

# BEST PRACTICES

December 28, 2006

## Ajax Or Flex?: How To Select RIA Technologies

by **Jeffrey S. Hammond**

with Carey Schwaber, Ron Rogowski, and Jacqueline Stone

### EXECUTIVE SUMMARY

When it comes to rich Internet application (RIA) technologies, application development professionals must choose between two paths: Ajax or Adobe. Ajax is the best bet for experienced Web development shops looking to incrementally evolve existing Web applications. Of course, selecting Ajax also raises the question of whether to go with a homegrown, open source, or commercial Ajax framework. In contrast, Adobe is best for shops looking to completely transform their Web experiences. But to do so, firms must acquire the skills necessary to use Flex, which is the development framework for the Flash runtime environment. Early adopters with time-to-market concerns and comfort with either smaller vendors or open source can select Ajax with confidence. But firms that require support from a larger vendor and full-featured development tools will do better with Adobe.

### TARGET AUDIENCE

Application development and program management professional

### CUSTOMERS DEMAND RIA, SO DEVELOPERS MUST DECIDE HOW TO DELIVER

The benefits are clear: RIAs help users find and manipulate relevant content, complete complex transactions, and consume multimedia in the context of a larger app.<sup>1</sup> The result? Improved customer experience, higher Web conversion rates, and thus improved revenues.

But what's less clear is which tools and technologies development organizations should employ in building RIAs. Today, there are two paths: Ajax and Adobe's Flex. There's no one answer to the question of which to use. App dev professionals should select based on their time-to-market requirements, whether they're looking to evolve or revolutionize their user experience, and whether they prefer tools from large established vendors.

### AJAX IS AN INCREMENTAL PATH TO A BETTER USER EXPERIENCE

App dev organizations that prefer a fast path to RIAs — for developers and users alike — select Ajax for the following reasons:

- **Ajax helps shops meet time-to-market needs.** Developers with existing Web skills find that adopting Ajax involves a shift in design patterns, not a wholesale change in programming language or development tools. When insurance lead provider InsureMe made a decision to use RIA for an insurance quoting application, it chose Ajax.<sup>2</sup> Why? Because going with Ajax had minimal impact on its productivity, and time-to-market was vital on their project.

- **Ajax favors gradual re-factoring of existing Web apps.** Ajax is based on open and standard languages like JavaScript and XML. This makes it easy for developers to integrate Ajax widgets into existing Web applications that are written in HTML, DHTML, or server side JavaScript.<sup>3</sup> This approach is especially useful when an existing application UI would benefit from targeted tweaks, when release cycles are short, or when it is impractical to do extensive A/B testing with users.
- **Ajax can deliver speedy performance.** Vendors of commercial Ajax frameworks like Backbase and JackBe invest considerable resources tuning their Ajax frameworks for speed. The result? Commercial Ajax clients have tiny footprints and download to browsers so quickly that there's no perceptible lag in app performance. Some open source Ajax tool kits share this trait, but others approach the size of Adobe's Flash and have equally long install times.<sup>4</sup>

Developers and architects may salivate over the technical merits of Ajax, but managers should note that using Ajax frameworks also introduces risk. Unmanaged, these risks can increase costs or derail a development effort entirely. For example:

- **It's easy to choose the wrong Ajax framework.** There are dozens of popular Ajax frameworks, and they are not created equal. Developers might want to cut a check based solely on a framework's rich widget set or light-speed performance, but architects will need to validate that the framework supports corporate standards for security, accessibility, service integration, and data access.
- **Finding best-of-breed Ajax development tools is difficult.** Experienced Web developers don't mind writing JavaScript and XML in text editors, but most developers in enterprise IT shops prefer visual designers and WYSIWYG editing tools. When it comes to Ajax, the only such tool support available at present is tied to commercial Ajax frameworks, meaning that the choice of one is necessarily a choice of the other.<sup>5</sup>
- **Cross-browser compatibility issues increase testing costs for Ajax.** Testing costs increase in direct proportion to the number of browser platforms and versions to be supported. However, using a commercial Ajax framework can mitigate these costs, as the framework vendor incurs the cost of cross-browser testing. For example, Software AG certifies its crossvision Application Composer on multiple versions of browsers like Internet Explorer, Mozilla, and Netscape.

### The Ajax Path Itself Forks, With Homegrown, Open Source, And Commercial Alternatives

Just because a development team has chosen to go with Ajax doesn't mean that its tool selection effort is complete. There remains the important decision of how many and what type of Ajax frameworks to use. Today there are three types:

- **Many developers create their own frameworks.** Development teams create their own Ajax widgets when they need customized controls that work in a specific way. For example, a developer at a North American utility company needed a control that would immediately display some discussion forum messages while loading others in the background. In this case, building a custom Ajax widget was the fastest way to get the job done.
- **Other developers build on top of open source frameworks.** Open source communities engage in a constant process of peer review, which quickly separates good design from bad. The result? Successful projects mature quickly, and poor frameworks lose critical mass. When selecting an open source framework, look for active development communities with committed corporate sponsors, as this is an indication of longevity. OpenLaszlo, Dojo, and the Google Web Toolkit (GWT), for example, are supported by corporate sponsors like Laszlo Systems, IBM, AOL, and Google.
- **Commercial frameworks provide a throat to choke.** Commercial Ajax vendors provide support and certification of their products, which is appealing for managers who are uncomfortable taking the open source route. They also focus on providing capabilities that enterprise developers require, including high-performance clients and tooling that makes it easy for developers to connect to enterprise applications.

Just as there are myriad Ajax frameworks, there are also myriad Ajax framework selection criteria. These criteria fall into five buckets: customizability; security and accessibility; widget set; tooling; and cross-browser support.<sup>6</sup> But not all of these criteria should be weighted equally (see Figure 1).

**Figure 1** Prioritized Ajax Selection Criteria

| Criteria                   | Why it matters  |
|----------------------------|---|
| Ease of customization      | Integration efforts, evolving UI design conventions, and self-support capability all require access to source code and customization capabilities |
| Security and accessibility | Difficult to retrofit after an application is deployed  |
| Richness of widget set     | A broad palette of controls improves design options when dealing with complex data or transactions  |
| Development tool support   | Minimize ramp-up time and improve overall cycle time  |
| Cross-browser support      | 88% of enterprise users choose Internet Explorer, Firefox gets the rest   |

## FLEX DELIVERS PRODUCTIVITY IN THE LONG RUN BUT TAKES MORE TIME TO LEARN

Adobe is an attractive RIA solution because it combines powerful development tools (Flex) with a near-universal browser plug-in (Flash) that's designed to simplify RIA deployment. Managers who have concerns about small commercial Ajax providers or support issues with open source Ajax frameworks consider Flex a safe alternative. But there is a downside: To be productive in Flex, developers have to learn two new languages and an extensive runtime framework — something that can take between one and three months to learn.<sup>7</sup> Once this initial hurdle is cleared, Adobe adopters see the following benefits:

- **Widespread adoption of Flash speeds deployment.** The firms we interviewed that had selected Adobe unanimously cited penetration of Flash as their key decision driver.<sup>8</sup> When Flash is already installed, application deployment requires no additional changes to the user's desktop. Unlike Ajax, Flash doesn't require browser-specific code or enablement of JavaScript in the user's browser.
- **Flex works well for large-scale RIAs.** Large-scale user productivity applications are often designed with conversation flows, which require access to business data and rules to drive application context. A US insurer used Flex to replace its conversation-based client server system for independent agents. The result? The new system downloads data to the browser the first time the application is accessed and provides business context and flow while keeping response times to a minimum.
- **Flex is evolving quickly and has a predictable trajectory.** Adobe has invested heavily in Flex, and the result has been sustained improvement from release to release. In contrast, the future of Ajax depends on the whims of a dozen vendors and open source projects that aren't even loosely federated. In the words of one architect we spoke with, Flex is "one support framework to rely on, not dozens to integrate."
- **Flex Builder improves developer productivity and collaboration with designers.** Flex Builder provides developers with a visual, WYSIWYG experience on par with the best commercial tools for Ajax. What sets Flex Builder apart from these tools, though, are features that improve collaboration between developers and designers.<sup>9</sup> Using Flex lets developers and designers at the American Cancer Society rapidly iterate through design changes and preview them with stakeholders within hours.

But Adobe isn't perfect. App dev professionals should consider the following Adobe drawbacks:

- **Flex developers are difficult to locate and command a premium rate.** Flex developers are harder to find and more expensive to retain than Ajax developers: In the United States, they start at \$60,000 and top out around \$120,000.<sup>10</sup> One executive at a North American financial

services firm had to retain talent from South Africa and the Czech Republic. As more developers ramp up on Flex this issue will subside, but for now it makes Flex more expensive in the short-term than Ajax.

- **Flash upgrade issues can defeat the deployment value proposition.** When users aren't on the latest version of Flash, requiring them to upgrade is risky because almost half will abandon the site.<sup>11</sup> Many are unwilling to wait through the lengthy install process and instead choose to do business elsewhere — potentially with the competition.
- **Success can create an unexpected user response.** Flash is a broad graphical canvas, and Flex allows designers to push the boundaries of rich media further than Ajax does. This can lead to a situation where users become distracted by a mix of Flash's rich media and Flex's new controls.<sup>12</sup> IT organizations can avoid this response by introducing rich media gradually and frequently conducting A/B testing with focus groups.<sup>13</sup>

### NEITHER PATH COMPLETELY SATISFIES ALL BASIC NEEDS

When aggressive developers push the boundaries of RIA development, they quickly run into barriers that affect their ability to take full advantage of RIA technology. The firms we interviewed cited several barriers to their RIA development efforts that no tooling available today could overcome (see Figure 2).

**Figure 2** Ajax And Adobe Struggle To Meet All RIA Development Needs

| Limitation                              | Ajax   | Adobe  |
|---|--|--|
| Ease of enterprise integration          | Connecting to multiple data sources and reliable messaging requires developer customization or integrating multiple frameworks | Integrating data sources and Web services is less intuitive than some commercial Ajax frameworks           |
| Client-side installation experience     | Client frameworks have low consumer penetration, and JavaScript may not be enabled   | Upgrade process is not seamless, causing users to abandon at high rates                                    |
| Maturity of ecosystem                   | Fractured ecosystem retards growth and delays improved tool support  | Adobe has not created a partner market for add-on controls, tools, analytics, or vertical offerings        |
| Support from extended development tools | Storyboard and wiring tools are not standard and are poorly integrated with professional design tools                          | Functional testing tool support of Flex is limited, but leading vendors will improve on this front in 2007 |

## RECOMMENDATIONS

### USE AJAX FOR TACTICAL IMPROVEMENTS AND ADOBE FOR STRATEGIC IMPLEMENTATIONS

Which RIA technology is best for your needs? It comes down to the following decision drivers:

- **Choose Ajax when time-to-market is critical and updates are frequent.** Experienced Web developers ramp up on Ajax quickly. Ajax also makes it easy to build RIA capabilities into existing apps with small, frequent releases. Finally, Ajax's customization capabilities make it possible to tune client frameworks to meet specific application footprints; this makes Ajax the better solution for smaller RIA deployments and for deployments where performance is critical.
- **Use Flex for large-scale user productivity apps.** Although Flash's client footprint is larger than Ajax frameworks, its market penetration makes it the preferred deployment platform for larger, comprehensive RIAs. Adobe is a safe choice for those who place a high value on reliable support and product continuity, concerns that are key when making strategic technology investments. The downside is that safety comes at the price of openness.

## WHAT IT MEANS

### AJAX CHAOS PUSHES DEVELOPERS TOWARD OPEN SOURCE OR ADOBE

The explosion of proprietary Ajax frameworks threatens to overwhelm the benefits of Ajax: openness, standard languages, and performance. The result:

- **Ajax users will act tactically while they wait for a market shakeout.** The sheer number of open source and commercial offerings is unsustainable. This will cause Ajax adopters to delay substantial investment decisions until the market shakes out. Open source tool kits like Dojo, GWT, and OpenLaszlo will benefit because the opportunity cost of using them is low.
- **Commercial Ajax providers will be caught between open source and Adobe.** Choosing a commercial Ajax solution means adopting a proprietary framework and development tools. In this light, commercial Ajax vendors look more like Adobe than like open source Ajax tool kits. But Adobe's heft and market presence will make it difficult for smaller vendors to compete effectively on this point.
- **Adobe will turn up the heat by improving its open source strategy.** Adobe will squeeze commercial Ajax providers even more by improving support for integrating custom controls into Flex and by pushing Active Script into the Mozilla organization.<sup>14</sup>

## ENDNOTES

- <sup>1</sup> Forrester has written about these RIA benefits. See the September 1, 2006, Best Practices “[Rich Internet Applications: Why and How](#).”
- <sup>2</sup> The results of InsureMe’s RIA development efforts can be viewed at <http://www.insureme.com/>.
- <sup>3</sup> RIAs can be full-blown software applications, multimedia experiences, entire Web sites, or embedded components on a Web page. While Ajax and Adobe can both be used in this manner, Flash 9’s download size makes it less practical compared with Ajax frameworks for the embedded component use case.
- <sup>4</sup> Examples of framework size run from 20 KB (JackBe) to 50 KB (Backbase) to 140 KB (Dojo Ajax Toolkit) to 1120 KB (GWT). This is in comparison with 1967 KB for Adobe Flash.
- <sup>5</sup> Examples include GWT Designer, Backbase Ajax: Developer Tools (Backbase Ajax: Widgets), and TIBCO General Interface (General Interface Framework).
- <sup>6</sup> A recent Forrester survey showed that 95% of enterprise can be reached by supporting just IE 6.0 and FF 1.5. See the October 24, 2006, Trends “[Enterprise Browser And Desktop Trends , Q3 2006](#)”
- <sup>7</sup> Flex consists of an XML-based markup language called MXML and a strongly typed scripting language called ActionScript that is similar to Java and C#.
- <sup>8</sup> A recent Forrester survey showed that 96% of enterprise users supported Flash Player 7 and above. See the October 24, 2006, Trends “[Enterprise Browser And Desktop Trends, Q3 2006](#).”
- <sup>9</sup> Examples include Eclipse-based plug-ins that allow developers to visually create widget styles and graphical property editors that allow developers to quickly set commonly used widget properties and immediately preview the results. In addition, developers can exchange graphical assets with design tools like Adobe Photoshop and import themes and skins to apply to Flex controls.
- <sup>10</sup> Salaries range widely based on geographic location and years of experience but tend to be US\$5,000-US\$15,000 higher than Ajax/JavaScript positions in similar locales with equivalent years of experience.
- <sup>11</sup> A recent Forrester survey showed that while 87% of Flash users are usually at the current or immediately previous version, 43% of them will abandon a site when requested to upgrade. See the October 24, 2006, Trends “[Enterprise Browser And Desktop Trends, Q3 2006](#)” and the September 13, 2006, Trends “[Web Users Want Rich Internet Applications](#).”
- <sup>12</sup> Introducing new controls and ways of manipulating displays that are enabled by Flex can actually distract users. For an example of how this happened at Intuit, see the Sept. 1, 2006, Best Practices “[Rich Internet Applications: Why And How](#).”
- <sup>13</sup> A/B tests allow users to compare a new RIA version of a UI to the existing Web version and can help identify unforeseen issues with human factors before an application is deployed.
- <sup>14</sup> Adobe will contribute source code for the ActionScript Virtual Machine to a Mozilla Foundation Project called Tamarin. Source: “Adobe and Mozilla Foundation to Open Source Flash Player Scripting Engine,” Adobe press release, November 7, 2006 (<http://www.adobe.com/aboutadobe/pressroom/pressreleases/200611/110706Mozilla.html>).

---

*Forrester Research (Nasdaq: FRR) is an independent technology and market research company that provides pragmatic and forward-thinking advice about technology’s impact on business and consumers. For 22 years, Forrester has been a thought leader and trusted advisor, helping global clients lead in their markets through its research, consulting, events, and peer-to-peer executive programs. For more information, visit [www.forrester.com](http://www.forrester.com).*

© 2006, Forrester Research, Inc. All rights reserved. Forrester, Forrester Wave, WholeView 2, Technographics, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. Forrester clients may make one attributed copy or slide of each figure contained herein. Additional reproduction is strictly prohibited. For additional reproduction rights and usage information, go to [www.forrester.com](http://www.forrester.com). Information is based on best available resources. Opinions reflect judgment at the time and are subject to change. To purchase reprints of this document, please email [resourcecenter@forrester.com](mailto:resourcecenter@forrester.com). 40989