

3. Usability Methodologies and Techniques

There are a variety of established methodologies and techniques that can be used to identify usability and accessibility issues with DL's. This section details a number of these and explains their goal, methodology, when they can be most effectively implemented and an example of how JISC services could employ them.

3.1 Usability Evaluation

Usability evaluations (UE) consist of methodologies for measuring the usability aspects of a system's user interface (UI) and identifying specific problems. They are an important part of the overall user interface design process, which consists of iterative cycles of designing, prototyping, and evaluating. (Dix et al., 1998, Nielsen, 1993). According to Preece (1994), evaluation is concerned with gathering data about the usability of a design or product by a specified group of users for a particular activity within a specified environment or work context.

Ivory and Hearst (2001) suggested that the main activities involved in an evaluation include:

- Capture: collecting usability data, such as task completion time, errors, guideline violations and subjective ratings;
- Analysis: interpreting usability data to identify usability problems in the interface;
- Critique: suggest solutions or improvements to mitigate problems.

There are various evaluation techniques commonly used by usability professionals at the moment. These techniques are applied in different stages of the design of products and services. The findings and results of the usability evaluation can vary widely when different evaluators study the same user interface, even if they use the same evaluation technique (Jeffries et al., 1991; Molich et al., 1998, 1999; Nielsen, 1993). The usability evaluation usually covers a subset of the possible actions users might take; as a result, it is often recommended to use several different evaluation techniques (Dix et al., 1998; Nielsen, 1993) in parallel.

When conducting evaluations, pre-defined test tasks should be selected prior to the evaluations for techniques such as cognitive walkthrough and user testing. Selected test tasks should represent characteristic user actions and goals.

Selecting Test Tasks

The test tasks selected should be based on the intended context of use and key scenarios of use. Tasks should aim to describe specific scenarios of how and why users would access the site, and what they want to achieve. Furthermore, according to Nielsen (1995) the following guidelines should be taken into account when designing the test tasks:

- Test tasks chosen should be as representative as possible of the users to which the system will eventually be put in the field.
- Test tasks should provide reasonable coverage of the most important parts of the interface.
- Test tasks should be small enough to be completed within the time limits of the use test, but should not be so small that they become trivial.
- Test tasks should specify precisely what result the user is being asked to produce.
- Test tasks should not be frivolous, humorous, or offensive.
- Test tasks should be academic-oriented and as realistic as possible
- Test tasks should relate to an overall scenario in order to increase both the users' understanding of the tasks and their sense of being realistic usage of the application.
- The very first task should be designed to be simple in order to guarantee the user an early success experience to boost moral.
- The last task should be designed to make users feel that they have accomplished something.

Selecting the Class of Users

It was important that the group of users that participate in the test matched the target user group of the evaluated services/applications. The assumptions made on the class of users should consist of the attributes collected from the requirements stage in terms of their demographic distribution such as gender, age, occupation, experience etc, depending on the type of evaluated services or application.

3.2 Stakeholder Analysis

Stakeholders are anyone who is affected by the success or failure of a system. This includes end users and others who have a stake such as managers, maintainers and marketing people etc. In defining stakeholders, a stakeholder analysis provides a baseline for effective requirements engineering and subsequent system design, as well as establishing requirements for all key stakeholders.

USTM

The USTM (User Skills and Task Match) identifies the groups of stakeholders who are responsible for system design and development, those involved with a financial interest; people whom are responsible for system introduction and maintenance as well as those who have an interest in the use of the system. (MacCaulay et al., 1990)

CUSTOM

According to Kirby (1991), the CUSTOM model distinguishes four categories of stakeholders:

- Primary: those who use the system
- Secondary: those who don't use the system directly, but receive output from it or provide input to it
- Tertiary: those not included in categories 1 and 2 but are affected by the success and failure of the system
- Facilitating those who are involved with the design, development or maintenance of the system.

In determining the user characteristics, the following criteria should be focused on:

- physical/mental abilities
- background, preference, motivation
- anticipated system user – discretionary or mandatory use
- patterns of use – continuous, intermittent
- knowledge of domain, operating system/GUI, computer familiarity
- probable learning curve – frequency of use, general abilities
- expertise – novices, skilled users, expert users

Implementation

Stakeholder analysis is a requirement gathering technique that is used in the early stage in the evaluation process for it is important to identify the anticipated user populations and their salient characteristics. Hence questions like who are the stakeholders of the service, what do they wish the service to provide, need to be addressed early in the design cycle.

JISC Implementation

In the evaluation process, JISC should conduct a stakeholder analysis at the early stage to gather the requirements of all stakeholders. This could be done by adopting the USTM and CUSTOM model to identify and determine who the stakeholders are and their characteristics at the early stage.

Additional Resources

Mark Kirby homepage on stakeholder requirements for computer systems
<http://scom.hud.ac.uk/scomdb2/ppp/mark/markkirby.htm>

D. Bell, A. Gupta, H. Rozendaal & E. Spencer (1995), Building Bridges between Human Factors and Software Engineering
<http://www.ash-consulting.com/HCI-95-paper.doc>

3.3 Personas

Personas are a usability technique designed to direct the focus of the development process towards the goals of the people who actually use the product. The process has traditionally been used by marketing companies, whereby a set of personas (detailed profiles of typical users) that best represent the intended audience based on statistical averages and demographics have been created to maximise the efforts of a marketing campaign (Glaze, 2002).

In the arena of software design the process is adapted and based around the needs of 'real' users, not on demographics and statistical averages. Alan Cooper (1999) notes that the simplest solution to rectify usability problems would appear to be to ask the user what they believe needs to be improved, but this does not work. Just because a user can identify a problem, it does not mean they will know how best to rectify it (Cooper, 1999). Therefore hypothetical archetypes of each user group are created, based upon facts discovered from the investigation process.

Designers often try to design a product that will suit every user's intended goal - the 'elastic user'. However, the broader the target audience that a piece of software is designed for, the less likely it will meet any individual's sole need.

Common issues with an interface that are found once the design team implements personas are:

- The design teams choose advanced technology over accessibility.
- Assumptions were made that users would be more impressed by a robust interface that they couldn't actually use, than by a less elegant application that serves their needs.
- Design teams perceive themselves to be the primary persona.

(Hourihan, 2002)

Designing for essentially one primary persona rectifies this problem, for the design teams personal preferences are removed and an increased awareness of the intended audiences varying skill levels and goals are recognised. Thus, focusing on one archetype creates a design for the broader group. In the case of JISC, not every user will have the same knowledge or requirements, so to focus the design around highly detailed personas will enable design teams to assess if they are meeting the actual needs of the users.

Methodology

Personas are almost always based upon the findings you retrieve from the requirements gathering stage via questionnaires and interviews. They are highly detailed and not just a job description – thus each is given an almost real world identity like first and last name, age, background story, goals, job title and even a photograph. The more specific the persona is the more effective design tools they become for their elasticity is removed. For example;

We don't just say that Emilee [derived persona] uses business software. We say that Emilee uses WordPerfect Version 5.1 to write letters to Grandma...We don't just let Emilee go to work. We give her a job as a New-Accounts Clerk in a beige cubicle at Global Airways in Memphis, Tennessee (Cooper, 1999).

These specific characteristics enable the persona to become a real archetypal user in the eyes of the design team.

The persona set should also remain small so they are manageable with each individual persona being allocated a designated status:

- **Primary** where the persona's need is sufficiently unique to require a distinct interface.
- **Secondary** where a primary interface will serve the needs of the persona with a minor modification/addition.
- **Supplemental** where the persona's needs are fully satisfied by a primary interface.
- **Served** where the persona is not an actual user of the product, but is indirectly affected by it and its use.
- **Negative** where the persona is created as an explicit, rhetorical example of whom *not* to design for. (Calde et al, 2002)

Implementation

Personas are designed to be used at the planning stage or at the early stage of your requirements capture. They can also be used to help guide other evaluation techniques.

JISC Implementation Example

JISC services are aimed at several broad user groups who all have different needs and abilities; information professionals; lecturers and teachers; students; publishers and researchers. Numerous personas would therefore be required if this technique were to be implemented.

The following example is a brief profile for a student that we created for this study.

Matthew Rogers. BSc Geography student at Sussex University.



Matthew is a 24-year-old student in the 2nd year of his degree course. He studied A-levels in history, geography and English literature at Harrow College in London before taking a gap year out to Australia. Matthew sits on the universities debating team and also works in the holidays with the National Trust. He loves gadgets and loves to explore the different tools that software applications make available. Matthew has a high level of computer competence and has taken additional courses in Java and Visual Basic. Matthew is familiar with mapping services but has never used Digimap before.

Figure 1: Example of a Persona

Matthew's requirements may mean that he is not regarded as a primary persona but is allocated the status of secondary persona for his requirements can be met by adding additional modifications to a primary interface. For a service such as this the technique may identify that more than one primary persona may be required. Therefore there may be more than one interface needed to '...create a truly comprehensive and integrated information system that enables each primary persona to achieve his or her goals' (Calde et al., 2002).

Additional Resources

Ennis provide a comprehensive list of the benefits that can be gained by including personas in the design cycle of a product:

http://www.eias.ie/aaa/usability_service_persona_devel.htm

Alan Cooper's web-site <http://www.cooper.com> and his book *The Inmates are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity* both provide leading information regarding the implementation of persona's in HCI Design.

3.4 Scenarios

Scenario-based methods is the description of people using technology and it is essential in discussing and analysing how the technology is (or could be) used to reshape their activities. A scenario describes a sequence of events when interacting with a system from the users' perspective and the scenario descriptions can be created before a system is built and its impacts felt.

'Scenarios' are similar to 'Use Cases', which describe interactions at a technical level, but scenarios can be easily understood by anyone regardless of the level of their technical knowledge. Scenarios are especially useful when you need to remove the focus from the technology in order to consider other design possibilities. Scenarios focus in terms of tasks rather than the technology used to support them. E.g. "User enters his pin" is incorrect because it mentions the technology used, whereas "User identifies himself" is okay because it keeps open other alternatives.

Methodology

Typically, three to four scenarios should be a good starting point to cover the standard users of a web-site, and more scenarios might be required when the site has a diverse group of audience with different needs. For example, JISC web-site has to cater the needs for disabilities when designing the web-sites. And for each scenario, validate it by asking the represented users to review each scenario. A standard scenario often links to a specific persona, therefore it consists of a profile of the user and an interaction episode or story. Figure 3 further shows the process of how scenarios should be applied into the entire development process.

Problem Scenario

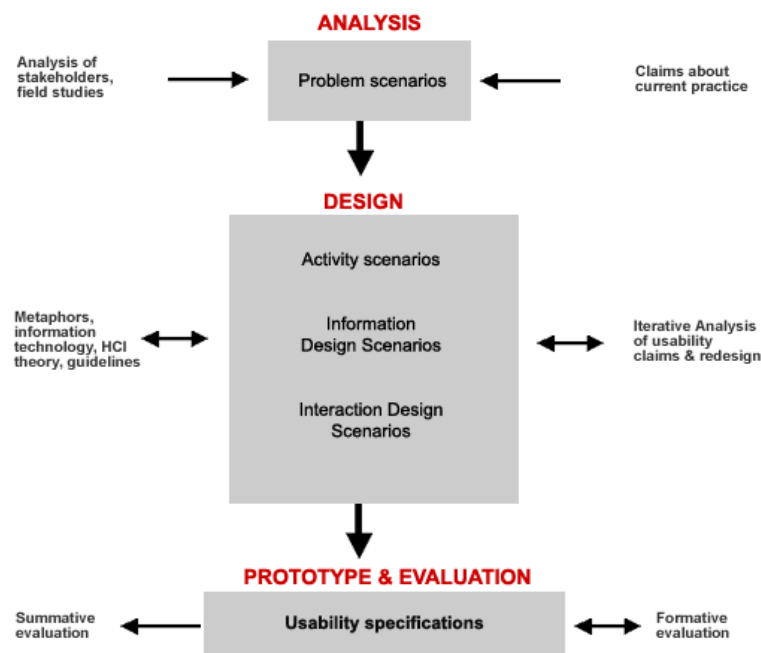
A problem scenario tells a story of the current practices. These stories are carefully developed to reveal aspects of the stakeholders and their activities that have implications for design while other members of the project team should be able to read the problem scenarios and appreciate the work-related issues that the field study has uncovered. It is called "problem scenarios" not because they emphasise problematic aspects of current practices, but because they describe activities in the problem domain. In scenario-based design, new activities are always grounded in current activities.

Activity Scenario

In the activity scenario, the design team first introduces concrete ideas about new functionality, new ways of thinking about users' needs and how to meet them is the focus of activity scenarios. As in the other steps of the process, a claims analysis is generated to help identify the tradeoffs as you move forward with prototypes.

Information and Interaction Design Scenarios

After completing the claims analysis, the goal of information design is to specify representations of a task's objects and actions that will help users perceive, interpret, and make sense of what is happening.



(Rosson & Carroll 2002)

Figure 2: Usability Engineering: Scenario-Based Development of Human Computer Interaction

The goal of interaction design is to specify the mechanisms for accessing and manipulating the task information and activities.

Implementation

Scenarios are a relatively inexpensive technique. They are most useful in collecting actual data about real users based on surveys, interviews, focus groups, or observations of work

environments. This is when the user's work environment would have a big impact on the use of the web-site. A scenario is to make sure that the site is practically usable and actually serves the needs of target users in real life. Scenarios are usually created after requirement gathering techniques have been performed. Scenario-based methods help to define user requirements during the design process and are used to ensure necessary features and needs are supported when the system is being designed. By doing so a scenario brings out additional functional requirements and ideas for the user interface that are driven by users' profiles and context. Also, a scenario provides a rationale for design decisions that can be useful in presenting designs to the development team or to decision makers.

Scenarios have limitations when the user population is extremely diverse and the time is restricted and this does not allow the generation of a good coverage of scenarios. Also, it would be less useful for simple marketing sites and whenever the context is likely to be a major factor, e.g. search engines.

JISC Implementation Example

JISC's services would require numerous scenarios to cover a wide variety of needs. By adapting scenario in the design, this could help to describe how specific individuals in specific circumstances would use the web-site. This is to make sure that necessary small details are being considered for actually accomplishing real tasks with the web-site. It would also ensure that the design has been considered in context.

Figure 4 identifies how persona's and scenarios complement each other, by displaying typical scenarios we created for this study (see section 3.3)

- Matthew's goals are to:**

 - Manipulate maps of Devon to display campsites in the region.
 - Be able to choose from a variety of design tools.
 - Incorporate the maps with other software like Arc-view.

Figure 3: Example of scenario

Additional Information

Clarke, L. (1991). The use of scenarios by user interface designers. In D Diaper & N. Hammnd (Eds.) *People and Computers VI*. Cambridge: Cambridge University Press, pages 103-115.

Design for Learning, Stanford University (2002). Provides an overview of the scenario-based framework with examples of how to apply the theory into the design.

<http://ldt.stanford.edu/~gimiller/Scenario-Based/scenarioIndex2.htm>

3.5 Participatory Design

Participatory design (PD), also called the 'Scandinavian Challenge', refers to a design approach that focuses on the intended user of the service or product, and advocates the active involvement of users throughout the design process. According to Blomberg and Henderson (1990), the PD approach advocates the principles and the goal to improve the quality of life, achieving collaborative orientation and iterative processes. In participatory design, a team of people who represent the major stakeholders in a web design team work together to create designs that reflect the way users will actually use the product in their own work. Users play a central role in the participatory design sessions, they will talk about their work environments and the tasks they're trying to accomplish, including what works for them and what doesn't when they use their current tools. This proactive user input can both result in better designs and help shorten product development and testing cycles.

Methodology

Depending on the complexity of the tool or feature to be designed, a complete participatory design exercise normally last from one to five days. The exercise is performed by a team of individuals working with a facilitator round a table creating a paper prototype of an interface design.

In a complete participatory design, this methodology can be implemented as a four-step process (Ellis, Jankowski & Jasper, 1998):

- 1) Building bridges with the intended users
- 2) Map user needs and suggestions to the system
- 3) Develop a prototype
- 4) Integrate feedback and continue the iteration

Participatory design requires a group of people representing major stakeholders of the product/services including users, web-site designers and developers, a usability engineer, and others as needed (for example, people from documentation, training, or testing). The team should consist of at least four individuals but no more than eight representatives. This will make the best use of all the participants' time and ensure that useful information will be collected from the PD session.

Implementation

Participatory design method gives users an opportunity to interact with their suggestions for the web-site before moving forward to the actual design. In most cases, these interactions lead to practical improvements on user suggestions. Such improvements could result in producing a web-site that better fits the users' needs.

JISC Implementation Example

Participatory design involves a higher cost and longer development process compared to other approaches of usability engineering. Under the JISC environment, for example, to enable a true participatory design exercise would involve a lot of human resources. It would require a lot of collaborations among different parties from different organisations that make it difficult to carry out in reality. It is recommended that participatory design could be conducted for a smaller size development within JISC but not on large scale ones.

Additional Information

Computer Science Department, Stanford University, gives a comprehensive introduction of what participatory design is, from the history of PD, the current application of PD and its directions in the future with some useful references to understand PD in a deeper perspective.

<http://www-cse.stanford.edu/classes/cs201/projects-00-01/participatory-design/history.html>

M. Silva & A. Breuleux (1994) The Use of Participatory Design in the Implementation of Internet-based Collaborative Learning Activities in K-12 Classrooms.

<http://www.helsinki.fi/science/optek/1994/n3/silva.txt>

3.6 Interviews and Focus Groups

Interviews and focus groups elicit information about user's experiences and preferences. An Interview is a formal, structured technique where the moderator interacts with users, asking them about their personal experience and preferences regarding the targeted web services. Focus groups are an informal technique that can help to assess user needs and feelings both before interface design and long after implementation.

Focus groups are a very efficient method to evaluate a web system. It helps to get user feedback and gauge initial reactions to a design. Focus groups are also good at discovering how the actual performance of the web system differs from users' expectations. It should be treated as a way to find out how people react to ideas. It could be an effective technique for seeking to learn more about particular services at all stages of a development cycle.

The main difference between the two methods is that interviews involve speaking to one individual at a time while focus groups discuss issues with a group of people.

Methodology

To begin with an interview or hold a focus group, the first thing to do is to start with formulating questions about the particular web services based on the type of information the moderator wants to know about. Then it is good practice to start getting to know the interviewees, as this will encourage them to speak more freely once they get used to the atmosphere. The difference between interviews and focus groups compared to surveys and questionnaires (see more about surveys and questionnaires in section 3.7) is that the moderator is present to interact and facilitate discussion about the issues raised by the moderator's questions.

In a focus group, the moderator brings together six to nine users to discuss issues and concerns about the features of a user interface. The group typically lasts about two hours and is run by a moderator who maintains the group's focus. At the beginning, the moderator should ask everyone to introduce him or herself which enables them to get used to the discussion environment and be willing to participate in the discussion.

The following are some keys to maximise the data collected from focus groups:

- Schedule a moderator with previous experience co-ordinating focus groups
- Obtain a facility with several computers and a projection screen
- Recruit representative users, perhaps from a user group or email discussion group
- Ask participants to provide anonymous feedback via a computer station, web-site or email
- Include a survey at the beginning or at the end of the focus group

Implementation

Interviews and focus groups are useful for getting subjective reactions to the designs and for finding out how people handle a particular task. They are appropriate at any stage of the design. Conducting them at an earlier stage will enable design teams to obtain important information and be able to contribute user views to the final design. If conducted at a later stage, this would enable designers to receive more specific and concrete comments and information from the interviewees on the actual product. It should be assessed whether the focus group methodology is the most appropriate way to research the topic, and appropriate steps should be taken to guarantee that the study is conducted in an effective manner.

The main advantage of an individual interview is that other people in the group would not influence the individual as might happen in a focus group. On the other hand, in a focus group, if a person raises an idea, another person could then develop and expand that idea, and the moderator could further explore in greater detail on some issues. When holding a focus group, it is however important to watch out for 'group-think' where people tend to conform to one another's view and are reluctant to disagree with the consensus view.

The benefits of focus groups are that they are less expensive than conducting interviews with the same number of people. Also, the group interaction triggers memories which lead to uncovering additional issues that may not come up during interviews. Common problems that most users might experience can also be identified.

JISC Implementation Example

When designing a JISC web-site, conduct an interview or focus group to elicit user needs and functionality ideas, and use it for exploring preferences, opinions and subjective reactions. If there are prototypes available, ask people to review what they think about the prototypes.

Furthermore, if there are screen shots or a storyboard to review, an interview or focus group is a good way to ask users to walk through the site or to perform informal user testing. However, it is important to note that these techniques are not practical to conduct with highly inaccessible user populations such as high-paid busy professionals (e.g. doctors or lawyers). When conducting a focus group, it is difficult to conduct for users who are geographically isolated or dispersed and for highly specialised fields in which the target user population is small. As an alternative, it would be feasible to approach them at conferences that they all attend or consider conducting online interviews instead of face-to-face interviews.

Additional Information

Nielsen, Jakob, "The Use and Misuse of Focus Groups" <http://www.useit.com/papers/focusgroups.html>

Focus group interview to evaluate library services <http://www.berea.edu/library/bieval/focus-Group.html>

Greenbaum, Thomas L., [The Handbook for Focus Group Research](#), 1997, Sage Pubns; ISBN: 0761912533

Templeton, JF. [The Focus Group: A Strategic Guide to Organizing, Conducting and Analyzing the Focus Group Interview](#) (second edition), Probus Publishing

3.7 Questionnaires

Questionnaires are an indirect usability assessment method since they do not measure the user interface itself, but the end-users' subjective opinions of the interface. As a result, they are a common technique used for the elicitation, recording and collection of information about a design. Questionnaires are also an inexpensive way to reach a wide audience, anywhere between 50 and 1,000 users (Nielsen, 1993). However, due to the very nature of questionnaires resulting in no direct contact, the response rate is often very low. Thus anything above 50% is regarded as a success (Colton, 1999). Kirakowski (2002) further notes that if '...the questionnaire is reliable... then this feedback is a trustworthy sample of what you [will] get from your whole user population.' The lack of direct contact can also be a bonus, as the complete anonymity of a questionnaire enables the researcher to retrieve more in-depth personal information than he would necessarily gain through other techniques.

On the negative side, users' perceptions as expressed through questionnaires early on in the design process have a tendency to change at a later design stage. For example, a study regarding new features on an interface revealed that:

The correlation between users' predictions of whether they would like the new features, and their ratings of the feature having tried them was very low, indicating that one should not always interpret the [questionnaire] results literally (Nielsen, 1993).

Methodology

A questionnaire should be directed at a representative audience, with care being taken in its design to ensure a high response rate. Dix *et al.* (1998) note that '[t]he first thing the evaluator must establish is the purpose of the questionnaire: what information is sought? It is also useful to decide at this stage how the questionnaire responses are to be analysed'. Appropriate questions may arise out of other activities like brainstorming sessions with the design team, analysing the results from other evaluations or liaising with other stakeholders. Additionally, there are many pre-designed questionnaires by usability experts already aimed at web-based interfaces.

There are a number of types of questions that can be explored to help the evaluator obtain the information they require: factual, opinion or satisfactory questions all provide different sorts of data. Open or closed questions, Likert scales or semantic differential scales also enable the evaluator to analyse and present the results in different ways (Kirakowski, 2002). The design of a

questionnaire must follow some established guidelines however if it is to be successful in its aims:

- Wording and terminology used must be clear and simple.
- Must be neutral as possible and not leading or biased.
- Closed questions must have a complete set of response alternatives that do not overlap.
- Each question to focus on only one issue.
- Scales should be appropriate and consistent.
- Must be clear where to place answer corresponding to question.
- Questionnaire must be piloted before wide scale distribution.

(Nicholas, 2001)

The structure of a questionnaire of this nature usually starts by explaining its purpose, then general factual questions asking the user background information. This is useful to find out the range of experience etc within the sample group. It then asks opinion questions relating to specific features, further contributing to the study's final evaluation goals.

Piloting of a questionnaire is essential to ensure the quality of data retrieved is maximised. Typical questions that should be answered as a result of piloting your questionnaire are:

- Do the questions adhere to the scope and focus previously established?
- Is the range of possible answers sufficient?
- Are the questions clear, concise and easy to understand?
- Does the questionnaire provide the respondent with enough opportunity to express their personal opinions?
- Is the questionnaire structure clear?
- Does the questionnaire manage to maintain the respondent's attention till the end?
- How long does it take to complete?

The key therefore to retrieving good usability data, as stated by Shneiderman (1992), is to ensure that '...precise, as opposed to general, questions are used in surveys' thus providing the designer with '...useful guidance for taking action' to redesign the interface.

Implementation

By selecting appropriate questions, questionnaires can be applied at any stage in the design process. They are especially effective:

- At the beginning of a project to establish the users' requirements.
- Once the web-based service has gone live to gather feedback from end-users.

They are less effective when it is not established who the core users are, or when the motivation to return the questionnaire is low (Brinck et al., 2002).

JISC Implementation Example

Questionnaires can be a very powerful tool by which JISC could reveal how end-users regard general and more specific aspects of their services. Therefore identifying key usability issues by collating and comparing the results with those from the other evaluation techniques.

For many services general questions in a variety of formats could be asked, such as:

- *Closed questions:*

How often do you access this service?

Once a day

Once a week

Once a month

Less than above

- *Semantic differential scales:*

Relationship between headings and content is:

Confusing

Clear

1 2 3 4 5

- *Open ended questions:*

Please describe any ways the service could be improved to make it more user friendly?

To ascertain direct information relating to one type of service then more specific questions could be presented:

- *Closed questions:*

How do you regard the quantity of information provided in the 'core records'?

Far too much

Too much

Satisfactory

Too little

Far too little

- *Likert scale:*

The size of the default thumbnails is adequate:

Strongly agree

Strongly disagree

1 2 3 4 5 6 7

The data collected by questions such as these would provide the JISC with both quantitative and qualitative information, which the design team can use to identify areas for possible improvement with regards to usability and accessibility issues.

Additional Resources

Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Wokingham: Addison-Wesley Publishing Company. This book provides clear guidelines on how to construct effective questionnaires.

Kirakowski provides a list of common questions related to executing questionnaires in the field of usability engineering: <http://www.ucc.ie/hfrg/resources/qfaq1.html>

QUIS provides questionnaires aimed at assessing users' subjective satisfaction to an interface, each questionnaire is designed to be configured according to the needs of each interface analysis:

<http://www.lap.umd.edu/quis/about.html>

3.8 Guidelines

End users' priorities may at times conflict with those of designers, developers or owners. In such cases guidelines will advocate the users' best interests. Recommended design guidelines are also used to ensure usability design principles are adhered to. Guidelines are vital for web-based technologies, since users display different habits when using this medium over more traditional information sources.

- 79% of users scan the page instead of reading word-for-word.
- Reading from computer screens is 25% slower than from paper.
- Web content should have 50% of the word count of its paper equivalent.

At the moment, there are lots of design guidelines and standards in relation to usability. With a series of International ergonomic standards, the ISO 9241 (1994) Ergonomics requirements for office work with visual display terminals (VDTs) provides guidance in the form of general principles and techniques by describing the basis of the user performance approach rather than in the form of requirements to use specific methods. According to the ISO 9241, usability is the extent to which a computer system enables users, in a given context of use, to achieve specified goals effectively and efficiently while promoting feelings of satisfaction.

After applying guidelines Sun Microsystems noticed a significant improvement in all of the key usability metrics, leading to an improvement of 159% in overall usability of their site (Nielsen, 2002).

Usability guidelines build upon previously derived usability principles, but their adoption is not as stringent as specific conventions that must be followed. Design teams may only find 90% of a set of guidelines to be applicable to an interface. Hence the team has to decide which guidelines are most appropriate (Thornton, 2002).

There are many sources providing web usability guidelines that address content accessibility and design (see Additional Resources). These sources fall into five categories:

1. Design rules – a set of functional or operational specifications applying to a particular user interface.

2. Ergonomic algorithms – a comprehensive and systematic procedure that usually appears as a software component.
3. Style guides – set of guidelines that apply to a specific collection of user interfaces.
4. Compilation of guidelines – devised for a wide range of user interfaces.
5. Standards – to regulate

However, Ratner, Grose & Forsythe (Scapin, 1999) have noted in a study that there is little consistency between different guidelines. In a study of 21 established style guides, only 25% of the total recommendations appeared in more than one guide. In addition, Spool (2002) notes that ‘web usability guidelines are very sensitive to the nature of the tasks and subtle differences in the content’. JISC has a number of different services like multimedia, electronic libraries and geo-spatial information. All of these would require some specific guidelines to ensure the usability and accessibility of each service is maximised.

Methodology

Guidelines are derived from a number of procedures like the information gathered in the requirements stage, developing an understanding of best practices by examining your own site and those of comparable sites, and manipulating existing guideline frameworks. Web-sites such as Amazon and eBay have developed guidelines by creating their own mutations.

They'll isolate one of their many servers and change the design of a few pages, just for users of that server. They'll compare the results with users of the other servers running the 'existing site'. Because of their traffic volume, they can learn a lot in just a few hours (Spool, 2002).

Once a set of guidelines has been established then the interface should be evaluated to ensure none of them have been violated. However, guidelines should not be singularly considered as a remedy to ensure a site is usable. They are designed to complement other HCI design techniques.

Implementation

Suitable guidelines should be applied throughout a user-centred design process. Whereby *local* guidelines should be identified and applied to a specific phase in the development life cycle.

Pervasive guidelines should be implemented through a number of continuous stages and *global* guidelines are relevant to all stages in the design (Scapin, 1999).

JISC Implementation Example

Many standard guidelines can be applied to a range of JISC services:

1. Create a style of presentation that is consistent across pages.
2. Use navigation mechanisms in a consistent manner.
3. Provide information about the general layout of a site (e.g., a site map or table of contents).
4. If search functions are provided, enable different types of searches for different skill levels and preferences.

More specific guidelines can also be devised to ensure that usability and accessibility issues are addressed (e.g. for an image service):

1. Give attention to visual organisation of the display
2. Ensure use of colours embedded in graphical images is consistent

(Gabbard & Hix, 1997)

Guidelines such as these can help ensure that JISC services meet the requirements of their varied users.

Additional Resources

The W3C provides 14 guidelines on web content and accessibility. Each is accompanied with additional checkpoints that have been awarded a priority rating.

<http://www.w3.org/TR/WAI-WEBCONTENT-TECHS>

Jakob Nielsen provides both guidelines for homepage design and specifically for multimedia on the web. www.useit.com

Ken Moffitt provides a compilation of usability guidelines for web-sites:

<http://www.unt.edu/benchmarks/archives/2002/august02/access.htm>

Organisations also provide their own guidelines such as:

- The National Cancer Institute: <http://usability.gov/guidelines>
- IBM: http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/572

3.9 Contextual Inquiry

Contextual inquiry, according to Holtzblatt and Beyer (1993) is a technique whereby usability issues of concern are identified by users, or by users and designers collaboratively, while the users are working in their natural environment on their own work. It is a structured field interviewing method, based on a few core principles that differentiate this method from plain, journalistic interviewing. It is a discovery process rather than an evaluative process; more like learning than testing.

Contextual inquiry is based on three core principles:

1. understanding the *context* in which a product is used (the work being performed) is essential for elegant design,
2. that the user is a *partner* in the design process,
3. that the usability design processes, including assessment methods like contextual inquiry and usability testing, must have a *focus*.

Methodology

Contextual inquiry follows many of the same process steps as field observations or interviews, with different considerations in some portions of the process. In contextual inquiry, the usability specialist actually becomes involved in the user's tasks, experiencing them as if they are the users themselves.

For example, interviewing during a contextual inquiry study usually does not include pre-set, broadly worded questions. Instead, the interviewer and interviewee create a dialogue conversation, where the interviewer can not only determine the user's opinions and experiences, but also his or her motivations and context.

Implementation

The contextual inquiry technique is best used in the early stages of development, it can be used to produce user needs analyses and task analyses. This is because a lot of the information the moderator receives will be subjective, such as how people feel about their jobs, how work or information flows through the organisation, etc. It is used to understand the context in which a task is being performed. It is one of the best methods to use to understand the users' work

context. This is due to the fact that the environment in which people work really influences how users use a product or services.

This technique helps finding out about work practices in domains that evaluators know nothing about - for example a lawyer looking up court cases in a DL.

Studies using this technique could be time-consuming and expensive, but they are effective when the cost could be justified. It is recommended that a successful way of conducting a contextual inquiry is to establish a master/apprentice relationship with a domain expert. This is because the domain expert could teach the usability specialist how to do a specific task or job.

JISC Implementation Example

JISC's BUFVC – the British Universities Film and Video Council - provides a range of services to promote the production, study and use of film and related media for education and research. When developing information services for BUFVC, a contextual inquiry could be conducted by establishing a master/apprentice relationship with a domain expert. For example, the domain expert (e.g. a film studies expert) demonstrates the production process of a film to the usability specialists in the studio. And based on information gathered from observations and interviews, the usability specialists will then be able to have deeper insight in terms of the context of how the production process of a film is by experiencing the task as if they are the users themselves.

Additional information

Information and Design. This link provides an introduction to contextual enquiry.

<http://www.infodesign.com.au/usability/contextualenquiry.html>

D. Wixon, K. Holtzblatt, S. Knox (1990) Contextual Design: An Emergent View of System Design.

<http://www.cs.indiana.edu/~connelly/Usability/Local/contextual-inquiry-paper.pdf>

3.10 Task Analysis

A task analysis is, according to Shepherd (1998), a process of sorting out what people actually do when they perform tasks, i.e. what actions they carry out; how they respond to different cues in their working environment and how they plan their activities. It is often applied to the design and evaluation of training, jobs and work, equipment and systems as well as in interactive system design. A task is a set of activities in which a user engages to achieve a particular goal, in which a 'goal' could be distinguished as the desired state of a system while a 'task' is the sequence of actions performed to achieve the goal, in which case it is a structured set of activities.

Use Case Analysis

Use cases analyse the development of a system from the perspective of how a user would typically interact with the system. Use cases combine a simple way of capturing user scenarios in a text document and diagramming how different user groups interact while using the system. The interaction between different actors in the web-site could be captured using the use case diagrams, and these diagrams provide a standard means for viewing an entire transaction in a single view.

Hierarchical task analysis

Hierarchical task analysis (HTA) describes the task in terms of a hierarchy of operations and plans based on structured chart notation. The hierarchical task analysis prompts the analyst to establish the conditions when various sub-tasks should be carried out in order to meet a system's goal. This method produces a hierarchy of three levels of task analysis: Goals (external task): system state that the human wishes to achieve, Tasks: structured set of activities in some sequence to achieve goals, Operations or actions: different things that a person must do within a system; simple tasks having no control structure.

Methodology

Use cases describe the activities of the users or actors of a system. Use cases include the typical or primary scenario that the user will go through to accomplish a particular goal or task and they also include alternative scenarios that the user might go through in other circumstances.

When conducting a HTA, the following have to be taken into consideration:

- It is necessary to look at the big picture. Identify the user groups that will be using the web-site, and how do users interact with other users using the site as well.
- Consider the pages that a single user will navigate in order to accomplish a particular task.
- Address the procedures that a user will utilise within each individual web page.

According to Brinck, Gergle & Wood (2002), it is recommended to use a hybrid method of task analysis with the combination of both use cases and HTA. This would be able to combine both the high-level interactions of users and other actors with the depth and psychological grounding of hierarchical procedure decomposition. It should be done with the following stages:

1. start with use cases
2. decompose tasks hierarchically
3. determine appropriate technologies.

Implementation

Task analysis should be used in the early stage of the design to capture user's requirements.

Use cases document the interactions between different user groups and are used as a first pass at high-level design. The limitations of use cases are that it might not be able to tell whether a procedure (scenario) is inefficient or whether these procedures are within the possibilities of human performance and the training that might be required.

While one advantage of HTA is that it is easy to learn and to use, a limitation is that it applies only to procedural activities and not to heavily parallel activities. HTA is also poor at capturing contextual information and it requires much time, skill and effort in use. As an alternative, by using use case analysis to supplement HTA, this would generate more specific information that is essential and also any additional improvements that could be made to the overall workflow.

Task analysis assumes that there is a correct and complete symbolic description of user tasks, and it seeks to capture that description. It consists of tasks as hierarchical structures of operations, articulated through systematic decomposition, often to fairly detailed levels. For task analysis, representing work activity is the primary objective. However, according to Diaper (2002),

informing and guiding design to optimise individual and collective performance is a hoped for benefit, but is not always easy to achieve.

JISC Implementation Example

When applying the task analysis technique, tasks should only be decomposed to a stage that the information obtained would affect the decisions on interface design choices.

In the JISC information environment, task analysis could be adopted to capture the user requirements in the early design stage. Services could adopt the hybrid task analysis approach in the early design stages that involve the combination of both use cases analysis and HTA.

Additional Information

Human Factors Design: <http://www.cdc.gov/niosh/mining/hfg/taskanalysis.html>

Usable Web – Topic on Task Analysis: <http://usableweb.com/topics/000876-0-0.html>

Kirwan, B. & Ainsworth, L.K. (Eds.) (1992). *A Guide to Task Analysis*. London: Taylor and Francis.

3.11 Accessibility Testing

Currently, the most authoritative standards for accessibility in web design are the W3C WAI guidelines (<http://www.w3.org/TR/WAI-WEBCONTENT>). There are several automatic tools available for evaluating web-sites in terms of accessibility at the moment. Bobby is an automatic web evaluation tool that provides detailed suggestions for improving web-sites (<http://www.cast.org/bobby>) while LIFT is both a usability and an accessibility automatic evaluation tool (UsableNet, 2002) (http://www.usablenet.com/products_services/lfdsnng/lfdsnng.html). Both Bobby and LIFT are based on either the Web Content Accessibility Guidelines 1.0 (WCAG) established by W3C or the American Section 508 accessibility standard. Also, RetroAccess (<http://www.retroaccess.com>) addresses usability errors on your web-site, it evaluates and corrects a single web page based on the Section 508 standards to see how the evaluation and correction process works.

Methodology

Automatic evaluation automates some aspects of usability and accessibility evaluations such as capture, analysis or critique activities. The advantages of automatic usability evaluation is that by automating the evaluating tasks we can:

- Reduce the cost of usability evaluation as the automation will minimise the time spent on usability evaluation and consequently the cost.
- Increase consistency of the errors uncovered
- Predict time and error costs across an entire design
- Reduce the need for evaluation expertise among individual evaluators
- Increase the coverage of evaluated features
- Enable comparisons between alternative designs.
- Incorporate evaluation within the design phase of UI development.

Bobby Automatic evaluation Tool

Bobby is the automatic evaluator for web accessibility and browser compatibility. It is one of the most widely used automatic accessibility evaluation tools. It recommends effective web page authoring for special web browsers. Bobby looks at the underlying HTML code that controls the presentation of a web page and analyses it against the Web Content Accessibility Guidelines 1.0 (WCAG) which is the W3C specification providing guidance on accessibility of web-sites for

people with disabilities. If Bobby detects a violation it will highlight it and then provide some guidelines on how to repair the HTML code. Bobby divides the accessibility errors into four sections to be tested:

- **Priority 1:** Accessibility problems that seriously affect a page's usability by people with disabilities. A Bobby Approved rating can only be granted to a site when none of the pages contain accessibility errors. Bobby Approved status is equivalent to Conformance Level A for the WCAG.
- **Priority 2:** Accessibility problems are second-tier problems that are considered important for access although not as fatal as Priority 1 access errors. Designers should try to fix the items in this section. If all items in this section in addition to the Priority 1 section, including relevant User Checks, are passed, the page meets Conformance Level AA for the WCAG. This is the preferred minimum conformance level for an accessible site.
- **Priority 3:** Accessibility problems are third-tier access problems. If all items in this section in addition to the Priority 1 and 2 sections are passed, including relevant User Checks, the page meets Conformance Level AAA for the WCAG.
- **Browser Compatibility:** Issues are HTML elements and element attributes that are used on the page which are not valid for particular browsers. These elements do not necessarily cause accessibility problems, but users might experience difficulties as the page may not be displayed as expected which may affect the usability and accessibility as a result.

(Zaphiris & Ellis, 2001)

As a general rating, once the Web-site reaches a Bobby Approved rating, the organisation is entitled (though not required) to use the Bobby Approved icon on the site. These icons identify the organisation as one committed to inclusion.

LIFT Automatic Evaluation Tool

LIFT combines both usability and accessibility evaluation in a single evaluation. It provides a report of a number of catastrophic errors (errors that disable users to complete tasks), major errors (errors that cause users to face major impediments), minor errors (errors that are really a nuisance for users) and cosmetic errors (low priority materials).

Automatic evaluation is the convenient and cost effective way to perform usability and accessibility analysis, however, due to the fact that some of the usability and accessibility elements cannot be fully determined and judged solely by automatic evaluation, it is not recommended to totally rely on these evaluation results to determine whether a web-site is truly usable and accessible or not. It is recommended that manual evaluations be carried out after automatic evaluations to supplement the evaluation results.

Analytic Evaluation - Manual/Non-Automatic Evaluation

Manual evaluations require an evaluator to conduct the evaluation and be familiar with the evaluation techniques - something that requires training, is time consuming, and requires many resources (in particular human resources) when there are a number of pages or a number of sites to evaluate within a limited time period.

Accessibility Guidelines

The same principles apply here as were introduced in the Guidelines section earlier in section 3.8.

Implementation

Accessibility guidelines should be applied when designing web-sites. The guidelines are focused on the concerns of people with visual impairments and those with motor difficulties that affect their physical ability such as to type or position a mouse pointer precisely. In order to ensure accessibility, it is necessary to follow the recommendations from guidelines as well as possible and then most importantly test the site with users with disabilities. Automatic tests are performed later in the design cycle.

JISC Implementation Example

In order to comply with the DDA, it is essential to take accessibility into design considerations and apply the guidelines into the design of the web-site where applicable. Additionally, it may also conduct an automatic accessibility evaluation.

3.12 User testing

User testing is a process of analysing how users really use an interface. It can often uncover very specific areas needing improvement, where focus groups and task analysis often find more general problems. According to Nielsen and Landauer (1993), testing with 15 users will be able to discover all the usability problems and the ultimate user experience is improved much by three tests with 5 users. According to Nielsen (1993) eight users is enough to test and the first eight users are expected to detect 85% of the site's usability problems.

Methodology

There are two types of user testing addressed below, formal and retrospective.

Formal User testing

User testing involves a testing session with users. The participants that test the system are potential users of the system. A set of test scripts should be prepared in advance aiming to collect both qualitative and quantitative data from the evaluation, ideally in a usability laboratory. Participants should be carefully selected from the potential/target user population.

A facilitator is responsible for conducting the testing and recording the results during the testing. Participants should be asked to perform a set of pre-selected tasks and their interactions with the interface as well as their facial expressions should be recorded during the test. They are asked to 'think aloud' whilst performing each task. The 'think aloud' technique is intended to generate a concurrent verbal protocol to capture what they were thinking while interacting with the system. It is a simple technique and was originated in psychological research methods (Ericsson & Simon 1984). Recently, it is increasingly popular for practical evaluation in usability research (Denning et al.,1990).

It is a good practice to ask participants to fill out a survey relating to the tested site in between intervals of each individual test task. This is because human memory has very limited capacity, especially regarding small details that are crucial for interface design. Informal interviews should be carried out with participants aiming to investigate the behaviours of users and evaluate how users interact with the interface on a task by task basis.

Retrospective User Testing

Retrospective user testing carries out a testing session with a group of participants performing the test at the same time under the same environment. A set of observation scripts is prepared in advance aiming to collect both qualitative and quantitative data from the evaluation. Retrospective verbal protocols are adopted requiring the subjects to report what they did after the task has been completed. It requires the subjects to remember what they did, but this could be supported by a video of task performance. Users are asked to fill out a survey relating to the site after each individual test task due to the limited capacity of human memory. After the user testing, a focus group is conducted with the participants to investigate the information seeking behaviour of the users, and evaluate how they interacted with the interface on a task by task basis.

In user-centred design, the focus is to match the needs and capabilities of the people that are going to use it. So if some target users are disabled, their needs should be considered in the design process. In an evaluation, the facilitator should do the review from the perspective of a disabled customer. When the web-site undergoes a usability test, some of the participants in the test should be from the disabled community.

Implementation

Usability evaluation could be conducted at any stage of the development process to ensure that the design is staying on track.

JISC Implementation Example

Evaluation is a critical component of every stage of the design. Under the JISC information environment, it is important to raise the quality of the design and keep costs under control by preventing the design from drifting off the track.

For example, when developing a JISC service user testing should be carried out throughout different stages of the development cycle. This would provide an informative approach to involve users in the design and to ensure that the JISC services conform to the usability and accessibility standards.

Additional Information

Cost of usability inspection methods: <http://www.pages.drexel.edu/~zwz22/Cost.html>

Usability evaluation techniques: <http://www.dcs.napier.ac.uk/marble/Usability/Evaluation.html>

G. Brajnik (2000) Automatic web usability evaluations: What needs to be done

<http://www.tri.sbc.com/hfweb/brajnik/hfweb-brajnik.html>

3.13 Cognitive Walkthrough

Cognitive walkthrough is an expert evaluation where usability experts, who may be sourced from outside or within an institution, possibly through a local psychology department, step through a scenario/task to question the design, focusing on the users' knowledge and goals.

Methodology

A cognitive walkthrough involves a detailed sequence of actions, i.e. the steps that an interface would require a user to perform in order to accomplish some tasks. Expert evaluators step through the sequence to check it for potential usability and accessibility problems. The main focus is to establish how easy a system is to learn.

Implementation

Before conducting a cognitive walkthrough a scenario/task should be created. A usability expert then, based on the scenario/task created to question the design, walks through by simulating the users' knowledge and goals.

The scenarios/tasks should represent the goals that users would be trying to achieve when using the web-site. The sequence of actions that the expert follows represent the pathway that real end-users follow to accomplish tasks, with potential problems be identified and documented. The evaluation should first start with high frequency tasks and then explore more specific or critical tasks, such as error recovery.

A set of templates based on the CE+ theory (Wharton et al, 1994), which is an information-processing model of human cognition, should be adopted to assist the walkthrough. This set of templates is designed in a way that the usability experts could then simulate the roles of each person's profile created in the persona and scenarios. The walkthrough should then aim to simulate the actual users making sure that the site actually serves the needs of specific people in real life.

JISC Implementation

A cognitive walkthrough could be conducted by JISC to evaluate the usability and accessibility of the current applications or throughout the development stage of a new application or in a re-design. This could start with creating scenarios and personas for that specific JISC web-based

application. The scenarios created would be adopted as the primary tasks that represent tasks that most users would be doing when using the JISC web-based application.

Usability experts could then examine the web-based application from the user's point of view based upon the personas created prior to the walkthrough and simulated the persona's behaviour by carrying out the predetermined scenarios. The experts would examine each of the correct actions needed to accomplish a task, and evaluate whether the four cognitive steps from the template were satisfactorily addressed. Based on the template designed for use in the cognitive walkthrough, the usability experts would repeat the four steps several times (CE+ theory by Wharton et al. (1994)), thus achieving a series of sub-goals that define the complete task for the scenarios that a user would attempt with the web-based application.

Additional Information

J. Rieman, M. Franzke, and D. Redmiles (1995) Usability Evaluation with the Cognitive Walkthrough

http://www.acm.org/sigchi/chi95/proceedings/tutors/jr_bdy.htm

Gregory Abowd (1995) Performing a Cognitive Walkthrough

<http://www.cc.gatech.edu/computing/classes/cs3302/documents/cog.walk.html>

3.14 Heuristic Evaluation

A heuristic evaluation is an expert evaluation method that uses a set of principles to assess if an interface is user friendly. One of its key advantages is that it is not dependent on end-user involvement, therefore it is a lot quicker and often cheaper to conduct - factors that are especially appealing when resources are limited (Kantner & Rosenbaum, 1997).

Nielsen and Molich re-developed the technique and provided the industry with an original set of nine heuristics, which were subsequently improved and extended (Dix et al., 1998). Since their introduction many new sets of heuristics have been devised. For example Nielsen himself has developed HOMERUN, a set he regards as more suitable for evaluations of commercial web-sites (Preece et al., 2002).

The effectiveness of an evaluation is largely dependent on the experience and number of experts involved, with three to five experts being regarded as sufficient to identify the majority of the key problems. Danino (2001) states that ‘...5 evaluators who are experts in software ergonomics and in the field in which the software is applied... will typically find 81%-90% of usability problems’.

However, it should be noted that heuristic evaluations are commonly regarded as an inferior form of evaluation since they are subject to evaluator bias and often miss or wrongly identify problems. Thus, Preece et al. (2002) believe that heuristic evaluations should not be considered a substitute for user testing.

Methodology

To carry out a heuristic evaluation the following steps must be followed:

1. Select your expert evaluators, preferably with extensive usability knowledge and domain awareness.
2. Define suitable tasks for the evaluators to attempt, or ask them to just walkthrough the site.
3. Each evaluator should independently attempt the tasks looking at each element of the interface, assessing it against the set of heuristics.

4. If a heuristic is contravened then the evaluator records a description of the problem, the location on the site, the heuristic that was violated and often a rating identifying how severe the problem is.
5. The problems identified by all the evaluators are then appropriately grouped, and possibly categorised (i.e. navigational problem, consistency problem).
6. The findings are presented in a written report.

As mentioned in step 4 a severity rating is also often applied to the usability problems uncovered, generally on a scale of 0 to 4 (4 being most critical). Ranking of usability problems by severity helps to determine the key problems that should be addressed, given that not all problems can be fixed due to constraints of time and costs. The severity rating of a problem is a combination of three key factors:

- **Frequency** – how often the problem occurs?
- **Impact** – how easy or difficult is it to overcome?
- **Persistence** – once aware of the problem can it be overcome, or will it repeatedly impede their performance?

(Nielsen, 1995)

The findings from an evaluation of this nature provide the design team with a large amount of qualitative data, which can be transformed into clear re-design proposals.

A note of caution however - as previously mentioned heuristic evaluations are perceived to contain judgements biased on the side of the evaluators to some degree. This is most prominently displayed by an inconsistency between experts with regard to severity ratings (Capra, 2001). Additionally, heuristic evaluations often identify usability issues that real users do not perceive as being a problem.

Implementation

Heuristic evaluations are suitable at almost any time during a user-centred design cycle. Thus the technique can be applied to prototypes or fully implemented interfaces to retrieve valuable information regarding issues of usability.

JISC Implementation Example

Heuristics are a powerful technique and there are currently a variety of sets focused upon web-based services. However, the diversity of JISC's services would imply that some modification of these sets would further ensure their effectiveness was maximised.

A study of heuristic evaluations has been carried out at Indiana University on their Digital Music Libraries interface. Two separate heuristic evaluations were conducted, one by users and the other by experts. Contrary to popular belief, they found that the users uncovered more problems with higher degrees of severity than the experts did. Thus, Minibayeva (2002) notes that there is potential for successful heuristic evaluations to be carried out by users:

[E]mploying user-based versions of certain expert-based methodologies could potentially ensure higher validity of usability evaluation, reduce the time and number of usability experts, and involve users more actively at earlier stages of systems design.

Traditional research has however shown that a high level of usability experience is imperative in achieving quality results. Although there is no question over domain knowledge, section 1 of this report has already identified that the level of usability experience varies greatly between JISC services. Therefore, heuristic evaluations do not present themselves as the most appropriate method to be widely implemented by internal JISC staff.

Additional Information

Jakob Nielsen's web-site *Useit.com* provides information on how to conduct a heuristic evaluation, and also presents different sets of heuristics:

http://www.useit.com/papers/heuristic/heuristic_evaluation.html

Danino provides a step by step guide to conducting heuristic evaluations:

<http://www.webmasterbase.com/article/520>

Instone provides an explanation and list of usability heuristics for web evaluations: <http://user-experience.org/uefiles/writings/heuristics.html>

A background introduction into Minibayeva's, research *A User-Based Approach to Cognitive Walkthrough and Heuristic Evaluation* is available at:

http://www.slis.indiana.edu/news/story.php?story_id=406

3.15 Summary of Costs and Benefits of the Methods by Olson & Moran (1996)

Methods	Type	Benefits	Costs
DEFINE THE PROBLEM			
Naturalistic observation (diaries, videotape, etc.)	Empirical	Tasks	2 days
Interviews (including focus groups, decision tree analysis, semantic nets)	Empirical	Tasks	1 day
Scenarios or use cases (including envisioning)	Analytic	Tasks	1 day
Task analysis (including operator function model)	Analytic	Tasks	2 days
GENERATE A DESIGN			
Building on previous designs (steal and improve, design guidelines)	Constructive	Tasks, perform, learn, accept	1 day
Represent conceptual model	Constructive	Learn	1 day
Represent interaction (GTN, dataflow diagram)	Constructive	Perform, learn	2 days
Represent visual display	Constructive	Perform, learn	2 days
Design space analysis (QOC, decomposition analysis)	Analytic	Tasks, perform, learn	3 days
REFLECT ON THE DESIGN			
Checklists	Analytic	Perform, learn	1 day
Walkthroughs	Analytic	Perform, learn	2 days
Mapping analysis (task action, metaphor, consistency)	Analytic	Perform, learn	2 days
Methods analysis (GOMS, KLM, CPM, CCT)	Analytic	Perform, learn	3 days
Display analyses	Analytic	Perform, learn	3 days
BUILD A PROTOTYPE			
Prototyping tools	Constructive	Testable system	1 month
Participatory prototyping	Empirical	Tasks, accept	1 week
TEST THE PROTOTYPE			
Open testing (storefront or hallway, alpha, damage testing)	Empirical	Perform, learn, accept	1 month
Usability testing	Empirical	Perform, learn, accept	1 year
IMPLEMENT THE DESIGN			
Toolkits (Motif, NeXTstep, Apple, etc.)	Constructive	Fully testable system	6 months
DEPLOY THE SYSTEM			
Internal testing	Empirical	Perform, learn, accept	1 month
Beta testing (logging, metering, surveys)	Empirical	Tasks, perform, learn, accept	1 month