

Designing web-based applications that work

© 2002, Jody Levinson, [TroutDream Graphics, Inc.](#)

Software companies are jumping on the bandwagon to deliver their software online. A serious lack of understanding about designing and implementing web-based applications is hampering usability.

Simply defined, a web-based application is software delivered over the Web that performs more complicated tasks than just collecting or retrieving information. A web application can be a traditional client-application (contact management, accounting) or a service (online banking, investing, EDI) or an automated approach to activities traditionally handled "live" (customer service, product customization).

Some of the advantages of distributing software via the Web:

- **Attractive to users** – It's available on demand, instantly, without the hassle of installing or maintaining software on multiple machines. It's accessible anywhere, from any computer, and always the latest version.
- **Attractive to developers** – Since the software resides only on the company's servers, only one configuration needs to be tested and no installation scripts need to be written. Upgrades are a snap.
- **Attractive to the bottom line** – A wider variety of pricing options (subscriptions, pay-per-use, advertising) will allow you to reach more users who have different needs. You also save the cost of reproduction, packaging, shipping and breakage.

What are the issues in designing web-based applications?

Whether the application is created by a company's in-house staff, an independent programmer or outsourced to an Application Service Provider (ASP), it is important to make sure that the user interface is expertly designed. Good web-based application design comes from understanding what makes it different from both corporate Web sites and traditional client applications.

Web applications are different from traditional Web sites in many ways:

- **More interactive** - requiring constant user action and often more complex interactions than traditional Web sites.
- **Used more intensively and more often** – Traditional Web sites are often designed like ads or brochures – the intent is grab the user's attention. This is not appropriate for an interface users will be working with frequently, perhaps even daily.

- **Users are willing to invest more time in learning to use the application** – Traditional Web guidelines recommend against any functionality on a website that requires training the user. While the application should be as clear and easy to understand as possible, users want the functionality of the software and are more willing invest a little effort in order to get it.

Web applications are different from traditional client applications in many ways:

- **User expectations** – Users have come to expect to interact with Web sites in certain ways, such as one-click navigation and following underlined links. These conventions should be observed in applications as well.
- **Delays in downloading and connection errors** – Users often multitask while waiting for information to download and may need to be reminded to return to the task. You will also need to handle errors that arise from connection failures.
- **Browsers don't handle multiple windows well** – While multiple windows and floating palettes may work well in traditional client-applications, they tend to be more confusing in browsers, where windows are likely to become lost or hidden.

Some guidelines for designing web-based applications:

These are the most important factors which will guide you in creating a good web application user interface:

Know the user.

- **Who is using the application?** A low-paid clerk with minimal training? A highly-skilled professional with years of experience? An at-home neophyte trying something new?
- **How have they performed this task in the past?** Are they used to a particular way of doing this?
- **Are there industry standards for this task?** How are your competitors doing it? You should not blindly follow the methods used by industry leaders in their applications, but it's a good thing to know what else your users may have seen.
- **Are there capabilities of the web that make industry standards obsolete?** How do you overcome resistance to change?

Know the workflow.

Find a balance between too many tasks on a screen and too many clicks to get through the tasks. Grouping many tasks on a single page may seem "efficient" and avoids delays and errors that can arise between page loads, but users will be confused by pages that start to look like space shuttle dashboards. However, system response time is always an issue and it's a delicate task to walk the line between wading through pages that are too cluttered and waiting for too many page refreshes.

Organize screens logically - Group related tasks together on the same screen. Standardize the locations of various work areas. High level navigation (Home, Log Out, etc.), task level buttons (Add, Print, Clear, Submit, etc.) and workspace areas (data entry fields, checklists, etc.) should appear in consistent places on every screen.

Think through the workflow - The user should be able to complete a task by proceeding in a linear fashion without backing up or making side trips.

Rules of thumb for web-based applications

While every application has different requirements, these are a few of the general guidelines that will help create a consistent, easy-to-use interface.

- **Label buttons in plain English** - Avoid labels that are graphics only. Use terms users are most familiar with. For example, labeling a button "*Create EDI*" may be accurate from the system's point of view, but from the user's point of view, "*Send Invoice*" would be clearer.
- **Minimize use of multiple windows** - It's less confusing to the users and more consistent with their expectations of browser functionality. An exception would be instances where the user would expect to see a new window, such as a dialog box.
- **Hide browser tools** - Simplify the workspace. Web browsing toolbars can become confused with application tools.
- **Use consistent layouts** - Care should be taken to see that the application appears consistent on various browsers and platforms. Establish clear distinctions between tool areas, work areas and header, label, and text styles.
- **Use Web conventions** - This is no time create a new paradigm. Users are comfortable with many Web conventions and you should take advantage of that in implementing links, hot-spots, radio buttons, checkboxes, pull-down menus and navigation.
- **Use rollovers** to provide instant access to explanatory information while reducing screen clutter.
- **Avoid double clicks** - Users are accustomed to single-clicking on the Web. An exception would be instances where users are very accustomed to double-clicking in a traditional application, such as to move items from one list to another.

- **Limit Submits to one per screen** - It may seem efficient to avoid making the user load another screen for the next task, but it's confusing.
- **Keep the look and feel simple** - The application should be consistent and have a look and feel appropriate to its function. While it should blend with the company's corporate site and other branding, the glitz and graphics associated with marketing are sure to irritate users who may be spending many hours using your application.

Go forth and applicate!

Web-based applications are the wave of the future. They allow companies to reach a wide audience with their services and keep their products up-to-the minute while providing users with convenience in a familiar environment. Moving an application to the Web is more complex than simply retooling the off-the-shelf version to run in a browser window, but with careful consideration and planning, the results can be a joy for both provider and user.

[TroutDream Graphics, Inc.](#) provides web-based application user interface design and consulting. To get your application in shape, please [drop us a line](#).