

# most significant bits

News and Information For High-Tech Professionals

Published by Pocket Protector Press™

A division of Stout Systems



Summer 2008

## In This Issue

### Silverlight Goes For The Gold

What do you know about Silverlight and its impending impact on video streaming? Do you know how you may be using it soon?

### Usability and User Centered Design

Too often overlooked in application development, usability is looked on more and more as the make-break in software success.

### Recent News

Stout Systems welcomes new employees Alan Clark, Larry Siden and David Crane.

Please visit our newly designed Website:  
[www.StoutSystems.com](http://www.StoutSystems.com).

### Job Openings

Check out current permanent and contract openings on [StoutSystems.com](http://StoutSystems.com) or on our mobile site [StoutSystems.MOBI](http://StoutSystems.MOBI).

### Subscribe To Our Newsletter

If you would like to receive this newsletter directly at your desk or in-box, send a note to [info@stoutsystems.com](mailto:info@stoutsystems.com) providing your subscription info (e-mail or ground mail address).

**Call us:**  
(734) 663-0877

## Seen The Light?

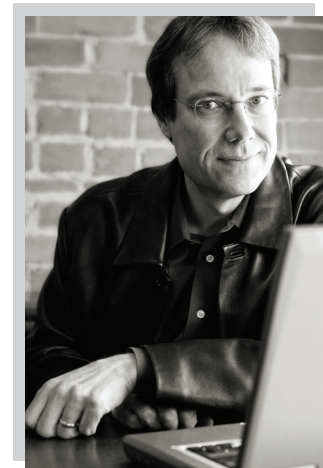
# Silverlight Goes For The Gold

by John W. Stout

**A**lthough Silverlight hit the scene in April 2007, for many people their first exposure to Silverlight has been from watching streaming video of the 2008 Olympics on [nbcolympics.com](http://nbcolympics.com).

To estimate the success that Silverlight is having with carving out a piece of the market, consider that for the entire 2004 Olympics in Athens there were 2.2 million video streams. In the first four days of the 2008 Olympics, there were 13.5 million with Silverlight. NBC and Microsoft are enabling viewers to watch more than 3500 hours of Olympics coverage, with nearly 3000 hours of *live* coverage being made available. On August 11th alone, 250 terabytes of data was delivered via Silverlight. The price of admission for an Olympic video stream? A free download of Silverlight coming in at a mere 4 MB. And whereas Adobe claims to see up to 10 million downloads of Flash each day, Microsoft has been seeing 8 million downloads a day since the start of the Olympics. As product debuts go, many feel this one has been a gold-medal standard for Microsoft.

So, what exactly is Silverlight? In Microsoft's own words, Silverlight is a cross-browser, cross-platform, and cross-device plug-in for delivering the next generation of media experiences and rich interactive applications for the Web. Silverlight actually has as its roots a technology introduced by Microsoft known as Windows Presentation Foundation (WPF). WPF is to desktop applications what Silverlight is to Web-based apps—namely, it enables rich control, design and development of the visual aspects of Windows programs. WPF used XAML (Extensible Application Markup Language) and C# for code-behind and ran under the .NET Framework 3.0. Microsoft's direction then



focused on providing a rich Internet experience for Web pages in alignment with the emerging concept of Web 2.0. From this decision was born Silverlight 1.0, which was basically XAML script and some code-behind to control it. The difference between Silverlight 1.0 and 2.0 (still in beta) was simply the code-behind—in 1.0 it was JavaScript. In 2.0 it is C#. With the addition of a larger number of controls, animations and the ability to code in C#, Silverlight truly does provide the opportunity for creating Rich Internet Applications (RIAs). For a sampling of Silverlight apps, browse through some of the showcased Websites at [silverlight.net/showcase](http://silverlight.net/showcase).

We asked some of our colleagues in the industry if they had begun using Silverlight yet. One of them commented that he had used it in relation to a SharePoint 2007 portal project. Part of this developer's application was a dashboard that contains custom Web parts to display gauges and charts. With Silverlight he was able to make these charts more interactive without having to resort to what he referred to as "cumbersome AJAX". He also felt that he was able to have a

*continued on page 4*

# Usability and User Centered Design

# H

ave you ever:

By Mike Monan

- Watched a product or Website go to market and not succeed simply because the users couldn't figure out how to use it effectively?
- Seen support lines light up when your product is difficult to use?
- Heard about a product that failed because it was designed so that migration from a competitor was too difficult?
- Seen a business go under because the cost of man hours for maintaining its products was too high or the number of sales generated using a Website was too low?

All of these examples can be related to the "usability" of those systems.

"[Usability refers to] the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." (ISO 9241-11)

This article will discuss a general design method and some specific activities related to Web and general software application usability.

Usability relates to the most important part of nearly any commercial or Web application: the user interface. The user interface is the only thing ever used by your target audience to judge the quality and effectiveness of your product relative to that of a competitor (or previous versions). The quality of the user interface can't be overstated. It can't be an afterthought, and it must drive functional requirements.

Can usability be enough to get you over the success hump? The simple answer is **yes!** A compelling, enjoyable and usable interface is a market differentiator. A well-tested interface will save dollars and efforts on post-launch user support, software maintenance costs and documentation costs. It will also increase user satisfaction (think Google), user loyalty (think Macintosh), and user adoption (think iPod). When the usability is well designed, your product will experience a longer market life—saving development effort and dollars. "The rule of thumb in many usability-aware organizations is that the cost-benefit ratio for usability is \$1:\$10-\$100. Once a system is in development, correcting a problem costs 10 times as much as fixing the same problem in design. If the system has been released, it costs 100 times as much relative to fixing in design." [1]

Usability is based around user-centered design processes. **User-centered design (UCD)** is an approach to design that grounds the process in information about the people who will **use** the product. UCD processes focus on users through the planning, design and development of a product. UCD advocates grounding functional requirements and design to observed user behavior patterns and tendencies. The difference between usability and user-centered design is that usability is the result, the goal. User-centered design is the methodology used to reach that result or goal.

The UCD design and development methodology can be contrasted to the traditional "waterfall" development methodology in which a project progresses through the following phases, nearly always flowing down:

1. Scope
2. Design
3. Build
4. Test

Each of these steps flows downward to the next. It is traditionally difficult and expensive to move backward between these discrete phases, and

the emphasis is on the step-by-step discipline of the development model.

The main characteristic of waterfall methodology is that it assumes that you know all the answers in the first step, *before* you begin the design work. How many projects have you worked on (of any size or consequence and where real-life users were using it) where you knew all the answers ahead of time? It's a recipe for failure. A downside to waterfall methodology is all the testing is done at the end of the project. Much easier to unit test small parts throughout as opposed to trying to identify and test every possible scenario and code path at the end of the process.

A UCD design model is analogous to a spiral shaped development model, in which you follow an iterative process that covers the following stages:

1. Requirements gathering
2. Design
3. Implementation
4. Testing
5. Analysis of testing results

Prototypes and requirements are developed and refined throughout each iteration. Results from the previous round of these five steps are used to inform the current iteration of these five steps. Actual user feedback is gathered and analyzed early and often in this process and fed back into the development of the product. User-centered design saves development costs by not expecting everyone to know all the answers ahead of time.

Sounds great, right? So how do we gather requirements? What types of design do we use? How do we get good data out of user testing?

**Step 1. REQUIREMENTS GATHERING.** Gathering requirements is important so that you can generate an accurate model for what needs to be designed. A few important areas to explore include:

1. Audience identification and analysis
2. Needs analysis
3. Competitive analysis
4. Task analysis
5. Usability success metrics requirements definition

Who is your audience? What is the age, computer skill level, preferred platform and browser and education level of your basic users? What design considerations must be made for these audience members? Your client's stakeholders and marketing folks can identify these audience members. You may need to expand their analysis to include software or Web specific information like platform or browser.

Once you identify audience members, you can create some user personas to concretely describe representative members. User personas are descriptions of a specific, plausible user of the system, against which to check your designs.

What are they going to need to accomplish with your product for it to be considered a success? What do you need them to do? What problem are you solving? Why are they going to come and use your software or your Website over what they currently use? Traditional marketing focus groups and one-on-one interviews are great for getting need analysis data.

Focus groups and interviews establish the needs, lifestyles, habits and work preferences of the target audience. In order to gain a deeper understanding of work and living practices relevant to the use of your services, you should use ethnographic studies (observation of people in their workplace) that include multiple techniques such as user observations, user interviews and artifact analysis. Focus groups and user interviews are two tools used in this practice.

What are your competitors doing? What are they doing both well and badly? What can you improve? What features are expected by the

market you're serving? Your client's (or your own) marketing department and Google are great resources to help you with this!

What are the required tasks needed to accomplish an audience member's needs? What are the ten most important things a user will need to do on the site? Talk to your customers/stakeholders. Do their answers correlate with the identified user needs? Challenge them on this if it doesn't. Good task analysis is critical to a good design.

Critical task identification and analysis is a method that evaluates how people actually seek to meet their needs on your Website. Through observation and/or user and stakeholder interviews, you can determine the critical set of goals belonging to the target user. A set of tasks supporting these goals is determined and prioritized based on criteria such as the importance of the goal to the stakeholder or customer and the frequency a user will attempt task execution.

Traditionally, the highest priority tasks are broken up into individual steps. You then suggest ways to make the task more efficient, or suggest new steps that more effectively support the goals. It is important to recognize that the analysis is done from the perspective of the end-user, rather than from the point of view of managers or executives who do not typically use the system.

Task identification and analysis is an essential step to undertake before user testing (step four). User testing will identify the success rate for a given set of users completing the critical tasks.

Finally, what will be the task success rate metric on the site? Is 80% successfully completing the identified tasks good enough to go to market? Does each task have a time goal to be accomplished by users for the design to be regarded as a success? These metrics are important because they will let you know when you've iterated your design process enough to say you are complete. It gives an objective goal against which design progress can be tested.

**Step 2. DESIGN.** In order to get something accomplished in the design process quickly, coming up with low-fidelity prototypes is critical. Low-fidelity (e.g. pencil, pen, marker drawings on paper) prototypes have a couple of critical features. First, they convey to your evaluators that they can be changed, thrown out, modified, drawn on, marked up, etc. They don't look complete, and so they invite critique. Secondly, they can be put together by someone with only a minimum of design skill, and they can be created very quickly. Low fidelity prototypes can be translated into wireframe models, can act as design direction for a graphic designer, can be turned into effective HTML prototypes, etc. The goal of the design phase is to map the tasks identified in the requirements phase to a real-world design in a way that supports the identified needs. The designs should support the audience members' characteristics and limits.

Low-fidelity prototypes don't just need to refer to the graphical layout of a page. Information architecture can be designed in a low-fidelity way (index cards, card-sorting exercises), interaction design (storyboarding to show state transitions of an interface), and even user testing of a system can be started using this low-fidelity modeling. Card sorting helps to define common mental maps for the system in question. It is most commonly used in Website architecture creation (for complex sites). Users actually organize the content of a Website into groups using index cards. The results tend to clearly indicate where users expect to find information on your site, how they would group this information, and how they would label the main categories of your site. You can then use this information to adjust your own ideas about your site architecture and navigational labeling.

**Step 3. IMPLEMENTATION.** The implementation should support an architecture that separates out (as much as possible) function from design. This will ensure that interface changes can be made with minimal disruption to the underlying logic that drives it.

**Step 4 is TESTING, and Step 5 is ANALYSIS.** Testing and analysis can encompass many different activities, but here we discuss user testing.

User testing is the most effective method for identifying usability problems. Nothing is more convincing than watching person after person encounter difficulties with the same part of a site. Problem areas that repeat themselves between multiple test participants give specific directions on what should be studied and changed by designers and developers. User testing can often uncover very specific areas needing improvement, where focus groups and task analysis often find more general areas needing improvement.

In order to have baseline data, test on a previous version or a competitor's site. This generates a starting point for comparing results. Identify goal metrics in the Requirements gathering that defines an end point. The number of interim user testing rounds to perform (one per iteration of the UCD model) depends on the quality of the design and the project timeline. A final measurement versus metrics round is usually completed to see how the finished product stacks up relative to the starting point and goal metric.

A trained usability specialist isn't required to conduct the testing—although I do recommend this in order to get the most value out of the test. Someone who is very observant and a good listener is required. Representative users of the target audience are asked to perform the identified set of critical tasks (in a created context) on the site. A user testing report is created

that identifies problems (prioritized by importance) and includes recommendations to correct these problems in the next design iteration. In addition, measurements are made against the success metrics that have been defined for the usability of the system. This enables the ongoing evaluation of progress in the design.

That's it! Lather, rinse, repeat! You've now got a user-centered design-based product that you can be sure meets the objective goals you've set for critical task completion and speed. It will also support your users' needs. Congratulations, your product is probably just fine!

**Mike Monan** is co-founder and usability lead at Ann Arbor-based Switchback ([switchbackcms.com](http://switchbackcms.com)), a Web software company focusing on content management systems. He has over ten years' experience building Internet-based software that incorporates usability best practices. Email him at [michael.monan@switchbackcms.com](mailto:michael.monan@switchbackcms.com).

**Reference:** [1]: \*Gilb, T. (1988). Principles of software engineering management. In Usability is good business. Retrieved October 15, 2001, from <http://www.compuware.com>.

*Some additional web resources can give you information on any of the terms or practices that were banded about in this article:*

[www.usabilityfirst.com](http://www.usabilityfirst.com)

[www.upassoc.org](http://www.upassoc.org)

[www.upassoc.org/usability\\_resources/usability\\_in\\_the\\_real\\_world/index.html](http://www.upassoc.org/usability_resources/usability_in_the_real_world/index.html)

[www.upassoc.org/usability\\_resources/usability\\_in\\_the\\_real\\_world/roi\\_of\\_usability.html](http://www.upassoc.org/usability_resources/usability_in_the_real_world/roi_of_usability.html)

[www.disambiguity.com/waterfall-bad-washing-machine-good-ia-summit-07-slides](http://www.disambiguity.com/waterfall-bad-washing-machine-good-ia-summit-07-slides)

**“The user interface is the only thing ever used by your target audience to judge the quality and effectiveness of your product relative to that of a competitor (or previous versions). The quality of the user interface can't be overstated. It can't be an afterthought and it must drive functional requirements.”**

*continued from page 1*

more usable UI than he could with HTML. For some samples of interactive charts similar to what this developer is referring to, you can go to [infragistics.com/launch-faceOut](http://infragistics.com/launch-faceOut).

What does it take to start developing Web apps with Silverlight? First, you'll need to download and install Visual Studio 2008. Then, you'll need to install Silverlight Tools Beta 2 for Visual Studio 2008. This add-on will allow you to use .NET to create Silverlight 2 Websites ([silverlight.net/GetStarted](http://silverlight.net/GetStarted)). Technically, you could begin creating Websites with just this, but to take full advantage of the RIA features available, you'll also want to download preview versions of Expression Blend 2.5 and Deep Zoom Composer. Expression Blend will allow even more control over the visual and interaction design for the development of rich interactive Web applications. Deep Zoom Composer will allow you to do some amazing image zooms. Check out [memorabilia.hardrock.com](http://memorabilia.hardrock.com) for an excellent sample. Just as an example, when at this site, select the Beatles memorabilia. In the lower row, you will see a postcard next to some letters. Zoom in on the stamp in the upper left of the postcard. You'll see that the stamp is actually 12 separate images, and although you can zoom in on any one of these with surprising results, you'll want to zoom in on the second image from the left on the third row. You'll see the Hard Rock café in the distance—continue zooming in toward the window under the Hard Rock marquee on the left. How can you learn more about Silverlight? A great place to begin is [silverlight.net/GetStarted](http://silverlight.net/GetStarted). Here you'll be able to download all of your Silverlight components and dive right into some great tutorials. If you take the plunge into Silverlight, feel free to e-mail us about your experiences.

**John W. Stout** is the founder and president of Stout Systems Development. He has nearly thirty years' experience in the software industry. He is also sought after as a technology speaker, presenting sessions at developer conferences and user groups. E-mail [john@stoutsystems.com](mailto:john@stoutsystems.com).

**Stout Systems Addresses IT Worker Shortage**  
In September, John Stout will be a featured speaker at the Emerging Technologies Breakfast event at Automation Alley headquarters in Troy, MI. The event covers the crucial topic of the shortage of IT workers in Michigan and what can be done to attract and retain top technical talent in our state. The event includes dynamic speakers from Microsoft Corporation and TechTown in Detroit. The event is scheduled for 7:30 AM on Wednesday September 17. Contact us for more information.

4



ADDRESS SERVICE REQUESTED

Presorted Standard  
US Postage Paid  
Permit #187  
Ann Arbor MI

In This Issue:

- Silverlight Goes For The Gold!
- Usability and User Centered Design