted/7834.

In general, Adobe Air or Silverlight is a good solution. Within the Microsoft stack, Silverlight is a *very* good solution, because even though there is a runtime, it is structured for seamless deployment and it is easy to get users updated. "Essentially, Microsoft takes care of the updating experience," says Jebaraj. "You don't have to worry about getting the application down to end-user machines."

HTML5 provides a similar deployment scenario; however, it has hidden traps because it relies on a browser. "The team using an HTML5 solution has to be reasonably sure that their customers will have access to the latest browsers and will install them," notes Jebaraj. In addition, he says that HTML5 has broad support: IE9, Chrome, and Firefox support it; IE10 will work better with it. "All mobile clients will support it in the future, even if they do not now."

In short, while a small deployment difference exists between Silverlight and HTML5, the deployment model is basically seamless in both.

In terms of richness of user interface (UI), Jebaraj says Silverlight has some advantages. "To get a rich UI done quickly, Silverlight is probably a better solution," he says. Nonetheless, HTML5 is catching up to Silverlight along this parameter. Soon it will have much more prepackaged content available to facilitate the building of rich UI environments.

The two tools are also comparable in terms of their interaction models. Neither demands that one wait for a page refresh, and working with either is similar to the way one works with a desktop application.

On Closer Examination

The strong functional similarities between HTML5 and Silverlight tend to dissolve on closer examination. First, Silverlight is more suited for intranet applications that have relative control over the deployment environment than for true Web-based deployment.

"If you look a little deeper at the deployment scenario in Silverlight, it is still not a true end-user solution," says Jebaraj. So if the developer's purpose is to have application users download Silverlight and run it on their machines, then the developer needs to have a good picture of the customer. *Is the customer going to have a system that can run Silverlight? Is it something that will be allowed?*

For example, if a user goes to amazon.com, it's probably not a good experience to be prompted to download the Silverlight client. For Web site use, the more seamless the experience, the better.

However, in cases of an intranet solution—where developers have more control over the machines and know that they are Windows machines—they may not have quite the degree of control needed for a desktop application, but they know that those machines are capable of running Silverlight. That provides the developer with a great deal of flexibility.

"Now he or she can choose to go with Silverlight; when compared to HTML5, it is definitely a more productive development experience," says Jebaraj.

Microsoft has excellent tools that make it easy to build and deploy applications with Silverlight. HTML5 requires a bit more work, while Silverlight is more structured and the tools, according to Jebaraj, are in a class of their own. "If you are confident that your deployment environment is relatively familiar to you, perhaps a close band

of customers to whom you can say, 'these are the minimum requirements,' then Silverlight is more suitable," he says. This is further supported by the quality of Silverlight tools, which empower the developer to build Silverlight applications in a quick, drag-and-drop manner. Silverlight also makes UI development, and most other development, highly productive by using the built-in control abstraction model and native controls that Microsoft has provided.

Lingua Franca or Straining to Understand?

Programming language is another consideration. C# (Silverlight) is easier to work with and debug than JavaScript (HTML5). "It goes back to the tooling and the nature of the language," says Jebaraj. "JavaScript is a different beast to work with; even experienced JavaScript programmers know that it is a bit harder to understand."

Silverlight Limitations and Other Considerations

Silverlight is limited if mobile deployment is required. Currently, Silverlight is only supported on Windows Phone. It may be supported on other platforms in the future, but this is not certain. "It's not something that is likely to happen in the short term," says Jebaraj. "Currently, to develop a Silverlight application that mobile clients can use, the Windows Phone device must be mandated."

If developers don't have control over their mobile clients yet they want to support them, HTML5 is a viable option. Already supported by iOS, Android 3, and with Windows committing to support it in IE10, HTML5 is the clear choice right now.

However, Silverlight may offer better performance than HTML5. Over the last few months, Microsoft implemented a hardware solution for Silverlight 5, so it may offer a slight performance advantage over HTML5 on newer machines. "But there is no reason that HTML5 won't catch up," says Jebaraj. "IE9, for example, does a lot with HTML5, so this performance difference is more transitory than decisive. It isn't much of a difference."

Notably, HTML5 is a standards-based environment, something of great concern to some developers but of little concern to others. If a developer is working on the Microsoft stack, he or she will be aware of and have an appreciation for standards; however, this is mitigated by the understanding that Microsoft is going to be around for awhile, so there isn't a reluctance to adopt proprietary technologies from Microsoft.

In many cases there is a push to adhere to standards, so that if things change, the migration path is easier. "HTML5 is standards-based and about as global as it gets," says Jebaraj. "The path forward is very clear with HTML5; with Silverlight, less so."

Silverlight has a considerable advantage over HTML5 in that 90 to 95 percent of code can be shared with desktop applications. "If you have a full-fledged desktop application and a solution to get on the Web," says Jebaraj, "it's easier with the Silverlight model."

In HTML5, developers can keep the UI separate and have a business layer; but, there is a great deal of UI code that has to be written on the two platforms. So it takes more effort and allows almost no code sharing.

Means to an Informed End

In the end, the tools must be selected with a clear goal in mind to avoid the confusion Einstein identified. According to Jebaraj, choosing the right path based on one's needs will minimize the likelihood of expensive mistakes. For example, developers may be comfortable with WPF and Silverlight and so choose that path, but after exploring what they want to do with the application, they may hope to launch a mobile client within six months to a year. What's more, they want people with multiple devices to have access.

"At that point, it's okay if they go with the Silverlight solution up front, as long as they are prepared to eventually go to HTML5 or write native applications for each mobile platform," says Jebaraj. "If not, this will be a very expensive choice."

The flip side holds as well: there may be a push to go with HTML5 because everyone is saying, "this is the future." So a decision is made to go in that direction, even though the customer has Visual Studio for all its developers and is licensed to use the latest and greatest Microsoft tools.

"If they have good control over the deployment environment, and if they're not going to put in Linux or Mac devices in the next six months, then they're probably making an expensive mistake," says Jebaraj. "They could have gone with Silverlight, which would have allowed them to get to market faster, as well as use their current skill set and toolset more effectively."

The bottom line: as long as developers make an informed choice, either Silverlight or HTML5 is fine.

Applying the Best

Microsoft has no control over the mobile market right now since they have to support HTML5 strongly. They're not dictating terms, and in Jebaraj's opinion, it's questionable whether they would if they could. "Microsoft's strategy is to apply the best tools and platform possible, regardless of whether it is proprietary," he says.

According to Microsoft, "On the web, the purpose of Silverlight has never been to replace HTML; it's to do the things that HTML (and other technologies) couldn't in a way that was easy for developers to tap into. Microsoft remains committed to using Silverlight to extend the Web by enabling scenarios that HTML doesn't cover. From simple islands of richness in HTML pages to full desktop-like applications in the browser and beyond, Silverlight enables applications that deliver the kinds of rich experiences users want."[2]

These words underscore what is apparent to us: Silverlight and HTML5 will both thrive in the foreseeable future.

---

[2] Silverlight Team Blog (September 1, 2010) "The Future of Silverlight," Microsoft Corporation, http://team.silverlight.net/announcement/the-future-of-silverlight/.

About Syncfusion

Syncfusion is the enterprise technology partner for developers on every Microsoft platform, delivering the broadest range of .NET components and controls coupled with a service-oriented approach throughout the entire application lifecycle. Offering the fastest high-performance Silverlight and WPF grid on the market and quarterly product updates – Syncfusion is trusted by developers worldwide for use in mission critical applications. The company's flexible licensing and 24/7 support meets the changing needs of enterprises across the globe. Founded in 2001 and headquartered in Research Triangle Park, North Carolina, Syncfusion has more than 7,000 customers including large financial institutions, Fortune 100 companies and global IT consultancies.

For questions about products, pricing, and licensing, please contact sales@syncfusion.com or call 1.888.9DOTNET (US only) or +1 919.481.1974.