

A Successful Approach to Implementing a Corporate Web Design Standard

White paper

Dick Rubinstein, Ph.D., CUA
Managing Director
Human Factors International

May 22, 2003

Human Factors International
410 West Lowe
PO Box 2020
Fairfield, IA 52556
800-242-4480
hfi@humanfactors.com
www.humanfactors.com

Table of contents

About the author3

Chaos in the fast lane4

Why bother?4

Usability hell5

Affordances and behaviors6

Branding7

Cost justification7

Tried and failed8

What to do? The hard and soft problems9

The technical dimension: the state of the art10

But not too restrictive11

Page structures13

The organizational dimension: ten organizational secrets14

Conclusions14

Appendix: ten organizational secrets
for implementing a successful Web standard16



Dick Rubinstein, Ph.D., CUA
Managing Director
Human Factors International

About the author

Dick Rubinstein has worked in the professional human factors field, designing interfaces since 1975. He worked as a Senior R&D Analyst with Bolt Beranek and Newman, Inc., then as a Consultant Engineer with Digital Equipment Corporation. He was an Independent Consultant with Cognitive Construction Company, a member of Technical Staff with Sun Microsystems, a User Interface Architect/ Manager of the UI Team with Marcam Corporation, and the lead User Interface Designer for Time0, a Perot Systems eCommerce company. Currently he is a Managing Director with Human Factors International, Inc.

His special skills include interaction design, user interface design, usability enhancement, product conceptualization and vision, team leadership, cross-function communication, and collaboration.

His technical qualifications include design, task analysis, prototyping, testing, usability enhancement, and interpersonal communication. He has broad knowledge and experience in computer systems, including hardware and software architecture, the Internet, multimedia, human factors, digital typography, graphic design, computer assisted instruction, strategic planning, and technology assessment.

Dick is a frequent conference committee member, including SIGCHI (Human Factors), SIGGRAPH (Graphics), EP (Electronic Publishing), RIDT (Raster Imaging and Digital Typography), and DOCPROC (Document Processing). He is known as an excellent speaker and has taught courses, seminars, and given presentations on design principals since 1975.

Chaos in the fast lane

A few years ago, I was working as a consultant for a big corporation that had come early to the Web, very much to its credit. This was a company with a decentralized corporate culture—rather than the big, hierarchical kind of organization you might expect of a large bank, insurance company, or computer vendor. The result, of course, was that every department had its own way of doing things right. For the Web, this meant about 15 Webmasters, one for each group, all intelligent, hardworking, and opinionated about what a Web page should be.

With so many people doing things in different ways, and with the resulting large investment in code, third-party software, and other sunk costs, the possibility of creating one standard might have seemed hopeless. How do you get so many different interests to converge?

Why bother?

What's so bad about having these different approaches? After all, isn't each well-tuned to the needs of its sub-organization, users, requirements, and environment? Maybe this isn't a problem worth fixing, after all.

It is a bad situation. First, when there are users who move from one fiefdom's pages to another, they are likely to get confused, make mistakes, and generally waste time that could be spent more productively. The lack of consistency is itself a usability problem.

One of the worst problems created by the independent creation of Web content by each sub-organization is that the Web architecture comes to reflect the organization, not the needs of users. This is frequently a crippling usability problem for users of the public Web site—they have no reason to know how the company is organized, but that structure determines where things are located on the Web site. The resulting mismatch makes the site hard for outsiders to use.

A site composed of the work of a number of groups doesn't look like the work of one company, which can be a problem for both Intra- and Internet sites. Inside a company, the lack of common page structure, branding, and terminology has a negative effect on morale and the overall effectiveness of the organization. HFI has had many customers who make presenting a carefully chosen internal image a primary goal of their Intranet efforts. For example, one banking customer that was acquiring many smaller banks created its Intranet home pages for just that purpose—to bring together all the new members of the organization and create a sense of working for a single company.

Most companies care what the public or its customers thinks of them. They often pay large amounts of money to influence that opinion. For sites used by outside people, a lack of standards can mean that the projected corporate image is a hodge-podge. The chaos not only makes

the site harder to use, it also reflects on the character of the organization. If the site is hard to use, the company may be hard to deal with, too. We call this a transfer effect: it's sort of guilt by association for Web sites.

Cost is another important reason for solving the standards problem. While many of the costs of not having good usability may be hidden or indirect, they can be substantial. Some of the internal conventions developed by working groups may provide good usability; others may not. Any loss of productivity for internal corporate users shows up on the bottom line, especially in times of constrained resources. When customers and business partners use Internet facilities of poor quality, we incur opportunity costs from lost business, support costs for customer service, and productivity costs in coordinating with business partners. Any of these costs alone may be enough to justify a standards effort.

This paper describes the practical difficulties in implementing a corporate user interface standard, and HFI's successful approach to these problems. We have been helping corporations create, deploy, and stick with user interface standards for over a decade, and have developed a number of effective techniques for creating standards. I will explain HFI's methods, and why they work.

Usability hell

Consider a site that has the appearance that typically results from different groups building sections of an organization's Web site. I've chosen



here a government agency, Health and Human Services, but I'm not picking on them specifically. Thousands of Web sites are like this, and for the same reason: no standardization.

In this case, starting at the home page (top left), there is one navigational structure presented, visible as ovals relating to site sections on the left. Navigating into the site, the first-level pages mostly look like the one at the bottom left. They have a different banner structure for navigating within the site. To the site's credit, it's the same list of sections, but displayed in a different way graphically, across the top of the page instead of the left side.

However, moving down another level brings greater divergence. The image on the top right is a sister agency page, sharing the navigational space. It's not just a link off the site, but a coordinated piece of the site run by another organization. Finally, the bottom right image shows the appearance of many of the content pages in the site: the navigational framework is completely lost.

Affordances and behaviors

Worse than the different navigational structures is the use of different conventions for how things work. One section of the site will have buttons with one appearance and behavior; another section, a different set of choices.

"Affordance" is the match between the appearance and the behavior of things on Web pages, or more broadly, in engineering artifacts in the world. We all base our expectations of how something will work on our experience with similar-looking things. For example, consider the door handle below.

You may never have seen a door handle quite like this one before, but based on your experience with the world, you can predict how to use it!



Have you ever seen a door handle like this? (Picture courtesy of Lindustries, Inc., Westond, MA. The handle is an attachment to a standard doorknob that makes it easier to use for people with disabilities.)

Because it's attached to a door, you know it's probably lets you open the door. Its lever shape suggests pressing up or down on the thin section. The round area suggests an axis of rotation. Or, you can grab the thin section and pull. Thus, this structure *affords* turning, pulling, and other actions.

Similarly, the visual appearance of controls on a Web page suggests their behavior. If users correctly predict the behavior from the appearance, it's good affordance; otherwise, the affordance is poor. Because there are a lot of conventions on the Web, and because it's OK to present controls that are a bit unconventional if the affordance is good, software often has some variation in what controls look like, and how they work. But if different parts of the site or application have different conventions, no good will come of it.

Branding

Marketing and PR people are especially sensitive to branding and brand image in corporate products. This is for a good reason. It's hard enough to get customers to visit Web sites. Once there, it takes very little to turn most users off. On a first visit, if the site doesn't look like your intended destination, the browser's "back" button is right there at hand. If navigating for a step or two brings up pages with a different style, users will think they've left the first site and gone to another. Corporations spend millions of dollars on their image, product positioning, and brand values. Inconsistency in presentation can invalidate this effort for a user in about two clicks.

Cost Justification

The cost calculations that justify addressing usability for internal sites make a strong business case for standards. We had an engagement with a telecommunications company that had calculated that they saved \$1 million per year for every *second* they reduced the average time it takes for a customer service representative to handle a customer care call. Even with highly trained, very experienced customer service people, it's easy to see that extra navigational steps, or difficulty finding the right information, costs real money.

The costs of poor usability in a corporate Intranet site are less obtrusive, but just as real. One corporation I worked with had a terrible HR site on their Intranet. Everyone hated it, because it took so long to do things, especially the everyday things. All employees were required to enter their time using this Web facility, and they were also required to complete many of their HR interactions using the self-service pages. Employees were also expected to maintain their correct mailing address, life insurance beneficiary, and dependent information using the site. Unfortunately, the pages that provided these HR services were very hard to locate on the Intranet, and many of the pages were confusing and hard to use. Whose budget was affected by the lost productivity? Everyone's. HR didn't have the budget (or the will) to fix the problem.

As an estimate of the internal cost, consider that this company had many external projects, and as a result, employee's time reporting was

important, meticulous, and rather complex. If the 10,000 employees spent an average of 5 extra minutes each filling out their time sheets and performing other Intranet tasks (an underestimate), that's over 800 hours *per week* wasted on inefficient overhead tasks. It's the equivalent of 20 extra employees across the company, and over \$2 million per year that could be saved by addressing these usability problems.

The situation is a bit different for e-commerce and B2B. Why should my company care if our system does not fully optimize our customers' productivity? After all, we're already boosting their productivity by providing B2B services, which save customers money in paperwork and purchasing staff. The watchword in B2B has been "disintermediation"—eliminating middlemen and their associated costs. B2B intrinsically saves businesses money. While that may be true, people perceive quality in relative terms. If your system isn't as easy to use as a competitor's, then you may lose the business. Even in situations in which competition isn't a big factor, improved productivity means more added value. What the B2B service is worth is in proportion to its value to its users and their organizations. Customers pay for the value they receive, at least in the long run. It's important to add as much value as we can in our products.

Tried and failed

Thus, there are many reasons to want our applications and Web facilities to be as productive and easy to use as we can make them, subject to our other constraints. Many companies have set out, with high purpose, to create interface standards for this reason, but failed because of organizational reasons. Here are some of the things that can go wrong:

- *Senior staff commitments:* A frequent scenario in some companies involves a failure of participation. Everyone's too busy. Someone rounds up all of the senior people, the architects, programming managers, software designers, and so on, and has one or two meetings about making a standard. Everyone agrees that it must be done, and everyone agrees to help. However, these are just the people who don't have the time. None of these people can drop everything and take responsibility for the standard, so with the best of intentions, the process fizzles out. Even if there is someone whose job description includes making the standard happen, the critical resources are unavailable for the length of time required.
- *Lack of agreement:* One of our customers, a large bank, worked on a standard for two and a half years of committee meetings, drafts, and reports. No standard emerged because there was no consensus on what the standard had to be, how it should be structured, and who would follow it.

- *Expense of change:* A frequent and valid objection to imposing a standard is the cost of changing existing software to meet the requirements of the standard. The most expensive code to change is code that's already written, debugged, and deployed. It's working, and no one wants to throw it away, or "fix" it. Everyone has better uses for the resources that would be required to make these changes.
- *Lack of management support:* Standards are extremely hard to do from the bottom up. If the benefits of having a standard are not a real priority for senior management, it's very hard to achieve the focus, level of effort, and clarity of purpose required to create a successful one.
- *No buy-in:* There are many corporate standards, created at great expense (and often individual sacrifice) that do not get used. Everyone pays lip service, but few follow the dictates of the standard. This can happen in two different ways. Sometimes, everyone is just too busy doing his own thing to bother. In other cases, individual organizations continue to do their work according to their own standards and practices, at variance with the corporate standard. In the extreme case of no buy-in, different organizations go out of their way to do things differently, as a way of promoting their organizational interests, expressing their philosophy (not carried by the official standard), or as one method of asserting political dominance over other groups.
- *We forgot:* This is the saddest scenario. A perfectly good standard exists... on the shelf. It's not used because the people who created it have moved on. Somehow, the organization, as a whole, forgot that it had done the work.

What to do? The hard & soft problems

With all of these impediments, getting a standard written and deployed seems a Herculean task. There is clearly value to be had by having the standard, but how do you get there?

At HFI, we have successfully helped many corporations implement standards. Over time, we have developed a variety of techniques for addressing the problems described above, and increasing the reliability of the standards process and the likelihood that the resulting standard will be used in the organization.

The first thing to say about what makes our process work is that it has two equally important parts: a *technical* component, and an *organizational development* component. As with most activities that involve both people and technology, addressing just one or the other isn't good enough. The reason that we're so frequently successful in our standards work is that we know how to attend to both sides of the problem well.

The technical part is developing a standard that meets the needs of the organization. It must enable users of the standard to design consistent

and high-quality pages and sub-sites. The standard must allow practitioners to be creative in their fields, not tying their hands creatively, but also leading them to good interfaces that match what their colleagues are doing elsewhere in the company. Thus, graphic designers and page designers should have a degree of freedom in the page designs, to make the pages interesting and attractive, while remaining supportive of the brand. The resulting pages should incorporate best practices for usability on the Web.

The organizational component is equally demanding. People not only must agree on the standard, but they must support and use it. Organizations are complex organisms. People have different goals, viewpoints, and priorities. When we seek agreement across many parts of the company, we are always bucking what's called "The Tragedy of the Commons." Seeking common good from individual action is as old as civilization. It's in an individual's interest to let his cows graze on the common land, but if people don't limit their use of the common, it's overused and ceases to be of value to all but a few. We have the same problem in modern organizations.

The technical dimension: the state of the art

The technical requirements for a standard are stringent. There are many different kinds of standards that one can develop. For example, one distinction among standards is descriptive vs. prescriptive. We have found over the years that standards that describe the qualities of software interfaces don't work. It's the same problem that collections of guidelines suffer from: describing the result doesn't tell a programmer how to achieve it. The shortest such standard consists of the words, "Make it easy to use." The advice is correct, but nearly useless. Standards that describe the goals of structures and features beg the question of how to achieve these goals.

For example, we have often found "dead" standards that were full of valid guidelines such as "Don't use saturated colors for text or text backgrounds," or "Don't use animation," or "Divide the site so that the most important tasks fall into the banner silos." These guidelines are not wrong, though they are often over-restrictive. They are requirements, not directions on how to achieve the qualities they aspire to.

I speak as someone who has written a book full of guidelines. (*The Human Factor: Designing Computer Systems for People*, with Harry Hersh, Digital Press, 1984.) I've come to the conclusion that guidelines are useful mnemonics. They help people organize and remember what they already know. They can inspire people to figure out how to do something well. However, guidelines are not very helpful in communicating what to do. For a guideline to be helpful, you have to already

know what it means. It serves to remind you to do what it says, but that's all.

Even if guidelines were good means of communicating the principles they express, they would still not be particularly useful as the core of a standard. To be useful, a standard must be prescriptive. It must say what to do and what not to do, in definite terms. Let me give an analogy: A good standard is like an oil painting...

When an artist sets out to create a picture, he can do it within an established medium, such as oils, watercolors, or black and white photography. The choice of medium makes the job of creating a picture infinitely easier (though not necessarily easy, of course). By choosing oil painting, the artist chooses to stand on the shoulders of giants, those who have worked out the framework for oil. This choice means that the painting is done on a substrate of canvas or maybe wood, with oil-born pigments of certain types. There are established methods for creating the perception of light, shadow, and color. There is still enormous room for innovation, in the medium itself (use a new pigment), in the technique (a novel kind of brush work), in the subject (something in motion, say), and in the style (pointillist, naturalist, etc.). By choosing oil, the artist avoids having to invent the medium. This choice also excludes other means of expressing oneself, such as sculpture or writing a novel. Making "architectural" choices like this makes the job easier, rather than harder.

In this sense, an art medium represents a useful, constructive set of constraints, but not a straightjacket. There is plenty of room for the essential creative work, but effort is focused on what's to be expressed, not how to express it. A good standard is like this: it sets a useful set of constraints that make the work easier, prevents the need to reinvent the wheel, and puts the focus on achieving the goal (a working Web site or application), rather than on constructing the tools (such as a new style of interface).

But not too restrictive

The proper role of the standard is to create a useful set of constraints. A bad standard feels like a straightjacket. It's too restrictive. An overly-specific standard may show the exact page layout and content for all of the major pages, but may not leave the programmer the degree of freedom he needs to adapt his problem to the requirements of the standard. The worst such standard is one that canonizes a structure that is bad for usability.

Consider the structure of the page below, from a Microsoft site. If this structure were chosen as the standard for a company, what would the consequences be? (We frequently find, when we begin a consulting engagement, that the structure that is causing trouble looks much like

this.) There are some things right and some things wrong about this design. The design doesn't make it clear visually what the navigation hierarchy is, and there is no feedback about where in the site the user is sitting at any given time. These are both bad things to codify into a standard.



But there's a bigger problem. This is a structure that is designed for an informational site. The left-hand navigation, when it is useful, allows the user to navigate two or more levels of an informational tree structure. If the task that the user is doing is to find a piece of information, the left-hand table of contents is a fine approach. (We use it on most of our Web site pages (www.humanfactors.com) and in most of our Web standards.) But it is completely inappropriate for some other kinds of tasks.

For example, consider a customer service application. If the customer service representative has to view and perhaps edit information in many places in the structure in the course of servicing a single phone call or issue, the structure hinders success. It forces the user to do a great deal of navigation to complete a single task, and to remember what has been done, what has yet to be done, and where everything is. For that kind of application, a standard based on this structure is a usability disaster: it forces excessive navigation, with resulting loss of productivity, and it burdens memory, with resultant increases in training time and in the likelihood of mistakes.

So, one moral is that one structure *doesn't* fit all. In particular, it doesn't fit all task structures. A good standard may provide an overall navigational structure, but when the user gets to the task context, there is a navigational structure that supports the task, instead of fighting it.

A good standard includes a variety of local navigational structures that match the tasks that are most often performed within the system.

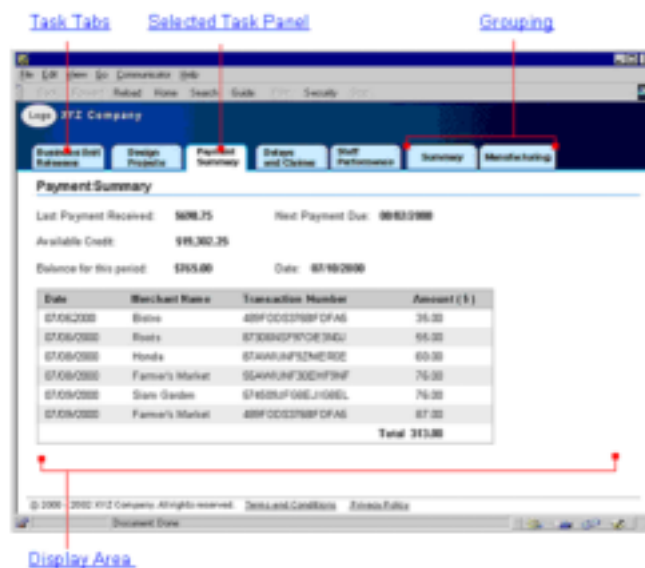
We have great respect for the skills and creativity of the programmers and site builders who must implement applications and Web sites within the constraints of a standard. We believe that a good standard expresses this kind of respect clearly, by leaving programmers or site builders the degrees of freedom they need to meet the functional and usability requirements, to be creative in the approach to this goal, but also to avoid reinventing the wheel or repeating work that others have done before.

Page structures

HFI interface standards incorporate a number of page structures, customized to the specific needs of the company. We have a set of starting-point pages that cover the range of task structures that we usually find are needed in a given standard, and we create the few that we find are missing.

For example, consider the proto-page design in the figure below, chosen from a set included in our Usability Central product. In addition to getting the internal usability issues right (embedding, location feedback), this design also encapsulates a particular workflow. It's appropriate for tasks that have the same workflow. In this case, there are two levels of navigation. The outer level corresponds to the choice of a particular macro

Basic Task Panel



A standard template from Usability Central

task, and the inner level corresponds to the details of the chosen task. For example, the inner tab notebook could represent an account, with all of its sub-components that may need to be viewed or edited.

One of the important features of the standards we create is a decision tree that leads a user of the standard to the right page for a particular purpose. The user must analyze the task to understand its internal workflow. The decision tree asks questions about the task, and leads to the right proto-page design. Of course, the programmer or page designer then must adjust the design to meet the specific requirements of the problem. The good news is that the essentials of good usability are already built in at that point.

The organizational dimension: ten organizational secrets

Let's propose that we know how to construct a standard that is technically sound in terms of usability, appropriate to the organization, easy to use (specifically, much easier than not using it), and that meets the organization's technical needs. Unfortunately, this is not enough.

Equally challenging is creating an organizational process that will support the technical processes of creating the standard, publishing the standard, supporting its use in the company, and maintaining the standard over time.

HFI has developed a number of practices and processes that facilitate the organizational development needed to create a successful standard. These are the real secrets of succeeding with a standard. (For a complete list of these secrets, see the appendix at the end of this paper).

Conclusions

It is essential to the success of a standards effort to attend to both the technical content of your standard and the organizational process needed to complete it and put it into everyday use. With attention to both aspects, as I've outlined above, you can give life to a useful and enduring user interface standard.

With a standard in place, and developers trained to use it, the level of usability will rise. But this isn't the end of the story, only the beginning. An interface standard is a good start toward making usability routine—an everyday part of the product and tool development cycle.

In addition, it is necessary to see that user-centered design becomes a part of project planning. The standard, alone, doesn't ensure good interfaces. A design process is also needed.

HFI recommends a development process, which we call The Schaffer Method™. This process provides a way to integrate user-centered design with the whole software life cycle, including product creation, under-

standing users and their tasks, design, development, and revisions. We also provide an Intranet-based resource called Usability Central that details The Schaffer Method™ and provides standard templates and other tools that are invaluable in creating a standard and infusing knowledge of usability in a corporation. (For further information on The Schaffer Method™ and Usability Central, please visit www.humanfactors.com, or contact HFI at 800-242-4480.)

There are certainly other important considerations that affect the creation of a new tool or product, such as technical requirements, budget, competition, or schedule. But make sure that usability gets its fair share of attention in the tradeoffs that are essential to engineering a successful Web standard.

1. Identify stakeholders

This may be hard for an organization to do itself, and is a good reason to get the help of a consultant, who can look at the whole organization objectively. A standard affects the people who use it, so including the MIS or IT department is a no-brainer. But the standard also affects every organization that has users of software created with the standard. If customers will be users of such software, marketing and sales organizations will also be stakeholders. Leaving a stakeholder out of the standards process invites sabotage and non-compliance. To be effective, the process must be inclusive. Few organizations are autocratic enough at the top to ensure that a standard will be followed everywhere, even by people who don't participate, or don't approve.

2. Get support at the top

A standard will only span that portion of an organization that reports to its highest sponsors. This can be a coalition of those who are responsible for various parts of the company, but it's best to get the support of the person these people report to. We recommend *not* attempting the standard until this criterion is met. It's worth quite a fuss at the beginning of a project to get high-level support, because without it, the project may be doomed to a "tower-of-Babel" failure.

3. Ask the right questions

Data gathering is as important to a user interface standard as it is to any other design process. Without knowing enough about the users and their needs, you can't design the right thing. A standard is particularly hard in this respect, since it's a meta-design. It has to solve future problems for a variety of people, tasks, and situations. HFI combines data gathering with identifying the stakeholders. We complete a contextual inquiry process, observing a wide range of people doing jobs that will be affected by the standard. The goal is to understand the user's problems and perspective, rather than to get their ideas about the technology, interface features, or the like. We welcome input of this sort, but the users are not the designers. The standards committee will own the design for the standard, and it's their responsibility to specify the standard. We don't pass the buck to the users.

4. Limit the commitment of the participants

It's hard to get the high-powered and responsible people who must own a standard to commit the time to work through its creation and maintenance. Therefore, we "contract" with the players for a modest, finite commitment to create the standard, and a limited continuing commitment to maintain it. For example, we might ask 6 days of each committee member, over a 4-month period, which includes reviewing documents and attending standards meetings. After the standard is released, the commitment is usually one day a quarter, at a maximum.

5. Do the legwork for the committee

Don't ask committee members to write the standard! No one has the time, even if you can get someone to agree to do it. Yet, someone must be responsible for the actual writing who can give it his or her full attention. In an HFI standards engagement, we do all of the writing. This means that committee members have something to read and react to. Their job is to come to the standards committee meeting prepared to comment on the draft they've just reviewed.

6. Accumulate progress

Put a definite schedule in place that will result in a standard. We overlap revised and new materials, so there is cumulative progress. Our standards process has a fixed number of committee meetings—usually 4. At the first meeting, we present a draft of the first section of the standard. At the second meeting, we present the revised first section, and the draft of the second section, and so on. In this way, sections receive review, improvement, and approval incrementally. We ask the committee to agree to the process up front. This avoids points of order, late reconsideration of the process, and a variety of other process failures.

7. Train the committee

Not everyone on the standards committee will understand the underlying usability principles that go into the standard. It's important to provide training to as many as will avail themselves of it. We also spend some time in the committee meetings teaching: explaining the reasons for doing things in one way or another.

8. The committee owns the standard

As consultants, we eventually leave. It's important for the committee to own the standard. This means that there must be a process in place for changes and improvements to be made by the committee, as the needs arise. For HFI standards, the primary kind of change is the addition of a page type for a new task flow or application. There may also be broader changes, such as rework of branding or company logos, which require revision of the standard. The committee meets only when there are such issues pending, to resolve them. For a new standard, we find that meetings are required every quarter for a while, then less frequently.

9. Grandfather it!

No one likes change, and most organizations will squelch any standard that requires going back and revising everything to match it. It's important that everyone understand which software has to be revised to use the standard, and that there be as little of this as possible. It's much easier for people to accept and adapt to a prospective change than one that requires revisiting old ground. The cost of revising almost everything would usually be prohibitive, as well.

10. Put a usability process in place

HFI runs a half-day course in how to use the standard, with the goal of having someone in the organization take over teaching it. We train the trainers, so that the organization can do its own training on the standard. A user interface standard doesn't do all of the work. It provides a useful framework. To improve usability across all software created in the organization, there needs to be training and good information available to anyone creating software. Corporate courses on usability principles (and the connection with the new standard) support a general understanding of the principles underlying the standards effort. Good sources of reference information, such as HFI's Usability Central and a good corporate library collection of usability books, are also important.