3. Place a link back to the homepage on all pages, as students hate to have to hit the back button.
4. Provide a profile of incoming freshmen—SAT scores, GPA, ethnic diversity, etc.—and pictures of current students.
5. Include an overview of the college location and description of the surrounding community.
6. Make the site design distinctive—make sure positioning messages are clear and the homepage isn't generic (students equate generic design with a boring institution).

Whether your Web site is for internal or external use, for e-commerce, or for a university, you should still design it to support the goals of the user. But how do you go about following all of this good advice for creating effective Web sites and avoiding all the pitfalls of bad Web design? By adopting a user-centered design process, of course. Now that we've examined the first steps in this process—understanding users and their goals and creating prototypes to test the concept for the site—we can move to the next step in this process, which is to conduct a heuristic evaluation of the Web site, either in its present form or in the development stage. In Chapter 2, we presented several sets of heuristics, or rules, for evaluating any product. In the next section, we present specific heuristics for the Web.

Usability Heuristics for the Web

Most people who conduct heuristic evaluations are familiar with Jakob Nielsen's 10 usability heuristics (see Chapter 2). Recently, Nielsen and others have clarified these rules to show how they apply to Web usability evaluations. As Table 9.3 on pages 380–381 shows, Nielsen's original list ("Heuristic Evaluation" 30) can be applied to Web reviews, with the help of a few specific comments from two usability experts, Denise Pieratti and Keith Instone.

Pieratti adds other categories to these 10 heuristics that have specific application to the Web (23):

- *Respect for users and their skills.* Does the Web site support, extend, supplement, or enhance a user's skills, background knowledge, and expertise?
- *Pleasurable and respectful interaction.* Do users' interactions with the site enhance the quality of their work life? Does the design reflect the users' role as they navigate through the site? Is the design both visually and functionally pleasing?
- *Quality of work.* Does the user's interaction with the Web site produce usable output?
- *Privacy.* Does the Web site help protect personal and private information and interactions?
- *Readability and legibility.* Is the text readable?
- *Information structure.* Is the information ordered into a hierarchy based on precedence, significance, and frequency of use? Does the underlying structure support intended functions, which are based on user tasks, goals, and needs?

Nielsen weighs in with some comments of his own regarding the original list of 10 heuristics, especially concerning *reset, cancel,* and *undo* options in Web applications ("*Reset* and *Cancel* Buttons"). He contends that the reset and cancel buttons, if removed, would improve usability. Although cancel makes sense in a GUI-based application, because it allows the user to close a dialog box without accepting any changes the user may have made, it does not match what the user does in a Web environment. The same with undo, which works well in editing systems, because it allows the user to revert to a previous state. However, Nielsen contends that the user prefers the back button to escape from any unwanted situation on the Web, making reset and cancel options confusing or redundant. Worse, when the reset button is placed beside the accept button, users can click on the wrong choice and thus lose all the data they may have entered on a form. Cancel can be used sparingly in situations where the user is completing a multipart form, progressing through several pages. In this case, the back button would not undo the information; thus, cancel is useful. An additional button needed in sales transactions is a remove button to delete items placed in a shopping cart. This lets users know they can take things out of their cart without cancelling the order or going back to a previous page.

A special issue of *Technical Communication,* devoted to articles on heuristics for Web communication, focuses on information sites. The researchers whose articles appear in the special issue created five new sets of heuristics for Web development and evaluation, based on the following aspects (van der Geest and Spyridakis 304):

- The rhetorical situation as it is created by authors for and with their readers
- Navigation as a means to signal the information structure of a site and to guide visitors to and through the information
- The presentation of verbal information so that users can comprehend it
- The visual display and presentation of information
- The involvement of users, either directly or indirectly, in the design and evaluation of Web sites

Each of the articles in this special issue covers one aspect of these Web heuristics. Well-researched and tested by both students and Web professionals, the heuristics are complex and detailed. However, each article ends with a checklist, which can be accessed, along with the complete articles, at the journal's Web site, <www.techcomm-online.org>.

As we have discussed in this section, heuristic evaluation provides an excellent way to assess a Web site against a list of rules, or guiding principles. It is one tool in the usability toolkit. In the next section of this chapter, we discuss other tools that provide different kinds of information.

Other Tools for Web Analysis

Other tools for Web analysis fall into one of two categories: (1) computer-based tools that generate data without human intervention and (2) computer-based tools that use humans to provide feedback.

(*continued on page 382*)

TABLE 9.3 Nielsen's 10 Heuristics Adapted for the Web*

Nielsen's heuristics	*Web adaptation*
1. *Visibility of system status.* The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.	Are the URL line and the status line used to provide effective feedback? (P) Most important to users is to know "Where am I?" and "Where can I go next?" That requires branding each page and indicating what section it belongs to. Links to other pages should be clearly marked. (I)
2. *Match between system and the real world.* The system should speak the user's language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.	Does the Web site reflect users' language, tasks, and intentions? (P) Because users come from different backgrounds on the Web, this issue is a challenge. (I)
3. *User control and freedom.* Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.	Can users select and sequence tasks? Can they easily return to where they were if they choose an inappropriate path? (P) Even though many emergency exits are provided by the browser, a "home" button on every page is a simple way to let users feel in control of the Web site. Be careful when forcing certain font choices, colors, screen widths, or browser versions, including the use of "advanced technologies" that users may not have. (I)
4. *Consistency and standards.* Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.	Do the Web pages work with different browsers? Can people resize windows or adjust browser options without compromising information or the task? (P) One of the most common cases of inconsistent wording is with links, page titles, and headers. Inconsistent wording can confuse users when the destination page has a different title from the link that took them there. Standards on the Web mean following HTML and other specifications. Deviations will create opportunities for unusable features to creep into your site. (I)
5. *Error prevention.* Even better than good error messages is a careful design, which prevents a problem from occurring in the first place.	Does the Web site give enough directions and information so that users can find desired pathways and complete desired operations? If an error does occur, can users recover easily? (P) Because of the limitations of HTML forms, inputting information is a common source of user errors. GUI-style widgets, coming into more common use, cut down on the errors, but you still have to doublecheck these after submission.
6. *Recognition rather than recall.* Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.	For the Web, this heuristic is closely related to system status (#1). If users can tell where they are by looking at the current page, they are less likely to get lost. Good labels and descriptive links are crucial for recognition. (I)
7. *Flexibility and efficiency of use.* Accelerators—unseen by the novice user—may often speed up interaction for the expert users to such an extent that the system can cater to both experienced and inexperienced users. Allow users to tailor frequent actions.	Do links take users where they expect to go? Are images and data loaded as effectively as possible? (P) Some of the best accelerators, like bookmarks, are provided by the browser, so pages should be easy to bookmark. Do not use frames in a way that prevents bookmarking. (I)
8. *Aesthetic and minimalist design.* Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.	Are the best media and screen images used to convey a message? (P) Extraneous information on a page is a distraction and a slow-down. Make rarely needed information accessible via a link. Use links for progressive levels of detail. If users jump into the middle of a progression, make sure there's a way to go "up" to get the bigger picture. (I)
9. *Help users recognize, diagnose, and recover from errors.* Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.	Every error message should offer a solution or a link to a solution. For example, if a search yields no hits, don't just tell the user to broaden the search: provide a link that will broaden the search. (I)
10. *Help and documentation.* Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.	Is the site self-documenting? (P) For the Web, the key is to integrate the documentation into your site, either through links to specific help or into each page. (I)

*Pieratti's comments indicated by P; Instone's indicated by I.

Source: Heuristics from Jakob Nielsen, "Heuristic Evaluation." *Usability Inspection Methods.* Eds. Jakob Nielsen and Robert L. Mack. New York: Wiley, 1994, p. 30. © 1994 John Wiley & Sons, Inc. (Reprinted by permission of John Wiley & Sons, Inc.) Web adaptations from Keith Instone, "Usability Heuristics for the Web." Webreview.com 28 June 1999 <http://webreview.com> and Denise D. Pieratti, "Usability and the Web." *Intercom* June 1998, pp. 20–23.