The Universal Access Handbook, C Stepanidis (ed), CRC Press, 2009
www.crcpress.com/product/isbn/9780805862805

# The evaluation of accessibility, usability and user experience


## Helen Petrie

*University of York, Department of Computer Science*

*Heslington, YO10 5DD York*

*United Kingdom*

*Email: petrie@cs.york.ac.uk*

## Nigel Bevan

*Professional Usability Services*

*12 King Edwards Gardens*

*London, W3 9RG*

*United Kingdom*

*Email: nigel@nigelbevan.com*


**Abstract:** This chapter introduces a range of evaluation methods that assist developers in the creation of interactive electronic products, services and environments (eSystems) that are both easy and pleasant to use for the target audience. The target audience might be the broadest range of people, including people with disabilities and older people or it might be a highly specific audience, such as university students studying biology.

The chapter will introduce the concepts of accessibility, usability and user experience as the criteria against which developers should be evaluating their eSystems, and the iterative user-centred design lifecycle as the framework within which the development and evaluation of these eSystems can take place. Then a range of methods for evaluating accessibility, usability and user experience will be outlined, with information about their appropriate use and strengths and weaknesses.

**Keywords:** accessibility, usability, user experience, iterative user-centred design lifecycle, evaluation, evaluation methods

## 1. Introduction

This chapter introduces a range of evaluation methods that allow developers to create interactive electronic systems, products, services and environments[1] that are both easy and pleasant to use for the target audience. The target audience may be the broadest range of people, including people with disabilities and older people, or it may be a highly specific audience, such as university students studying biology. eSystems are also specifically developed for people with particular disabilities to assist them in dealing with the problems they encounter due to their disabilities (commonly such technologies are called *assistive technologies*); these include screen readers for blind computer users and computer-based augmentative and alternative communication systems for people with speech and language disabilities (Cook and Polgar 2008).

The chapter will introduce the concepts of accessibility, usability and user experience as the criteria against which developers should be evaluating their eSystems, and the user-centred iterative design lifecycle as the framework within which the development and evaluation of these eSystems can take place. Then a range of methods for assessing accessibility, usability and user experience will be outlined, with information about their appropriate use and strengths and weaknesses.

## 2. Accessibility, usability and user experience

Developers work to create eSystems that are easy and straightforward for people to use. Terms such as *user friendly* and *easy to use* often indicate these characteristics, but the overall technical term for them is *usability*. The ISO 9241 standard on *Ergonomics of Human System Interaction*[2] (Part 11 1998) defines *usability* as:

> The extent to which a product [service or environment] can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

*Effectiveness* is defined as the accuracy and completeness with which users achieve specified goals; *efficiency* is defined as the resources expended in relation to the accuracy and completeness with which users achieve those goals; and *satisfaction* is defined as "freedom from discomfort, and positive attitudes towards the use of the product [system, service or environment]". Although not components of the ISO definition, many practitioners (Gould and Lewis 1985; Shackel, 1990; 1991; Sharp, Rogers and Preece 2007; Stone et al. 2005) have long considered the following aspects part of *usability*:

> *flexibility:* the extent to which the system can accommodate changes desired by the user beyond those first specified;
>
> *learnability*: the time and effort required to reach a specified level of use performance with the system (also known as *ease of learning*);

---

[1] For ease of reading, we will use the term *eSystems* or simply *systems* to refer to the full range of interactive electronic products, services and environments which includes operating systems, personal computers, applications, websites, handheld devices and so on.

[2] This standard was originally called *Ergonomic Requirements for Office Work with Visual Display Terminals*. A programme of revision and expansion of the standard is currently underway.

> *memorability*: the time and effort required to return to a specified level of use performance after a specified period away from the system; and
>
> *safety*: aspects of the system related to protecting the user from dangerous conditions and undesirable situations.

ISO standards for software quality refer to this broad view of usability as *quality in use*, as it is the user's overall experience of the quality of the product (Bevan 2001).

The discussion above shows that usability is not given an absolute definition, but is relative to the users, goals and contexts of use that are appropriate to the particular set of circumstances. For example, if one is developing an online airline booking system for professional travel agents to use at work, the requirements or criteria for usability components such as efficiency and learnability will undoubtedly be different than if one is developing a website for the general public to book airline tickets. People who use an eSystem on a daily basis for their work will be prepared to put higher levels of time and effort into learning to use the system than those who are using an eSystem only occasionally, however they may also have higher requirements for efficiency.

Like usability, accessibility is a term for which there is a range of definitions. It usually refers to the use of eSystems by people with special needs, particularly those with disabilities and older people. ISO 9241-171 (2008b) defines accessibility as:

> the usability of a product, service, environment or facility by people with the widest range of capabilities

This definition can be thought of as conceptualizing accessibility as simply usability for the maximum possible set of *specified users* accommodated; this fits within the *universal design* or *design for all* philosophy (see section 3.2, below; see also the Chapter in Part I of this Handbook). However, accessibility is also used to refer to eSystems that are specifically usable by people with disabilities. For example, the Web Accessibility Initiative (WAI)[3], founded by the World Wide Web Consortium (W3C) to promote the accessibility of the Web, defines web accessibility to mean:

> that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web (WAI 2006).

The WAI definition suggests that accessibility as a sub-set of usability (i.e. that accessibility is only concerned with issues for a sub-set of users, being older and disabled people), whereas the ISO definition suggests that usability is a sub-set of accessibility (that accessibility is about issues for the largest possible range of users, including older and disabled people). This highlights the current lack of consensus about accessibility. However, for practical purposes, when discussing the development of eSystems for mainstream (i.e. non-disabled, younger) users and the problems that these users have with such systems, usability is the term used; whereas, when the development of eSystems for disabled and older users and the problems these users have with such systems, accessibility is the term used.

*User experience* (often abbreviated to UX) is the newest term in the set of criteria against which an eSystem should be evaluated. It has arisen from the realization that as eSystems

---

[3] www.w3c.org/WAI

become more and more ubiquitous in all aspects of life, users seek and expect more than just an eSystem that is easy to use. Usability emphasises the appropriate achievement of particular tasks in particular contexts of use, but with new technologies such as the Web and portable media players such as iPods, users are not necessarily seeking to achieve a task, but also to amuse and entertain themselves. Therefore the term *user experience,* initially popularized by Norman (1998), has emerged to cover the components of users' interactions with, and reactions to, eSystems that go beyond effectiveness, efficiency, and conventional interpretations of satisfaction.

Different writers have emphasised different aspects of UX: these are not necessarily contradictory to each other, but explore different aspects of and perspectives on this very complex concept. For example, Hassenzahl and Tractinsky (2006, see also Hassenzahl 2006; Hassenzahl, Law and Hvannberg 2006) delineate three areas in which UX goes beyond usability:

- *Holistic*: as previously discussed, usability focuses on performance of and satisfaction with users' tasks and their achievement in defined contexts of use; UX takes a more holistic view, aiming for a balance between task-oriented aspects and other non-task oriented aspects (often called *hedonic* aspects) of eSystem use and possession, such as beauty, challenge, stimulation and self-expression;

- *Subjective*: usability has emphasised objective measures of its components, such as percentage of tasks achieved for effectiveness and task completion times and error rates for efficiency; UX is more concerned with users' subjective reactions to eSystems, their perceptions of the eSystems themselves and their interaction with them;

- *Positive:* usability has often focused on the removal of barriers or problems in eSystems as the methodology for improving them; UX is more concerned with the positive aspects of eSystem use, and how to maximize them, whether those positive aspects be joy, happiness, or engagement.

Dillon (2001), while sharing the view that a move beyond usability is needed in the design and evaluation of eSystems, suggests that an emphasis on three key issues of users' interaction with eSystems is also required:

- *Process*: what the user does, for example navigation through a website, use of particular features, help, etc. This allows the development of an understanding of users' moves, attention and difficulties through an eSystem;

- *Outcomes*: what the user attains, for example what constitutes the goal and end of the interaction. This allows an understanding of what it means for the user to feel accomplishment or closure with the eSystem;

- *Affect*: what the user feels; this includes the concept of satisfaction from the definition of usability, but goes beyond that to include all emotional reactions of users, which might be empowered, annoyed, enriched, or confident. This allows the development of an understanding of users' emotional interaction with eSystems and what interaction means for users.

The new ISO Draft International Standard 9241-210 (2008c) defines UX as:

> A person's perceptions and responses that result from the use or anticipated use of a product, system or service

Bevan (2008) suggests that the definition of usability can be extended to encompass user experience by interpreting satisfaction as including:

- *Likability*: the extent to which the user is satisfied with their perceived achievement of pragmatic goals, including acceptable perceived results of use and consequences of use;
- *Pleasure*: the extent to which the user is satisfied with the perceived achievement of hedonic goals of stimulation, identification and evocation (Hassenzahl 2003) and associated emotional responses, for example Norman's (2004) visceral category;
- *Comfort*: the extent to which the user is satisfied with physical comfort; and
- *Trust*: the extent to which the user is satisfied that the product will behave as intended.

UX is still a concept that is being debated, defined and explored by researchers and practitioners (see, for example, Law et al. 2008). However, it is clear that this concept is already an important part of the evaluation of eSystems and will become more important in the future.

## 3. Design and evaluation processes: iterative user-centred design and inclusive design

In considering when and how to conduct evaluations of eSystems, it is necessary first to situate evaluation within the overall design and development process. Software engineers have long used some form of *waterfall* process of development (see for example, Sommerville 1995) in which phases such as requirements definition, system and software design, implementation and unit testing, integration and system testing, and operation and maintenance are temporally and organizationally distinct. When each phase is complete, a set of documentation summarizing that phase is handed to the next phase in order for it to start. Experts such as Sommerville acknowledge that this is a theoretical idealization, and that in practice adjustment is required between phases, captured in a *spiral* model of development. As Sommerville notes:

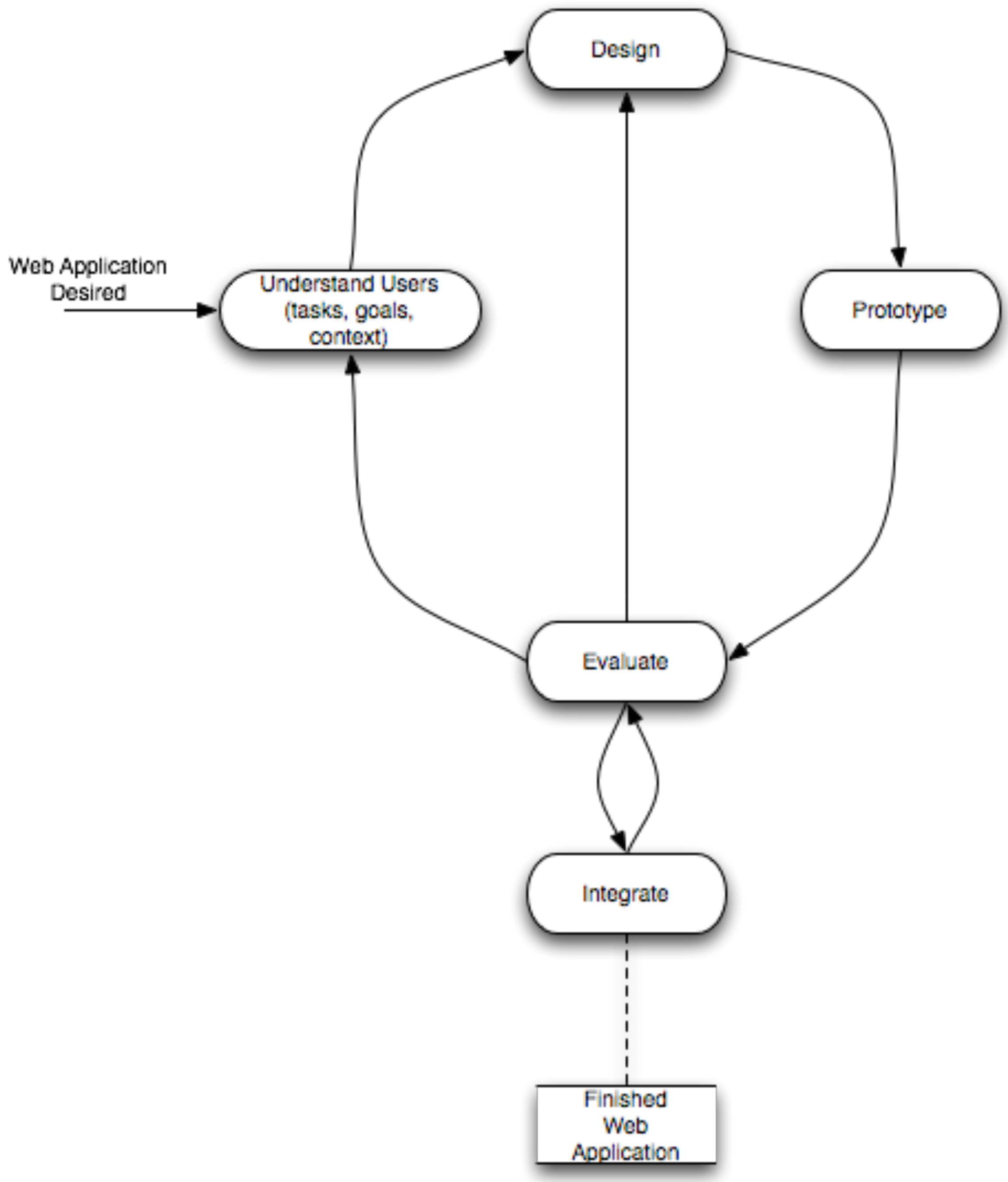
> the development stages overlap … the process is not a simple linear model but involves a sequence of iterations of the development activities (p7).

However, those working on the development of highly interactive eSystems argue that the design and development process must be explicitly iterative and user-centred, to address the difficulties of fully understanding user requirements, and developing eSystems that provide usable and pleasant experiences for users.

### *3.1 Iterative, user-centred design*

A typical iterative user-centred design and development process is illustrated in Figure 1. The phases of the process are as follows:

*Understanding users, tasks, contexts*: This might involve studying existing style guides, guidelines, or standards for particular type of system; interviewing current or potential users of an eSystem about their current system, its strengths and weaknesses, and their expectations for a new or re-designed eSystem; conducting an ethnographic (Ball and Omerod 2000) or context of use (Beyer and Holtzblatt 1997) study of a particular situation. All this contributes to an initial understanding of what the eSystem should do for users and how it should be designed. It is advisable to encapsulate the information gained at this phase in a user requirements document (complementing system requirements documents), that can then be used to track how the subsequent design and development work meets these initial requirements and can be updated to reflect changes in the understanding of the user requirements. A Common Industry Specification for Usability Requirements (CISU-R) has been proposed to provide a standard format for specifying and reporting user requirements and performance and satisfaction criteria (but not UX criteria) (NIST 2007). This specification also proposes formats specifying the context/s of use for an eSystem and test method and context of testing for evaluations.



*Design:* initial design ideas can now be explored. It is often important to explore the design space as much as possible, to consider alternative designs and how they will meet users' needs, rather than immediately settling on one design. This will also facilitate the next stage.

*Prototype:* once an initial potential design, or hopefully a range of potential designs, have been developed, then prototypes can be built (Snyder 2003). These can take many forms, from very simple to complex (often called low fidelity to high fidelity), from sketches on paper with no interactivity, to Microsoft PowerPoint™ or Adobe Flash™ animations with considerable interactivity. In fact, for the initial prototypes, it is usually better to make them obviously simple and unfinished, as that allows people involved in evaluations to realize that it is acceptable to criticize them. Prototypes might also only address part of the functionality of an eSystem, but it is important to explore particular design problems before considerable effort is put into full implementation and integration of components of an eSystem.

In producing prototypes one might realize that some design ideas are not going to be feasible, and this is the first loop of iteration, as it will feed back into the design process.

*Evaluate*: the heart of the process, and the figure, is evaluation. Prototypes can be evaluated by experts and particularly by potential or current users, using a variety of methods (see section 4.3, below). A number of iterations of evaluation, designing and prototyping may be required before acceptable levels of usability, accessibility and user experience are reached. A document that encapsulates the target levels may also be helpful, and again this can be used to track how successive prototypes meet these levels. The evaluations can feed back to both the design phase and to the understanding of the users, their tasks and contexts. Because people are such complex entities, even an eSystem designed on the basis of a very good understanding of users from previous and current work will be unlikely to succeed on the first prototype. As Whiteside, Bennett and Holtzblatt (1988) commented "users always do surprising things" (p799). A number of iterations of prototyping and designing are likely to be required. Nielsen and Sano (1994) reported that in designing a set of icons for the Sun Microsystems website, 20 iterations of the icon designs proved to be necessary. This is quite a high number of iterations, but the main design and development process took only a month, with four main evaluations. Three to five iterations would seem much more typical. It should also be noted that the iterative user-centred design and development of the interactive aspects of an eSystem can usually go on in parallel with back-end developments, so this iterative user-centred process should not hold up the overall development of the eSystem.

*Integration and final implementation*: once the design of the various components of an eSystem has reached acceptable levels of usability, accessibility and user experience, integration of components and final implementation of the interactive systems may be required. Prototypes of eSystems or components may not be implemented in the same language and/or environment as the final eSystem. Once such implementation and integration has taken place, a further evaluation may be appropriate to ensure any issues that relate to using the integrated system are addressed. Finally, once the eSystem is released to users, an evaluation of its use in real contexts may be highly beneficial. Both these final evaluations can feed back into understanding of the users, their tasks and contexts and into the design process, if not for this version of the eSystem, then for subsequent versions.

### *3.2 Inclusive design*

In considering the iterative user-centred design process outlined in the previous section, it should be clear that including people with disabilities and older people amongst the evaluators can be part of this process, and that target levels for accessibility can play an important role in the overall process. This is in contrast with many writers, who only include a consideration of disabled and older people at the end of the design and development process. However, it is clear that if the full range of potential users is considered from the beginning of the process, the overhead of considering the needs of disabled and older users is minimal – it is simply part of the overall design process. On the other hand, if one designs only for young, mainstream users and then attempts to expand the process for disabled and older users at a late stage, one is contradicting the user-centred design process and it is very likely that complex and expensive retro-fitted solutions will be necessary for these users. In some cases, it is impossible to retro-fit a solution to include the needs of particular disabled and older users, and the design process really needs to be started again. For example, in the case of producing an eSystem using Adobe Flash™, if accessibility issues are considered from the beginning, there is no particular additional cost for making the system accessible to disabled users; however, experience has shown that if accessibility is only considered late in the development process, it is almost impossible to retro-fit a solution for disabled users[4].

A number of terms have been coined to cover the inclusion of disabled and older users and their needs in the design process: universal design (a termed coined by Ron Mace, see for example Story, Mueller and Mace 1998), widely used in North America; design for all, used more commonly in Europe (see EDeAN 2007); barrier free design and inclusive design. One difficulty is that all these terms suggest that *all* people should be included, for example universal design is defined as:

> … the design of products and environments to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design.

While this is a honourable aim, it is an ideal to be aimed for, which in practice cannot be met. The literal interpretation can rightly frighten designers and developers, who cannot see how it can be achieved, and may put them off attempting to develop accessible eSystems at all. It is important to get the problem in perspective: designers and developers need to start thinking beyond the needs of young, mainstream, and technology-literate users, and seriously consider the needs of the full range of users who might wish to use the eSystem they are developing.

It is very easy to fail to recognize the full range of users who might be interested in using or need to use a particular eSystem. For example, in developing an online payment system for a road taxing system, designers might think that only drivers, who by definition have good sight, will be interested in or need to use the system. However, a visually disabled friend of a driver may wish to pay the road tax when they are given a lift. Therefore, such an eSystem needs to be accessible to users with visual disabilities as well as fully sighted ones.

In addition, designers need to be aware that people with visual disabilities in particular will use assistive technologies to help them access many eSystems, particularly if they are accessing them in the workplace or at home (the situation for eSystems to be used in public places, such as automatic banking machines and ticket machines is more problematic, if alternatives are not available). This includes screen readers used by blind people (see the chapter on "Screen Readers"), screen magnification programs used by partially sighted people and a variety of alternative input devices used by people with physical disabilities (Cook and Polgar 2008). This means that the designers of a particular eSystem do not need to solve *all* the accessibility problems.

---

[4] This is not a criticism specifically made of Adobe Flash™, as Adobe have worked diligently to make it accessible, this situation holds for many technologies.

One can think of the population of users addressed by an eSystem as dividing into three groups. For users who do not use an assistive technology in the context of use, as many users as possible should be accommodated (this will include mainstream and older users); for users who do use an assistive technology, the system should work smoothly with assistive technologies (and will need evaluation with those assistive technologies to ensure that is the case); the final group – people who cannot use the system with or without an assistive technology, should ideally be an empty set.

Some universal design/design for all approaches propose guidelines to assist in the design of eSystems to meet the needs of disabled and older users. Such guidelines are indeed useful, and will be discussed in section 4.2, below. However, both the use of universal design/design for all guidelines and evaluation with disabled and older users should be integrated into the iterative user-centred design process for the most effective development of eSystems that are usable and pleasant for the widest range of users.

## 4. Methods for evaluation

Methods for usability, accessibility and UX evaluation can be grouped into the following categories:

- Automated checking of conformance to guidelines and standards
- Evaluations conducted by experts
- Evaluations using models and simulations
- Evaluation with users or potential users
- Evaluation of data collected during eSystem usage

Several methods are based on the use of guidelines and standards, so an outline and discussion of relevant guidelines and standards will be given first, and then outlines and discussions of the various methods will be presented.

### *4.1 Guidelines for accessibility and usability*

#### *4.1.1 Accessibility Guidelines*

Guidelines on accessibility for disabled and older users are available for a number of different types of eSystems. For example, the IBM Human Ability and Accessibility Centre[5] provides guidelines in the form of easy to follow checklists with hyperlinks to rationales to explain the need for the guideline, development techniques and testing methods. There are checklists for:

- Software
- Websites and applications
- Java applications
- Lotus notes

[5] http://www-03.ibm.com/able/

- Hardware
- Documentation

Three ISO standards containing accessibility guidelines are likely to be published in their final form in 2008 (see also the chapter on "eAccessibility Standardization"):

- ISO/IEC[6] 10779 (2008d): Office equipment accessibility guidelines for elderly persons and persons with disabilities
- ISO 9241-20 (2008a): Ergonomics of human-system interaction – Part 20: Accessibility guidelines for information/communication technology (ICT) equipment and services
- ISO 9241-171 (2008b): Ergonomics of human-system interaction – Part 171: Guidance on software accessibility

The key set of guidelines for assessing the accessibility of websites is the Web Content Accessibility Guidelines developed by the WAI (see the chapter "Accessing the Web"). The first version of these Guidelines (WCAG1) was published in 1999 (WAI 1999). The second version of WCAG (WCAG2, see WAI 2008) is currently a W3C Candidate Recommendation and it is expected to be a W3C Recommendation shortly. However it is expected that both WCAG1 and WCAG2 will be used in parallel for some time.

WCAG1 includes 14 high level accessibility guidelines, which are broken down into 65 specific checkpoints. Each checkpoint is assigned a priority level (Priority 1, 2 and 3). A web page or document must satisfy Priority 1 (P1) checkpoints, otherwise, according to WAI "one or more groups [of disabled people] will find it *impossible* to access information in the document". If Priority 2 (P2) checkpoints are not satisfied, one or more groups of disabled people will find it *difficult* to access information in the document. If Priority 3 (P3) checkpoints are not satisfied, one or more groups of disabled people "will find it *somewhat difficult* to access information". If a webpage or site passes all the P1 checkpoints, it is said to be Level A conformant; if it passes all P1 and all P2 checkpoints, it is Level AA conformant; finally is it passes all P1, P2 and P3 checkpoints, it is Level AAA conformant.

WCAG2 carries forward many of the ideas of WCAG1, including the three levels of conformance (the complexity of Priorities and Levels has been removed, so only the three levels A, AA and AAA are now used). However, rather than being organized around the 14 high level guidelines, it is now organized around four accessibility principles:

- Content must be perceivable
- Interface components in the content must be operable
- Content and controls must be understandable
- Content should be robust enough to work with current and future user agents (including assistive technologies)

---

[6] This standard has been developed jointly by the International Standards Organization and the International Electrotechnical Commission (IEC).

**Table 1** Summary of Web Content Accessibility Guidelines Version 2.0 (WAI, 2008)

| Principle | Guidelines |
|---|---|
| 1. Perceivable - Information and user interface components must be presentable to users in ways they can perceive | 1.1 Text Alternatives: Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language. |
| | 1.2 Time-based Media: Provide alternatives for time-based media |
| | 1.3 Adaptable: Create content that can be presented in different ways (for example simpler layout) without losing information or structure |
| | 1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background |
| 2: Operable - User interface components and navigation must be operable | 2.1 Keyboard Accessible: Make all functionality available from a keyboard |
| | 2.2 Enough Time: Provide users enough time to read and use content |
| | 2.3 Seizures: Do not design content in a way that is known to cause seizures |
| | 2.4 Navigable: Provide ways to help users navigate, find content and determine where they are |
| 3: Understandable - Information and the operation of user interface must be understandable | 3.1 Readable: Make text content readable and understandable |
| | 3.2 Predictable: Make Web pages appear and operate in predictable ways |
| | 3.3 Input Assistance: Help users avoid and correct mistakes |
| 4: Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies | 4.1 Compatible: Maximize compatibility with current and future user agents, including assistive technologies |

Each principle is associated with a list of guidelines addressing the issues around that principle. Many of the checkpoints from WCAG1 are retained, but the organization is more logical.

Another set of guidelines often mentioned in relation to web accessibility is Section 508 of the Rehabilitation Act of the United States Federal government (see the chapter on "Policy and Legislation as a Framework of Accessibility"). In fact, this legislation requires Federal agencies to make all their electronic and information technologies (not only websites) accessible to people with disabilities. In practice, this means that anyone who is supplying eSystems to the USA Federal government is obliged to make them accessible. A set of standards have been developed to specify what accessibility means for different types of eSystems[7], and those for websites are very similar to WCAG.

Accessibility guidelines and advice also exist for many specific technologies that are used for producing eSystems and for specific domains. For example, in relation to specific technologies:

Adobe Flash™ - see Regan (2004) and resources at the Adobe Accessibility Resource Centre[8]

Content Management Systems

Joomla! – see O'Connor (2007)

Eclipse open source development platform – see Accessibility Features in Eclipse[9] and the Eclipse Accessibility Tools Framework Project[10]

Java - see resources at Sun Accessibility[11]

Microsoft™ products – resources at the Microsoft Accessibility Developer Center[12]

*4.1.2 Usability Guidelines and Standards*

Guidelines and standards for ensuring good usability of eSystems have been developed for many years. They range from the high level guidelines (or heuristics) proposed by Nielsen and Molich (Molich and Nielsen 1990; Nielsen and Molich 1990; see also Nielsen, 1994) and Shneiderman (Shneiderman and Plaisant 2005) (see Tables 2 and 3, respectively) to the much more detailed guidelines provided by the ISO 9241 standard. Parts 12-17 of the ISO 9241 series of standards contain very detailed user interface guidelines. Although these are an excellent source of reference, they are very time consuming to employ in evaluations. A further problem with the detailed ISO guidelines is that they need to be interpreted in relation to a particular interface environment (such the Microsoft Windows operating system or the Web).

---

[7] See http://www.section508.gov/
[8] http://www.adobe.com/accessibility/
[9] http://help.eclipse.org/help33/index.jsp?topic=
/org.eclipse.platform.doc.user/concepts/accessibility/accessmain.htm
[10] http://www.eclipse.org/actf/
[11] http://www.sun.com/accessibility/
[12] http://msdn.microsoft.com/en-us/accessibility/default.aspx

**Table 2** Neilsen's usability heuristics (Neilsen, 1994)

| | |
|---|---|
| **Visibility of system status** | The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. |
| **Match between system and the real world** | The system should speak the users' 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. |
| **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. |
| **Consistency and standards** | Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. |
| **Error prevention** | Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. |
| **Recognition rather than recall** | Minimize the user's memory load by making 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. |
| **Flexibility and efficiency of use** | Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. |
| **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. |
| **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. |
| **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. |

**Table 3** Shneiderman's 8 golden principles of good interface design (see Shneiderman and Plaisant, 2005)

| | |
|---|---|
| **Strive for consistency** | Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout. |
| **Enable frequent users to use shortcuts** | As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user. |
| **Offer informative feedback** | For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial. |

| | |
|---|---|
| **Design dialogue to yield closure** | Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions. |
| **Offer simple error handling** | As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error. |
| **Permit easy reversal of actions** | This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions. |
| **Support internal locus of control** | Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders. |
| **Reduce short-term memory load** | The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions. |

Detailed guidelines for web design are also available. The most comprehensive, well-researched and easy to use set has been produced by the U.S. Government Department of Health and Human Services (HHS) (2006). This provides 207 guidelines derived from about 500 cited publications. Each guideline contains:

- A brief statement of the overarching principle that is the foundation of the guideline.
- Comments that further explain the research/supporting information.
- Citations to relevant web sites, technical and/or research reports supporting the guideline.
- A score indicating the "Strength of Evidence" that supports the guideline
- A score indicating the "Relative Importance" of the guideline to the overall success of a web site.
- One or more graphic examples of the guideline in practice.

Some examples of the guidelines are presented in Table 4.

While no set of guidelines can be totally comprehensive, the HHS guidelines appear to be more complete and easier to use than the equivalent ISO standard 9241-151 (Bevan and Spinhof 2007).

**Table 4** Research-Based Web Design and Usability Guidelines (U.S. Department of Health and Human Science, 2006)

| **Category** | **Example** |
|---|---|
| 1 Design process and evaluation | Establish user requirements |
| 2 Optimizing the user experience | Provide printing options |
| 3 Accessibility | Do not use color alone to convey information |
| 4 Hardware and software | Design for common browsers |
| 5 The homepage | Show all major options on the homepage |
| 6 Page layout | Set appropriate page lengths |
| 7 Navigation | Provide feedback on users' location |
| 8 Scrolling and paging | Eliminate horizontal scrolling |
| 9 Headings, titles, and labels | Use unique and descriptive headings |
| 10 Links | Provide consistent clickability cues |
| 11 Text appearance | Use black text on plain, high-contrast backgrounds |
| 12 Lists | Order elements to maximize user performance |
| 13 Screen-based controls (widgets) | Distinguish required and optional data entry fields |
| 14 Graphics, images, and multimedia | Use video, animation, and audio meaningfully |
| 15 Writing web content | Avoid jargon |
| 16 Content organization | Facilitate scanning |
| 17 Search | Ensure usable search results |

*4.1.3 Problems with guidelines*

Although guidelines would appear to provide objective criteria against which to evaluate a system, they present a number of difficulties.

- The large number of guidelines require substantial effort to learn and apply appropriately.

- For a thorough evaluation, every page or screen should be evaluated against every applicable guideline, which would be very time consuming. Selecting representative screens or pages may miss some issues.
- Following guidelines usually improves an eSystem, but they are only generalizations so there may be particular circumstances where guidelines conflict or do not apply (for example, because of the use of new features not anticipated by the guideline).
- It is difficult to apply guidelines appropriately without also having expertise in the application domain, and for accessibility guidelines, expertise in accessibility. For example, Petrie et al. (2006) reported that due to lack of experience with disabled people and their technologies, developers often do not have the conceptual framework needed to apply disability-related guidelines.

Evaluation of the characteristics of the user interface can anticipate and explain many potential usability and accessibility problems, and can be carried out before a working system is available. However, evaluation of detailed characteristics alone can never be sufficient, as this does not provide enough information to accurately predict the eventual user behaviour.

To be sure of product usability/accessibility requires user testing. Although a user test is the ultimate test of usability and accessibility, it is not usually practical to evaluate all permutations of user type, task, and environmental conditions.

A number of guidelines sets include ratings of the importance of different guidelines. As discussed in section 4.1.1, WCAG1 and WCAG2 include three levels of priority that indicate their importance in relation to the accessibility of websites for people with disabilities. As discussed in section 4.1.2, the HHS guidelines also provide a rating for the "Relative Importance" of the guideline to the overall success of the Web site. Few studies have investigated the validity of these ratings, but two recent studies (Harrison and Petrie 2006; Petrie and Kheir 2007) have found no correlation in the ratings given by both disabled and mainstream users of actual problems that they have encountered and the ratings of those problems as given by WCAG1 and HHS. Therefore, the ratings need to be treated with a certain amount of caution and further studies of this issue are required.

## ***4.2 Automated checking of conformance to guidelines or standards***

### *4.2.1 When to use automated checking*

When initial prototypes or initial versions of full implementations are available.

### *4.2.2 Why use automated checking*

To ensure that initial prototypes and initial versions of final implementations meet appropriate guidelines and standards and do not contain basic accessibility and usability problems.

### *4.2.3 Tools for automated accessibility checking*

The development of WCAG provided a considerable interest in creating tools to automatically check whether websites and pages conform with the guidelines, as many of the WCAG checkpoints seemed amenable to such automated checking. One of the first such tools, and the most well-known, the Bobby Tool, is no longer available, although it is still mentioned in the literature. A comprehensive list of these tools is maintained on the WAI website[13]. There appear to be no automatic accessibility checking tools for other types of eSystems.

Although automated accessibility checking has its role in the evaluation of websites, its strengths and weaknesses need to be understood. Many WCAG Checkpoints cannot be checked automatically, and for example only 23% of the WCAG checkpoints were checked by the Bobby tool (Cooper and Rejmer 2001). Even a single checkpoint may require several tests to check whether it has been passed, some of which can be automated and some of which cannot.

The well-known WCAG1 Checkpoint 1.1: Provide a text equivalent for every non-text element, provides an example. An automated tool can check whether there is an alternative description on every image, which can be a very useful function in evaluating a large website with many images. However, no automatic tool can check whether the alternative descriptions are accurate and useful (Petrie, Harrison and Dev 2005). So all the following alternative descriptions which have been found should fail to meet Checkpoint 1.1, but an automatic checking tool would pass them:

- Blah blah blah (possibly a forgotten placeholder?)
- Image of an elephant (the image was actually of a cat)
- Person at computer (gave no indication of why the image was included)

Thus, particular care needs to be taken in interpreting what it means when an automatic checking tools returns no failures for a particular WCAG Checkpoint. This means that there have been no failures for the tests that the tool has been able to make on the Checkpoint, but not necessarily that there are no failures at all for the Checkpoint. Unfortunately, many tools fail to make clear what tests they are conducting on each checkpoint, so it is difficult for those using such tools to evaluate the accessibility of websites to accurately assess their output.

No automated tools exist for automated accessibility checking of eSystems other than websites and pages.

#### *4.2.4 Tools for automated usability checking*

There are some automated tools that automatically test for conformance with basic usability guidelines. A review of tools by Brajnik (2000) found that the commercial LIFT[14] tool (which also covers accessibility) made the most comprehensive range of checks, with the free Web Static Analyzer Tool (WebSAT[15]) being the next most effective. Although these tools are useful for screening for basic problems, they only test a very limited scope of usability issues (Ivory and Hearst 2001).

### ***4.3 Evaluations conducted by experts***

#### *4.3.1 When to use expert evaluations*

When initial prototypes are available.

[13] http://www.w3.org/WAI/ER/existingtools.html#General
[14] http://www.nngroup.com/reports/accessibility/software/
[15] http://zing.ncsl.nist.gov/WebTools/WebSAT/overview.html

*4.3.2 Why conduct expert evaluations*

- To identify as many accessibility and usability issues as possible in order to eliminate them before conducting user-based evaluations;
- because there are too many pages or screens to include all of them in user-based evaluations;
- because it is not possible to obtain actual or potential users for evaluations;
- because there is insufficient time for user testing; and
- to train developers in accessibility and usability issues.

Expert-based methods ask one or more accessibility/usability or domain experts to work through an eSystem looking for accessibility or usability problems. These experts can use guidelines or they can work through task scenarios that represent what users would typically do with an eSystem (see section 4.4, below), see Table 5 (derived from Gray and Salzman 1998). Usability methods that do not use task scenarios are referred to as reviews or inspections, while task-based evaluations are referred to as walkthroughs. For accessibility, expert-based methods have tended not to include task scenarios (although there is no reason why they should not), but are divided into preliminary accessibility reviews, which use a small number of key guidelines, and conformance evaluations for accessibility, which use the full set of WCAG1 or WCAG2.

**Table 5** Types of expert-based evaluation method

| **Guidelines** | **Task scenarios** | |
|---|---|---|
| | No | Yes |
| None | Expert review | Usability walkthrough<br>Pluralistic walkthrough<br>Cognitive walkthrough |
| General guidelines | Heuristic inspection<br>Preliminary accessibility review | Heuristic walkthrough |
| Detailed guidelines | Guidelines inspection<br>Conformance evaluation for accessibility | Guidelines walkthrough |

### 4.3.3 Heuristic evaluation

The most popular type of expert evaluation is *heuristic evaluation* (Nielsen 1994). Heuristic evaluation originally involved a small set of evaluators examining each eSystem element in order to identify potential usability problems. The evaluators use a set of heuristics (such as those in Table 2) to guide them and rate the potential problems for how severe they are or how important they are to eliminate before an eSystem is released. Usually, a four level rating scheme is used (1 = Cosmetic problem only: need not be fixed unless extra time is available on project; 2 = Minor usability problem: fixing this should be given low priority; 3 = Major usability problem: important to fix, so should be given high priority; 4 = Usability catastrophe: imperative to fix this before product can be released). Several evaluators are usually involved, as each individual typically only finds about one third of the problems (Nielsen 1994). Although heuristic evaluation can be carried out by people who are not trained in usability methods, better results are obtained by trained experts (Desurvire, Kondziela and Atwood 1992).

Since it was originally proposed by Nielsen, heuristic evaluation has been adapted in many ways. Rather than inspecting individual elements, it is often carried out by asking the evaluator to step through typical user tasks. This can combine heuristic evaluation with some of the benefits of a cognitive walkthrough. Evaluators may also work as a group, identifying potential problems together, but then rating them individually and privately, so that they influence each others ratings. This is known as a Cello evaluation[16]. The relative effectiveness of different forms of heuristic evaluation have not yet been explored.

It is often difficult to get agreement between the evaluators on exactly which heuristic is associated with a particular potential usability problem. Heuristics are a useful training aid (Cockton et al. 2003), but their value to usability experts is not clear. Experienced evaluators often dispense with assessing problems against specific heuristics, preferring to rely on their understanding of the principles and their experience of observing users encountering problems.

### *4.3.4 Expert Walkthrough Evaluations*

A usability walkthrough evaluation identifies problems while attempting to achieve tasks in the same way as a user, making use of the expert's knowledge and experience of potential problems. This is usually more effective than inspecting individual pages in isolation, as it takes account of the context in which the user would be using the eSystem.

Variations include:

*Cognitive walkthrough*: for each user action, the evaluator analyses what the user would be trying to do, whether the interface supports the user's next step to achieve the task, and whether appropriate feedback is provided (Wharton et al. 1994). Although the original method required detailed documentation, it is often used in a more lightweight fashion (e.g. Spencer 2000).

*Pluralistic walkthrough*: a group of users, developers, and human factors people step through a scenario, discussing each dialogue element (Bias 1994).

---

[16] http://www.ucc.ie/hfrg/emmus/methods/cello.html

*4.3.5 Expert evaluations for accessibility*

Two expert evaluation methods exist for assessing websites and web-based applications for accessibility and will be outlined. No comparable methods exist for assessing the accessibility of other types of eSystems; this could be done on an ad hoc basis by an expert, or it would be preferable to employ user evaluation instead (see section 4.5).

Conducting a full expert evaluation of a website or web-based application for conformance to the WCAG1 or WCAG2 guidelines is a considerable undertaking. So an initial step is to undertake a *preliminary accessibility review*[17]. This involves:

- selecting a representative set of pages from the website or screen from the application
- test the pages/screens with a range of graphical browsers (e.g., Internet Explorer, Firefox), making the following adjustments:
  - turn off images, and check whether appropriate alternative text for the images is available
  - turn off the sound, and check whether audio content is still available through text equivalents
  - use browser controls to vary font-size: verify that the font size changes on the screen accordingly, and that the page is still usable at larger font sizes
  - test with different screen resolution, and/or by resizing the application window to less than maximum, to verify that horizontal scrolling is not required
  - change the display colour to greyscale (or print out page in greyscale or black and white) and observe whether the colour contrast is adequate
  - without using the mouse, use the keyboard to navigate through the links and form controls on a page (for example, using the "Tab" key), making sure that all links and form controls can be accessed, and that the links clearly indicate what they lead to.

A number of browser extensions and plug-in evaluation tools are available to make conducting these tests more efficient, for example the AIS Toolbar[18] for Internet Explorer (available in a wide range of languages) and Opera (currently available in English only), the Accessibar add-on[19] and WAVE Toolbar for Firefox[20].

- test the pages/screens using a specialised browser, such as a text browser (e.g., Lynx[21]), or a screen reader such as JAWS[22] or WindowEyes[23]. Screen readers are sophisticated programs with considerable functionality – an expert user, whether sighted or blind, is needed to use these programs effectively.

---

[17] http://www.w3.org/WAI/eval/preliminary.html
[18] http://www.visionaustralia.org.au/ais/toolbar/
[19] http://firefox.cita.uiuc.edu/
[20] http://wave.webaim.org/toolbar
[21] http://lynx.browser.org/
[22] http://www.freedomscientific.com/fs_products/software_jaws.asp
[23] http://www.gwmicro.com/Window-Eyes/

The results of such a preliminary accessibility review can guide the further development of a website or web-based application. Once the website or web-based application has been developed, it is important to undertake a full accessibility audit. The WAI has outlined the methodology for undertaking such an audit[24], which is similar to the methodology for the preliminary accessibility review, but includes manual checking of all applicable WCAG checkpoints.

A group of European Union funded projects[25] has developed a detailed standard methodology for the expert accessibility evaluation of websites, the Unified Web Evaluation Methodology (UWEM). Complete details on how to conduct a UWEM evaluation can be found on their website[26], and covers not only the evaluation procedures, but also statistical methods for sampling, critical path analysis, computer assisted content selection, manual content selection and interpretation and the aggregation and integration of test results.

The WAI have developed a standard reporting format for accessibility evaluation reports that is also used by the UWEM. Details of this are available on the UWEM and WAI[27] websites.

#### *4.3.6 Advantages and disadvantages of expert evaluation*

Expert usability evaluation is simpler and quicker to carry out than user-based evaluation and can, in principle, take account of a wider range of users and tasks than user-based evaluation, but it tends to emphasize more superficial problems (Jeffries and Desurvire 1992) and may not scale well for complex interfaces (Slavkovic and Cross 1999). To obtain results comparable with user-based evaluation, the assessment of several experts must be combined. The greater the difference between the knowledge and experience of the experts and the real users, the less reliable are the results.

### *4.4 Evaluations using models and simulations*

#### *4.4.1 When to use evaluations using models and simulations*

- When models can be constructed economically
- When user testing is not practical

#### *4.4.2 Why use evaluations using models and simulations*

- If time to complete tasks is critical, e.g., for economic or safety reasons

Model-based evaluation methods can predict measures such as the time to complete a task or the difficulty of learning to use an interface. Some models have the potential advantage that they can be used without the need for any prototype to be developed. Examples are the use of Keystroke Level Model (Mayhew 2005), the Goals Operators Methods and Selections (GOMS) model and the ACT-R model of human cognitive processes (St. Amant, Horton and Ritter 2007). However, setting up a model currently usually requires considerable effort, so model-based methods are cost effective in situations where other methods are impracticable, or the information provided by the model is a cost-effective means of managing particular risks.

---

[24] http://www.w3.org/WAI/eval/conformance.html
[25] The Web Accessibility Cluster, see www.wabcluster.org
[26] http://www.wabcluster.org/uwem1_2/
[27] http://www.w3.org/WAI/eval/template.html

Further information on the use of models can be found in Pew and Mayor (2007).

### *4.5 Evaluations with users*

#### *4.5.1 When to use*

- At all stages of development, if possible
- At the final stage of development, at least

#### *4.5.2 Why conduct user-based evaluations*

- To provide evidence of the accessibility and usability of an eSystem in real use by the target audience
- To provide evidence of accessibility and usability (or lack thereof) for developers or management

#### *4.5.3 Types of user-based evaluations*

In user-based methods, target users undertake realistic tasks which the eSystem is designed to support in realistic situations, or as realistic situations as possible. A minimum of assistance is given by those running the evaluation, except when participants get completely stuck or need information not readily available to them.

There are many practical details of planning and executing user evaluations, with excellent explanations in books such as Rubin and Chisnell (2008) and Dumas and Redish (1999) and the chapter by Lewis (2005). The interested reader is recommended to study one of these before undertaking user evaluations.

There are different types of user-based methods adapted specifically for formative and summative evaluations (see Table 6):

- Formative methods focus on understanding the user's behaviour, intentions and expectations in order to understand any problems encountered, and typically employ a "think-aloud" protocol;
- Summative methods measure the product usability or accessibility, and can be used to establish and test user requirements. Summative usability testing may be based on the principles of ISO 9241-11 and measure a range of usability components such as effectiveness, efficiency and satisfaction. Each type of measure is usually regarded as a separate factor with a relative importance that depends on the context of use.

#### *4.5.4 Selecting the sample size*

While the cost benefits of usability evaluation are well established (Bias and Mayhew 2005), there is no way to be sure that all the important usability problems have been found by an evaluation.

Deciding how many participants to include in formative evaluations depends on the target percentage of problems to be identified, and the probability of finding problems (Lewis 2006). Usability test sample size requirements for a particular desired percentage of problems can be estimated by calculating the probability of finding problems either based on previous similar usability evaluation, or from initial results of an ongoing study. A recent survey (Hwang and Salvendy 2007) found probabilities in the range 0.08 to 0.42. This would correspond to evaluating with between 3 and 19 participants to find 80% of the problems, or between 4 and 28 participants to find 90% of the problems. Complex websites and web-based applications in which different users may explore different aspects are likely to have lower probabilities.

Iterative testing with small numbers of participants is preferable, starting early in the design and development process (Medlock et al, 2002). If carrying out a single, user-based evaluation late in the development lifecycle (this is not the best procedure, as evaluations should be conducted on several iterations), it is typical to test with at least 8 users (or more if there are several distinct target user types).

**Table 6** Purposes of user-based evaluation

| **Purpose** | **Description** | **When in Design Cycle** | **Typical Sample Size (per group)** | **Considerations** |
|---|---|---|---|---|
| **Early formative evaluations** | | | | |
| Exploratory | High level test of users performing tasks | Conceptual design | 5-8 | Simulate early concepts, for example with very low fidelity paper prototypes. |
| Diagnostic | Give representative users real tasks to perform | Iterative throughout the design cycle | 5-8 | Early designs or computer simulations. Used to identify usability problems. |
| Comparison | Identify strengths and weaknesses of an existing design | Early in design | 5-8 | Can be combined with benchmarking. |
| **Summative usability testing** | | | | |
| Benchmarking/ Competitive | Real users and real tasks are tested with existing design | Prior to design | 8-30 | To provide a basis for setting usability criteria. Can be combined with comparison with other eSystems. |
| Final | Real users and real tasks are tested with final design | End of design cycle | 8-30 | To validate the design by having usability objectives as acceptance criteria and should include any training and documentation. |

For summative evaluation, the number of participants depends on the confidence required in the results (i.e. the acceptable probability that the results were only obtained by chance). If there is little variance in the data, a sample of as few as 8 participants of one type may be sufficient. If there are several types of users, other sources of variance, or if success rate is being measured (see ISO 20282-2, 2006), 30 or more users may be required.

*4.5.5 Conducting evaluations with disabled and older users*

There are a number of issues related to conducting evaluations with disabled and older users that need to be raised. It is appreciated that finding samples of disabled and older people willing and able to take part in evaluations is not easy (Petrie et al. 2006) and it may be that remote evaluations could be used to overcome this problem. Petrie et al. (2006) discuss the advantages and disadvantages of remote evaluations. Another method of overcoming this issue might seem to be using able-bodied users to simulate users with disabilities, for example by blindfolding people to simulate visual impairment. This is not a sensible solution to the issue at all – people who are visually impaired have developed strategies to deal with their situation, suddenly putting sighted people into a blindfolded situation is not at all comparable. Designers and developers may gain some useful insight into the situations of disabled and older users by experiencing simulations, but the usefulness and ethics of these are debated and highly controversial (Burgstahler and Doe 2004; Kiger 1992).

An important issue to consider when conducting evaluations with disabled users is whether they will use assistive technologies in using the eSystem under evaluation. If so, the correct versions and the preferred configurations of assistive technologies need to be provided for participants in the evaluation to ensure that the results of evaluations are valid. This can be an expensive and time-consuming undertaking. Again, if a suitable range of assistive technologies in not available in a testing laboratory, it may be easier to undertake the evaluation via remote testing (Petrie et al. 2006).

Finally, if evaluations are undertaken with disabled and older people, it is important that the needs of participants in the evaluation are taken carefully into consideration. Personnel running the evaluations need to be sensitive to the needs of the particular groups, such as visually disabled people, people in wheelchairs etc. Local organizations of disabled people can often provide training in disability awareness that can be very helpful to undertake before embarking on such evaluations. Issues to consider include:

- How will the participants come to the evaluation location (is public transport accessible, is the building easy to find for someone visually impaired)?
- Is the location itself accessible for the participants (e.g. are appropriate toilet facilities available, is there somewhere for guide dogs to exercise etc.)?
- Are explanatory materials and consent forms available in the appropriate alternative formats (e.g. large print, Braille, Easy Read)?
- Will the pace of the evaluation be suitable for the participants (e.g. older participants may appreciate a slower pace of evaluation)?

*4.5.6 Evaluating user satisfaction and user experience in user-based evaluations*

As noted in section 2, satisfaction and user experience move beyond performance-based measures that have traditionally been the focus of user-based evaluations. These aspects of the evaluation of eSystems can be assessed in a variety of ways, for example using Kansei techniques from consumer product development (Schütte et al. 2004). However, the simplest way is with rating scales and questionnaires. Psychometrically designed questionnaires, for example SUS, for usability (Brooke 1996), or AttrakDiff (Hassenzahl et al. 2003) for user experience, will give more reliable results than ad hoc questionnaires (Hornbaek 2006). See Hornbaek (2006) for examples of other validated questionnaires.

### *4.6 Evaluation of data collected during eSystem usage*

#### *4.6.1 When to use data during eSystem usage*

- When planning to improve an existing eSystem

#### *4.6.2 Why use data druing eSystem usage*

- Provides non-intrusive data about the use of a current eSystem.

#### *4.6.3 Satisfaction Surveys*

Satisfaction questionnaires distributed to a sample of existing users provide an economical way of obtaining feedback on the usability of an existing eSystem.

#### *4.6.4 Web server log analysis*

Web-based logs contain potentially useful data that can be used to evaluate usability by providing data such as entrance and exit pages, frequency of particular paths through the site, and the extent to which search is successful (Burton and Walther 2001). However, it is very difficult to track and interpret individual user behaviour (Groves 2007) without some form of page tagging combined with pop-up questions when the system is being used, so that the results can be related to particular user groups and tasks.

#### *4.6.5 Application Instrumentation*

Data points can be built into code that count when an event occurs, for example in Microsoft Office (Harris 2005). This could be the frequency with which commands are used or the number of times a sequence results in a particular type of error. The data is sent anonymously to the development organization. This real-world data from large populations can help guide future design decisions.

## 5. Conclusion

This chapter has introduced a range of evaluation methods which assist developers in the creation of eSystems that are both accessible and usable. There are many details in the use of these techniques that the interested reader will need to follow up, hopefully the many references and the further reading provided at the end of the chapter will provide further information on particular techniques. With all these methods, the best results will be obtained by someone experienced in their use. However, anyone planning to embark on evaluation should not be put off by the complexity of the situation, but should start with some of the simpler expert evaluations and simple user-based evaluations. With practice and patience, expertise in these areas can be developed.

## References

Ball, L. and Omerod, T. 2000. Putting ethnography to work: the case for a cognitive ethnography of design. International Journal of Human-Computer Studies 53(1): 147 – 168.

Bevan, N. 2001. Quality in use for all. In User Interfaces for All: methods, concepts and tools, Ed. C. Stephanidis, 353-368. Mahwah, NJ: Lawrence Erlbaum.

Bevan, N. 2008. Classifying and selecting UX and usability measures. In the Proceedings of Meaningful Measures: Valid Useful User Experience Measurement (VUUM), 5th COST294-MAUSE Open Workshop, 18th June 2008, Reykjavik, Iceland.

Bevan, N. and Spinhof, L. 2007. Are guidelines and standards for web usability comprehensive? In Human-Computer Interaction – Interaction Design and Usability (Part I), Volume 1 of the HCI International 2007 Conference Proceedings (LNCS 4550), Ed. J. Jacko, 407–419. Berlin Heidelberg: Springer-Verlag.

Beyer, H. and Hotzblatt, K. 1997. Contextual design: defining customer-centred systems. San Francisco, CA: Morgan Kaufmann.

Bias, R. G. 1994. Pluralistic usability walkthrough: coordinated empathies. In Usability Inspection Methods, Eds. J. Nielsen and R.L. Mack, 63-76. New York, NY: Wiley and Sons, Inc.

Bias, R.G. and Mayhew, D.J. (Eds.). 2005. Cost-Justifying Usability: An Update for the Internet Age. San Francisco: Morgan Kaufmann.

Brajnik, G. 2000. Automatic web usability evaluation: what needs to be done? In the Proceedings of 6th Human Factors and the Web Conference 2000. Available at: users.dimi.uniud.it/~giorgio.brajnik/papers/hfweb00.html

Brooke, J. 1996. SUS: A “quick and dirty” usability scale. In Usability Evaluation in Industry, Eds. P. Jordan, B. Thomas, and B. Weerdmeester, 189-194). London: Taylor and Francis.

Burgstahler, S. and Doe, T. 2004. Disability-related simulations: if, when and how to use them in professional development. Review of Disability Studies, 1(2), 8 – 18.

Burton, M. and Walther, J. 2001. The value of web log data in use-based design and testing. Journal of Computer-Mediated Communication, 6(3). Available at: jcmc.indiana.edu/vol6/issue3/burton.html [accessed: 15/06/2008]

Cockton, G., Woolrych, A., Hall, L. and Hindmarch, M. 2003. Changing analysts' tunes: the surprising impact of a new instrument for usability inspection method assessment. In Proceedings of HCI-2003: People and Computers XVII, Eds. P. Palanque, P. Johnson and E. O'Neill, 145-161.

Cook, A. and Polgar, J. M. 2008. Assistive technologies: principles and practice. St Louis, MI: Mosby/Elsevier.

Cooper, M. and Rejmer, P. 2001. Case study: localization of an accessibility evaluation. In the CHI '01 extended abstracts on Human factors in computing systems, 141-142. New York: ACM Press.

Desurvire, H. W., Kondziela, J. M., and Atwood, M. E. 1992. What is gained and lost when using evaluation methods other than empirical testing. In the Proceedings of the HCI '92 Conference on People and Computers VII, 89-102.

Dillon, A. 2001. Beyond usability: process, outcome and affect in human computer interactions. Canadian Journal of Library and Information Science 26(4): 57–69.

Dumas, J. S. and Redish, J. C. 1999. Practical Guide to Usability Testing. Intellect Books.

EDeAN. 2007. European Design for All eAccessibility Network. Home page. Available at: www.edean.org [accessed 15/06/2008]

Gould, J. D. and Lewis, C. 1985. Designing for usability: key principles and what designers think. Communications of the ACM 28(3): 300–311.

Gray, W. D. and Salzman, M. C. 1998. Damaged merchandise? A review of experiments that compare usability evaluation methods. Human-Computer Interaction 13(3): 203-261.

Groves, K. 2007. The limitations of server log files for usability analysis. Boxes and Arrows. Available at: www.boxesandarrows.com/view/the-limitations-of [accessed: 15/06/2008]

Harris, J. 2005. An Office User Interface Blog. http://blogs.msdn.com/jensenh/archive/2005/10/31/487247.aspx [accessed: 15/06/2008]

Harrison, C. and Petrie, H. 2006. Severity of usability and accessibility problems in eCommerce and eGovernment websites. In Computers and People XX: Proceedings of British Computer Society Human Computer Interaction Conference (BCS-HCI 06), 255-262. London: British Computer Society.

Hassenzahl, M. 2003. The thing and I: understanding the relationship between user and product. In Funology: from usability to enjoyment, M. Blythe, C. Overbeeke, A.F. Monk and P.C. Wright (Eds), pp. 31 – 42. Dordrecht: Kluwer.

Hassenzahl, M. 2006. Hedonic, emotional and experiental perspective on product quality. In Encyclopedia of Human Computer Interaction, ed. C. Ghaoui, 266--272. Idea Group.

Hassenzahl, M., Burmester, M., and Koller, F. 2003. AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. [AttrakDiff: A questionnaire for the measurement of perceived hedonic and pragmatic quality]. In J. Ziegler and G. Szwillus (Eds.), Mensch and Computer 2003: Interaktion in Bewegung, 187-196. Stuttgart, Leipzig: B.G. Teubner

Hassenzahl, M., Law, E. L.-C. and Hvannberg, E. T. 2006. User experience: towards a unified view. In User experience: towards a unified view. Proceedings of the 2nd COST294-MAUSE International Open Workshop, eds. E. L.-C. Law, E. T. Hvannberg and M. Hassenzahl. Available at: http://www.cost294.org/ [accessed 15/06/2008]

Hassenzahl, M. and Tractinksy, N. 2006. User experience: a research agenda. Behaviour and Information Technology 25(2): 91–97.

Hornbæk, K. 2006. Current practice in measuring usability: challenges to usability studies and research, International Journal of Human-Computer Studies, 64(2), 79-102.

Hwang, W. and Salvendy, G. 2007. What Makes Evaluators to Find More Usability Problems?: A Meta-analysis for Individual Detection Rates. HCI, 1, 499-507

International Standards Organization. 1998. ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs). Part 11: Guidance on usability. Geneva: International Standards Organization.

International Standards Organization. 2008a. ISO 9241-20: Ergonomics of human-system interaction – Part 20: Accessibility guidelines for information/communication technology (ICT) equipment and services. Geneva: International Standards Organization.

International Standards Organization. 2008b. ISO 9241-171: Ergonomics of human-system interaction. Part 171: Guidance on software accessibility. Geneva: International Standards Organization.

International Standards Organization. 2008c. ISO DIS 9241-210: Ergonomics of human-system interaction. Part 210: Human-centred design process for interactive systems (formerly known as 13407). Geneva: International Standards Organization.

International Standards Organization. 2008d. ISO/IEC 10779: Office equipment accessibility guidelines for elderly persons and persons with disabilities. Geneva: International Standards Organization.

International Standards Organization. 2006. ISO 20282-2: Ease of operation of everyday products - Part 2: Test method for walk-up-and-use products. Geneva: International Standards Organization.

Ivory, M. Y. and Hearst, M. A. 2001. State of the art in automating usability evaluation of user interfaces. ACM Computing Surveys 33(4): 470-516.

Jeffries, R. and Desurvire, H. 1992. Usability testing vs. heuristic evaluation: Was there a contest? SIGCHI Bulletin 24(4): 39-41.

Kiger, G. 1992. Disability Simulations: Logical, Methodological and Ethical Issues, Disability and Society, 7(1), 71 – 78.

Law, E., Roto, V., Vermeeren, A., Kort, J. and Hassenzahl, M. 2008. Towards a shared definition of user experience. In the CHI '08 extended abstracts on Human factors in computing systems, 2395-2398. New York: ACM Press.

Lewis, J.R. 2005. Usability Testing. In Salvendy, G. (Ed.), Handbook of Human Factors and Ergonomics, 3rd Edition.

Lewis, J. R. 2006. Sample sizes for usability tests: mostly math, not magic. Interactions, 13(8), 29-33.

Mayhew, D.J. 2005. Keystroke Level Modeling as a Cost Justification Tool. In Cost-Justifying Usability: An Update for the Internet Age, Eds. R. G. Bias and D. J. Mayhew, pp. Morgan Kaufmann.

Medlock, M. C., Wixon D., Terrano, M., Romero R. and Fulton B. 2002. Using the RITE method to improve products: a definition and a case study. In the Proceedings of Usability Professionals Association 2002. Available at: www.microsoft.com/downloads/results.aspx?pocId=&freetext=rite%20method [accessed 15/06/2008]

Molich, R. and Nielsen, J. 1990. Improving a human-computer dialogue. Communications of the ACM 33(3): 338-348.

National Institute of Standards and Technology (NIST). 2007. Common industry specification for usability – requirements (IUSR: CISU-R v0.90). NIST. Available at: http://zing.ncsl.nist.gov/iusr/ [Accessed 15/06/2008]

Nielsen, J. 1994. Heuristic evaluation. In Usability Inspection Methods, eds. J. Nielsen and R. L. Mack. New York, NY: John Wiley & Sons.

Nielsen, J. and Molich, R. 1990. Heuristic evaluation of user interfaces. In the Proceedings of CHI'90: ACM Annual Conference on Human Factors in Computing Systems, 249-256. New York: ACM Press.

Nielsen, J. and Sano, D. 1994. SunWeb: User interface design for Sun Microsystem's internal web. In the Proceedings of the 2nd World Wide Web Conference '94: Mosaic and the web, 547- 57. Available at: http://www.useit.com/papers/sunweb/ [accessed 25/05/2008]

Norman, D. A. 1998. The invisible computer. Cambridge, Mass: MIT Press.

Norman, D. A. 2004. Emotional design: Why we love (or hate) everyday things. New York, NY: Basic Books.

O’Connor, J. 2007. Joomla! accessibility. Birmingham, UK: Packt Publishing.

Petrie, H., Hamilton, F., King, N. and Pavan, P. 2006. Remote usability evaluations with disabled people. In the Proceedings of CHI 06: ACM Annual Conference on Human Factors in Computing Systems, 1133-1141. New York: ACM Press.

Petrie, H., Harrison, C. and Dev, S. 2005. Describing images on the Web: a survey of current practice and prospects for the future. In the Proceedings of 3rd International Conference on Universal Access in Human-Computer Interaction, part of HCI International 2005, ed. C. Stephanidis [CD-ROM]. Mahwah, NJ: Lawrence Erlbaum.

Petrie, H. and Kheir, O. 2007. The relationship between accessibility and usability of websites. In the Proceedings of CHI ’07: ACM Annual Conference on Human Factors in Computing Systems, 397-406. New York: ACM Press.

Pew, R. W. and Mayor, A. (eds.). 2007. Human-system integration in the system development process: a new look. National Academies Press. Section on Models and Simulations available at: books.nap.edu/openbook.php?record_id=11893&page=240

Regan, B. 2004. Best practices for accessible Flash design. San Francisco, CA: Macromedia. Available at: http://www.adobe.com/resources/accessibility/best_practices/bp_fp.html [accessed: 26 May 2008]

Rubin, J and Chisnell, D. 2008. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley.

Schütte, S., Eklund, J., Axelsson, J. R. C. and Nagamachi, M. 2004. Concepts, methods and tools in Kansei Engineering. Theoretical Issues in Ergonomics Science 5(3): 214-232.

Shackel, B. 1990. Human factors and usability. In Human-computer interaction: Selected readings, eds. J. Preece and L. Keller. Hemel Hempstead: Prentice Hall.

Shackel, B. 1991. Usability – context, framework, definition, design and evaluation. In Human factors for informatics usability, Eds. B. Shackel and S. Richardson, 21-37. Cambridge: Cambridge University Press.

Sharp, H., Rogers, Y. and Preece, J. 2007. Interaction design: beyond human-computer interaction. London: John Wiley.

Shneiderman, B. and Plaisant, C. 2005. Designing the user interface: strategies for effective human-computer interaction. Boston: Pearson Education.

Slavkovic, A. and Cross, K. 1999. Novice heuristic evaluations of a complex interface. In the CHI '99 extended abstracts on Human factors in computing systems, 304-305. New York: ACM Press.

Snyder. C. 2003. Paper prototyping The fast and easy way to define and refine user interfaces. San Francisco, CA: Morgan Kaufmann.

Sommerville, I. 1995. Software engineering (5th Edition). Harlow, UK: Addison-Wesley.

Spencer, R. 2000. The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company. In the Proceedings of CHI 2000: ACM Annual Conference on Human Factors in Computing Systems, 353-359. New York: ACM Press.

St. Amant, R., Horton, T. E. and Ritter, F. E. 2007. Model-based evaluation of expert cell phone menu interaction. ACM Transactions on Computer-Human Interaction (TOCHI) 14(1): Article 1.

Stone, D., Jarrett, C., Woodroffe, M. and Minocha, S. 2005. User interface design and evaluation. San Francisco, CA: Morgan Kaufmann.

Story, M., Mueller, J. and Mace, R. 1998. The universal design file: designing for people of all ages and abilities. Raleigh, NC: North Carolina State University. Available at: http://www.design.ncsu.edu/cud/pubs_p/pud.htm [checked 25 May 2008]

U.S. Department of Health and Human Sciences. 2006. Research-Based Web Design & Usability Guidelines. Available at: www.usability.gov/guidelines/

Web Accessibility Initiative (WAI). 1999. Web Content Accessibility Guidelines 1.0. Available at: http://www.w3.org/TR/WCAG10/ [checked: 18 May 2008]

Web Accessibility Initiative (WAI). 2006. Introduction to web accessibility. Available at: http://www.w3.org/WAI/intro/accessibility.php [checked: 18 May 2008]

Web Accessibility Initiative (WAI). 2008. Web Content Accessibility Guidelines 2.0. Available at: http://www.w3.org/TR/WCAG20/ [checked: 18 May 2008]

Wharton, C., Rieman, J., Lewis, C. and Polson, P. 1994. The Cognitive Walkthrough: A practitioner's guide. In Usability inspections methods, eds. J. Nielsen & R. L. Mack, 105-140. New York: Wiley.

Whiteside, J., Bennett, J. and Holtzblatt, K. 1988. Usability engineering: our experience and evolution. In Handbook of Human-Computer Interaction, ed. M. Helander, 791 – 817. North Holland.