



# Chapter 10

## Evaluation

### Contents

10.1 Introduction	239
10.2 Data analytics	242
10.3 Expert evaluation	246
10.4 Participant-based evaluation	250
10.5 Evaluation in practice	254
10.6 Evaluation: further issues	260
Summary and key points	265
Exercises	265
Further reading	266
Web links	267
Comments on challenges	267

### Aims

Evaluation is the fourth main process of UX design that we identified in Chapter 3. By evaluation we mean reviewing, trying out or testing a design idea, a piece of software, a product or a service to discover whether it meets some criteria. These criteria will often be summed up by the guidelines for good design introduced in Chapter 5, namely that the system is learnable, effective and accommodating. At other times the designer will want to focus on UX and measure users' enjoyment, engagement and aesthetic appreciation. Or the designer might be more interested in some other characteristic of the design such as whether a particular web page has been accessed, or whether a particular service moment is causing users to walk away from the interaction. UX designers are not concerned just with surface features such as the design of icons, or choice of colours, they are also interested in whether the system is fit for its purpose, enjoyable and engaging, and whether people can quickly understand and use the service.

Evaluation is central to human-centred design and is undertaken throughout the design process whenever a designer needs to check an idea, review a design concept or get reaction to a physical design.

After studying this chapter you should be able to:

- Understand data analytics
- Appreciate the uses of a range of generally applicable evaluation techniques designed for use with and without users
- Understand expert-based evaluation methods
- Understand participant-based evaluation methods
- Apply the techniques in appropriate contexts.

## 10.1 Introduction

The techniques in this chapter will allow you to evaluate many types of product, system or service. Evaluation of different types of system, or evaluation in different contexts, may offer particular challenges. For example, evaluating mobile devices, or services delivered by a mobile device, can be difficult to undertake and evaluation of interaction with wearable devices offer their own challenges.

Evaluation is closely tied to the other key activities of UX design: understanding, design and envisionment. In particular, many of the techniques discussed in Chapter 7 on understanding are applicable to evaluation. Evaluation is also critically dependent on the form of envisionment used to represent the system. You will only be able to evaluate features that are represented in a form appropriate for the type of evaluation. There are also issues concerning who is involved in the evaluation.

### Challenge 10.1

*Collect several advertisements for small, personal technologies such as that shown in Figure 10.1. What claims are the advertisers making about design features and benefits? What issues does this raise for their evaluation?*

**Design**  
Get the ultimate multimedia experience with Satio.



**Figure 10.1** Sony Ericsson T100 mobile phone  
(Source: [www.sonyericsson.com](http://www.sonyericsson.com))



In our human-centred approach to design, we evaluate designs right from the earliest idea – for example, very early ideas for a service can be discussed with other designers in a team meeting. Mock-ups can be quickly reviewed and later in the design process more realistic prototyping and testing of a partially finished system can be evaluated with users. Statistical evaluations of the near-complete product or service in its intended setting can be undertaken. Once the completed system is fully implemented, designers can evaluate alternative interface designs by gathering data about system performance.

There are three main types of evaluation. One involves a usability expert, or a UX designer, reviewing some form of envisioned version of a design. These are expert-based methods. Another involves recruiting people to use an envisioned version of a system. These are participant-based methods, also called ‘user testing’. A third method is to gather data on system performance once the system or service is deployed. These methods are known as data analytics. Expert-based methods will often pick up significant usability or UX issues quickly, but experts will sometimes miss detailed issues that real users find difficult. Participant methods must be used at some point in the development process to get real feedback from users. Both expert-based and participant-based methods can be conducted in a controlled setting such as a usability laboratory, or they

← Personas  
are described in  
Chapter 3

can be undertaken ‘in the wild’ where much more realistic interactions will happen. If real users are not easily available for an evaluation, designers can ask people to take the role of particular types of user described by personas. Data analytics can be gathered and analyzed once a system or service is implemented.

Evaluation occurs throughout the interaction design process. At different stages, different methods will be more or less effective. The form of envisionment of the service or system that the designer has, the questions to be asked and the people who are available are critical to what can be evaluated. Recall Oli Mival’s guide to framing and answering UX research questions (Box 7.2). He suggests documenting the outcomes of research (whether primarily focused on evaluation or on understanding) using the template in Box 10.1.

### BOX 10.1

#### UX Research Output Review

<b>Title:</b>	Name of the research activity
<b>Author:</b>	Who has written this research review and their role within the project
<b>Date:</b>	Date the research review was written
<b>Agency:</b>	Who has been commissioned to do the research (if an internal project, provide details of the team)

This document acts as an ‘executive summary’ of the research activity method and output detailing how the activity informs the design and insight objectives established in the relevant research brief document.

It should allow a reader unfamiliar with the project to determine:

- The research questions driving the research activity: what did we want to know?
- The participants: who did we ask?
- The research materials: what did the participants engage with?
- The data: what was the output and where is it now?
- An analysis summary: what did the data tell us?
- The insights derived: what was the value?
- The actionable output: what do we do next?
- The lessons learned: what would we do differently next time?

#### Part 1 The research objectives



Provide a brief summary of the insight objectives that motivated the research in the first place. Include what missing knowledge was holding back the design work and the high-level research questions that the research activity undertaken was seeking to answer.

#### Part 2 The research method



Provide a brief summary of the method deployed, detailing the research participants, what they were asked to do and the research materials they were provided with.

### Part 3 The research output



Provide detail on the data generated by the research activity – include storage location, format, quantity and an assessment of its quality. For example, was there anything that may have compromised the integrity or accuracy of the data?

### Part 4 The research analysis



Provide detail on the analysis undertaken, including the methods used to process the data generated. Provide links to supporting documents if available.

### Part 5 The actionable output



Provide details on the insights gained from the analysis of the data and how they will enable the design work to move forward.

### Part 6 Research review



Provide a brief reflection and critique on the success of the research activity and output gathered. For example, were the methods used effective in generating the type of data you needed or would you recommend alternatives when undertaking similar work in the future?

## Obtaining feedback to inform early design concepts

You may need to evaluate initial concepts, especially if the application is novel. Here, quick paper prototypes can help, or wireframes that can be produced rapidly. Evaluations of competitor products or previous versions of technology can also feed into the design process at this stage.

## Deciding between design options

During development, designers have to decide between options, for example between voice input and touchscreen interaction. Here it may be more appropriate to use well-focused quick experiments to look at the efficiency and effectiveness of different interface designs. Once a system is up and running, designers can try out new interface designs by making controlled changes to the live system and gathering data analytics about performance. This is often known as A/B testing.

## Checking for usability problems

Testing will identify potential problems once a stable version of the technology is available. This needs to respond when a participant activates a function, but it does not require the whole system to be fully operational (a horizontal prototype). Alternatively,

the system may be completely functional but only in some parts (a vertical prototype). This is sometimes termed formative evaluation because the results help to form – or shape – the design.

### Assessing the usability of a finished product

By contrast, summative evaluation is undertaken when a product or service is complete and designers want to demonstrate that the design meets some performance criteria. This may be to test against in-house guidelines or formal usability standards such as ISO 9241, or to provide evidence of usability required by a customer, for example the time to complete a particular set of operations. Government departments and other public bodies often require suppliers to conform to accessibility standards and health and safety legislation.

### Evaluating user experience (UX)

Usability is an important contributory factor to UX, but it is not the whole story. Mark Hassenzahl (Hassenzahl, 2010) talks about pragmatic and hedonic features of UX. There are a number of ways to measure the hedonic aspects of UX, such as the user experience questionnaire (UEQ) discussed in Section 10.6.

### Process

In the participatory design approach, stakeholders help designers set the goals for the evaluation work. Involving stakeholders has great benefits in terms of eventual uptake and use of the technology. (Of course, this applies only to technology that is tailor-made for defined communities, rather than off-the-shelf products.) A number of methods of involving people in the design process were discussed in Chapter 7 on understanding and Chapter 8 on envisionment.

### Qualitative or quantitative data

Evaluation methods pose the same issues of data analysis that were discussed in Section 7.10. Designers must decide whether to focus on gathering quantitative data such as the number of people who access a page of a website or the time users spend on a particular section of an app, or to gather qualitative data such as people's opinions of designs. One does not preclude the other, of course, and designers can design highly creative ways of gathering data that give them a good insight into particular aspects of their designs.

## 10.2 Data analytics

It has often been said that this is the era of 'big data'. Huge amounts of data are being generated across many different fields. The Internet of Things (IoT) refers to the interconnectedness of sensors and devices with one another and across the internet. Through these connections vast amounts of data are gathered and processed, potentially providing new insights into many aspects of our environment. Mobile devices are collecting increasing amounts of personal data, such as how many steps someone has taken in a day. Other sensors measure a person's heart rate, blood pressure or levels of excitement. As with IoT, this data has the potential to provide new insights into people's behaviours and performance.

## The quantified self (QS)

BOX  
10.2

The availability of various bio-sensors in mobile and wearable devices has led to a movement known as the quantified self, or personal analytics. Frequently associated with trying to get people to behave in a healthier way, QS poses interesting questions about data gathering and use. For example, a watch will vibrate if a wearer has not stood up or moved around for an hour. It monitors and displays heart rate data (Figure 10.2). Other personal data such as the number of steps someone has taken in a day or the number of stairs they have climbed is presented on personal 'dashboard' visualizations (Figure 10.3). How people react to these various representations of themselves is an interesting issue (e.g. see Choe *et al.*, 2014).



Figure 10.2 Measuring heart rate on a watch



Figure 10.3 Personal 'dashboard' visualization

In terms of evaluating UX and other aspects of interactive systems design, data analytics provides designers with data on system performance and the behaviours of individuals in interacting with systems and services. Data analytics also provides designers with interesting visualizations of the data and tools to help manipulate and analyze the data. The best known of the data analytics providers is Google Analytics. This is a free service that provides data about where website and app users have come from (including their country, and potentially more detailed information about location and the device they were using) and what they did when they interacted with the system (such as how long they used the system, which pages of a site they visited, the order in which they viewed pages and so on). Google Analytics can provide demographic information based on what users have told them, using a similar formula as that used to target Google Ads (advertisements). Facebook Analytics for Apps is a free service that can be installed and provides information about who used an app on Facebook. Since users on Facebook have often provided a lot of personal information, more user details can be found. The data from Google or Facebook Analytics is displayed using a ‘dashboard’, such as the one in Figure 10.4.



**Figure 10.4** Analytics for Apps dashboard

Using these data analytics services, designers can examine the activities of individuals and different groups such as Android phone users, people who access from a desktop machine using a particular browser, people who access the site from a particular location. Other important data for web analytics includes the number of visitors to a site over a period of time, the ‘bounce rate’ (the number of people who visited a site and then left immediately, without looking at any content), the number of pages viewed per session, time spent viewing pages and so on.



## Challenge 10.2

Look at the data in Figure 10.4. What inferences can you make?



Similar data can be obtained for mobile apps to help designers track which countries have downloaded their app, how often it has been used, for how long and so on. Indeed, there are increasingly large numbers of tools that can help with data analytics. Watson Analytics from IBM provides an easy-to-use interface to help designers and analysts express the focus of the data they want to look at. Watson Analytics undertakes sentiment analysis by searching social media channels such as Twitter and classifying tweets as positive or negative with respect to some topic. For example, a retailer may want to look at attitudes to Black Friday (the Friday before Christmas when many retailers discount products). Watson Analytics lets them input the topic, Black Friday, and helps the analyst choose synonyms and related topics that people on social media might be using. The system will then retrieve and classify social media traffic by sentiment, geographical area or demographics. The software allows the analyst to focus on specific individuals who may be particularly influential (for example, by highlighting those with the most followers on the social media channel). In this way Watson and similar pieces of software enable analysts to query the content of social media without having to write complex database queries. These products typically provide interesting graphical displays and other information visualization tools to help the analysts see trends and make comparisons across different media.

Other data analytic tools will provide a 'heat map' of a website showing which parts of a page visitors click on most frequently (Figure 10.5). Heat maps can also be produced

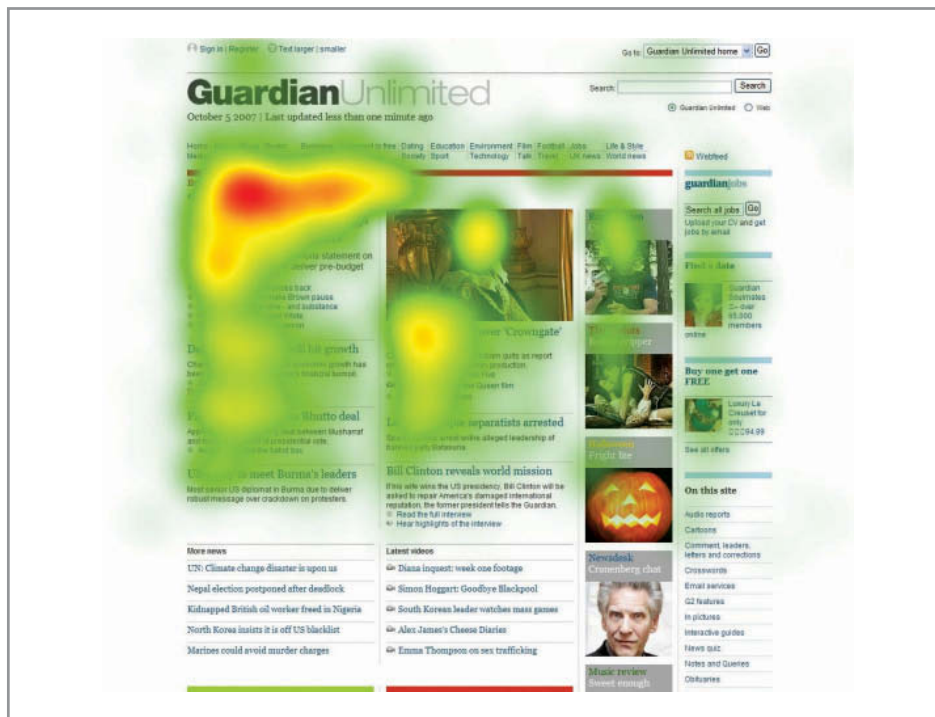


Figure 10.5 Heat map



from eye-tracking software which measures where people are looking on a display. Other tools will allow the analyst to follow people's browsing behaviour in real time, watching what they click on, how long they spend on particular sections and whether there are particular service moments at which people drop out of the customer journey.

The ability to understand user behaviour through data analytics, combined with the ability to rapidly deploy new versions of software, is changing the nature of interactive software development. For example, a games company that has deployed a game on Facebook can watch what players are doing in real time. If they notice some particular phenomenon – such as many people dropping out of the game before they move on to the next level – they can easily change the game, perhaps by introducing a surprise prize of extra money just before the end of the level, and thus encourage people to keep playing. In other circumstances a company may issue its software with two alternative interfaces, or with slightly different interfaces. The two interfaces are randomly assigned to users as they log onto a site. By looking at the analytics of the two interfaces, analysts can see which is performing better. This is known as A/B testing and is increasingly used to refine the UX of commercial websites.

### 10.3 Expert evaluation

A simple, relatively quick and effective method of evaluation is to get a UX or usability expert to look at a service or system and try using it. As we said in the introduction, this is no substitute for getting real people to use a design, but expert evaluation is effective, particularly early in the design process. Experts will pick up common problems based on their experience and will identify factors that might otherwise interfere with an evaluation by non-experts. Although the methods have been around for more than 20 years, expert-based methods are still widely used by industry (Rohrer *et al.*, 2016).

Sometimes called usability inspection methods, there is a variety of approaches to expert evaluation and an expert can simply be asked to look at a design and make suggestions. However, to help the experts structure their evaluation, it is useful to adopt a particular approach. This will help to focus the expert's critique on the most relevant aspects for the purpose. The general approach to expert evaluation is that the expert will 'walk through' representative tasks or scenarios of use. Additionally, they may adopt one of the personas. Thus expert evaluation is tied into scenario-based design (and central to it).

← Scenario-based design and the importance of evaluation are covered in Chapter 3

### Heuristic evaluation

Heuristic evaluation refers to a number of methods in which a person trained in HCI, UX or interaction design examines a proposed design to see how it measures up against a list of principles, guidelines or 'heuristics' for good design. This review may be a quick discussion over a colleague's shoulder, or may be a formal, carefully documented process.

There are many sets of heuristics to choose from, both general-purpose and those relating to particular application domains, for example heuristics for web design. Below is a brief list of the design principles – or heuristics – that we introduced earlier. (You should refer back to Chapter 5 for the details.)

- |               |                 |
|---------------|-----------------|
| 1 Visibility  | 7 Feedback      |
| 2 Consistency | 8 Recovery      |
| 3 Familiarity | 9 Constraints   |
| 4 Affordance  | 10 Flexibility  |
| 5 Navigation  | 11 Style        |
| 6 Control     | 12 Conviviality |

Ideally, several people with expertise in interactive systems design should review the interface. Each expert notes the problems and the relevant heuristic and suggests a solution where possible. It is also helpful if a severity rating, say on a scale of 1 to 3, is added, according to the likely impact of the problem, as recommended by Dumas and Fox (2012) in their comprehensive review of usability testing. However, they also note the disappointing level of correlation among experts in rating severity of problems.

Evaluators work independently and then combine results. They may need to work through any training materials and be briefed by the design team about the functionality. The scenarios used in the design process are valuable here.

## Discount usability engineering

The list of design principles above can be summarized by the three overarching usability principles of *learnability* (principles 1–4), *effectiveness* (principles 5–9) and *accommodation* (principles 10–12). If time is short, a quick review of the design against this triad can produce reasonably useful results. This is known as a discount heuristic evaluation.

This approach to evaluation was pioneered by Jakob Nielsen (1993), who uses a slightly different set of heuristics than ours, and enthusiastically followed by many time-pressured evaluation practitioners. It is now used for any ‘quick and dirty’ approach to evaluation where the aim is to get useful, informed feedback as soon as possible. Once again a number of usability experts ‘walk through’ concrete scenarios, preferably accompanied by personas, and inspect the design for difficulties.

### Challenge 10.3

Carry out a quick review of the controls for a domestic device, e.g. a stove, microwave or washing machine, for learnability, effectiveness and accommodation.



Unless there is no alternative, designers should not evaluate their own designs. It is extremely difficult to ignore knowledge of how the system works, the meaning of icons or menu names and so on, and the designers are likely to give themselves the ‘benefit of the doubt’ or to find obscure flaws which few users will ever happen upon.

Woolrych and Cockton (2000) conducted a large-scale trial of heuristic evaluation. Evaluators were trained to use the technique, then evaluated the interface to a drawing editor. The editor was then trialled by customers. Comparison of findings showed that many of the issues the experts identified were not experienced by people (false positives), while some severe difficulties were missed by the inspection against heuristics. There were a number of reasons for this. Many false positives stemmed from a tendency by the experts to assume that people had no intelligence or even common sense. As for ‘missing’ problems, these tended to result from a series of mistakes and misconceptions, often relating to a set of linked items, rather than isolated misunderstandings. Sometimes heuristics were misapplied, or apparently added as an afterthought. Woolrych and Cockton conclude that the heuristics add little advantage to an expert evaluation and the results of applying them may be counter-productive. They (and other authors) suggest that more theoretically informed techniques such as the cognitive walkthrough (see below) offer more robust support for problem identification. It is evident that heuristic evaluation is not a complete solution. At the very

least, the technique must be used together with careful consideration of people and their real-life skills. Participant evaluation is required to get a realistic picture of a system's success.

Heuristic evaluation therefore is valuable as *formative* evaluation, to help the designer improve the interaction at an early stage. It should not be used as a *summative* assessment, to make claims about the usability and other characteristics of a finished product. If that is what we need to do, then we must carry out properly designed and controlled experiments with a much greater number of participants. However, the more controlled the testing situation becomes, the less it is likely to resemble the real world, which leads us to the question of 'ecological validity'.



### Ecological validity

In real life, people multitask, use several applications in parallel or in quick succession, are interrupted, improvise, ask other people for help, use applications intermittently and adapt technologies for purposes the designers never imagined. We have unpredictable, complex but generally effective coping strategies for everyday life and the technologies supporting it. People switch channels and interleave activities. The small tasks which are the focus of most evaluations are usually part of lengthy sequences directed towards aims which change according to circumstances. All of this is extremely difficult to reproduce in testing and is often deliberately excluded from expert evaluations. So the results of most evaluation can only ever be indicative of issues in real-life usage.

Ecological validity is concerned with making an evaluation as life-like as possible. Designers can create circumstances that are as close to the real-life environment as possible when undertaking an evaluation. Designs that appear robust in controlled, 'laboratory' settings can perform much less well in real-life, stressed situations.

## Cognitive walkthrough

Cognitive walkthrough is a rigorous paper-based technique for checking through the detailed design and logic of steps in an interaction. It is derived from the human information processor view of cognition and closely related to task analysis (Chapter 11). In essence, the cognitive walkthrough entails a usability or UX analyst stepping through the cognitive tasks that must be carried out in interacting with technology. Originally developed by Lewis *et al.* (1990) for applications where people browse and explore information, it has been extended to interactive systems in general (Wharton *et al.*, 1994). Aside from its systematic approach, the great strength of the cognitive walkthrough is that it is based on well-established theory rather than a trial-and-error or a heuristically-based approach.

Inputs to the process are:

- An understanding of the people who are expected to use the system
- A set of concrete scenarios representing both (a) very common and (b) uncommon but critical sequences of activities
- A complete description of the interface to the system – this should comprise both a representation of how the interface is presented, e.g. screen designs, and the correct sequence of actions for achieving the scenario tasks, usually as a hierarchical task analysis (HTA).

→ Hierarchical task analysis is discussed in Chapter 11

Having gathered these materials together, the analyst asks the following four questions for each individual step in the interaction:

- Will the people using the system try to achieve the right effect?
- Will they notice that the correct action is available?
- Will they associate the correct action with the effect that they are trying to achieve?
- If the correct action is performed, will people see that progress is being made towards the goal of their activity? (Modified from Wharton *et al.*, 1994, p. 106.)

If any of the questions is answered in the negative, then a usability problem has been identified and is recorded, but redesign suggestions are not made at this point. If the walk-through is being used as originally devised, this process is carried out as a group exercise by analysts and designers together. The analysts step through usage scenarios and the design team are required to explain how the user would identify, carry out and monitor the correct sequence of actions. Software designers in organizations with structured quality procedures in place will find some similarities to program code walkthroughs.

Several cut-down versions of the technique have been devised. Among the best documented are:

- The ‘cognitive jogthrough’ (Rowley and Rhoades, 1992) – video records (rather than conventional minutes) are made of walkthrough meetings, annotated to indicate significant items of interest, design suggestions are permitted and low-level actions are aggregated wherever possible.
- The ‘streamlined cognitive walkthrough’ (Spencer, 2000) – designer defensiveness is defused by engendering a problem-solving ethos and the process is streamlined by not documenting problem-free steps and by combining the four original questions into two (*ibid.*, p. 355):
  - Will people know what to do at each step?
  - If people do the right thing, will they know that they did the right thing and are making progress towards their goal?

Both these approaches acknowledge that detail may be lost, but this is more than compensated for by enhanced coverage of the system as a whole and by designer buy-in to the process. Finally, the cognitive walkthrough is often practised (and taught) as a technique executed by the analyst alone, to be followed in some cases by a meeting with the design team. If a written report is required, the problematic interaction step and the difficulties predicted should be explained. Other checklist approaches have been suggested, such as the Activity Checklist (Kaptelinin *et al.*, 1999), but have not been widely taken up by other practitioners.

While expert-based evaluation is a reasonable first step, it will not find all problems, particularly those that result from a chain of ‘wrong’ actions or are linked to fundamental misconceptions. Woolrych and Cockton (2001) discuss this in detail. Experts even find problems that do not really exist – people overcome many minor difficulties using a mixture of common sense and experience. So it is really important to complete the picture with some real people trying out the interaction design. The findings will always be interesting, quite often surprising and occasionally disconcerting. From a political point of view, it is easier to convince designers of the need for changes if the evidence is not simply one ‘expert’ view, particularly if the expert is relatively junior. The aim is to trial the design with people who represent the intended target group in as near realistic conditions as possible.

Most of the expert-based evaluation methods focus on the usability of systems. For example, our own set of heuristics focuses on usability, as does Nielsen’s. Other writers develop heuristics specifically for websites, or for particular types of websites such as

e-commerce sites. However, there is no problem with designers devising their own heuristics that focus on particular aspects of the UX in which they are interested.

Recall from Chapter 7 the discussion on semantic differential and semantic understanding as ways of helping designers to understand users' views of a domain. Also in Chapter 8 we discussed how descriptive adjectives (which essentially are semantic descriptors) could be used as a method of envisioning the characteristics that some UX should aim to achieve. These descriptors can then be used as an evaluation tool, with UX experts working through an envisionment of a design and rating the experience against the specific characteristics that the design was intended to achieve. For example, we undertook an expert-based walkthrough of the TravoScotland app (Section 8.3) to see whether it achieved its objectives of being engaging, authoritative and modern.

## 10.4 Participant-based evaluation

Whereas expert, heuristic evaluations can be carried out by designers on their own, there can be no substitute for involving some real people in the evaluation. Participant evaluation aims to do exactly that. There are many ways to involve people that require various degrees of cooperation. The methods range from designers sitting with participants as they work through a system to leaving people alone with the technology and observing what they do through a two-way mirror.

### Cooperative evaluation

Andrew Monk and colleagues (Monk *et al.*, 1993) at the University of York (UK) developed cooperative evaluation as a means of maximizing the data gathered from a simple testing session. The technique is 'cooperative' because participants are not passive subjects but work as co-evaluators (Figure 10.6). It has proved to be a reliable but economical technique in diverse applications. Table 10.1 and the sample questions are edited from Appendix 1 in Monk *et al.* (1993).



**Figure 10.6** Evaluation

**Table 10.1** Guidelines for cooperative evaluation

Step	Notes
1 Using the scenarios prepared earlier, write a draft list of tasks.	Tasks must be realistic, doable with the software, and explore the system thoroughly.
2 Try out the tasks and estimate how long they will take a participant to complete.	Allow 50 per cent longer than the total task time for each test session.
3 Prepare a task sheet for the participants.	Be specific and explain the tasks so that anyone can understand.
4 Get ready for the test session.	Have the prototype ready in a suitable environment with a list of prompt questions, notebook and pens ready. A video or audio recorder would be useful here.
5 Tell the participants that it is the system that is under test, not them; explain and introduce the tasks.	Participants should work individually – you will not be able to monitor more than one participant at once. Start recording if equipment is available.
6 Participants start the tasks. Have them give you running commentary on what they are doing, why they are doing it and difficulties or uncertainties they encounter.	Take notes of where participants find problems or do something unexpected, and their comments. Do this even if you are recording the session. You may need to help if participants are stuck or have them move to the next task.
7 Encourage participants to keep talking.	Some useful prompt questions are provided below.
8 When the participants have finished, interview them briefly about the usability of the prototype and the session itself. Thank them.	Some useful questions are provided below. If you have a large number of participants, a simple questionnaire may be helpful.
9 Write up your notes as soon as possible and incorporate into a usability report.	
<p>Sample questions <i>during</i> the evaluation:</p> <ul style="list-style-type: none"> <li>• What do you want to do?</li> <li>• What were you expecting to happen?</li> <li>• What is the system telling you?</li> <li>• Why has the system done that?</li> <li>• What are you doing now?</li> </ul> <p>Sample questions <i>after</i> the session:</p> <ul style="list-style-type: none"> <li>• What was the best/worst thing about the prototype?</li> <li>• What most needs changing?</li> <li>• How easy were the tasks?</li> <li>• How realistic were the tasks?</li> <li>• Did giving a commentary distract you?</li> </ul>	

## Participatory heuristic evaluation

The developers of participatory heuristic evaluation (Muller *et al.*, 1998) claim that it extends the power of heuristic evaluation without adding greatly to the effort required. An expanded list of heuristics is provided, based on those of Nielsen and Mack (1994), but of course you could use any heuristics such as those introduced earlier (Chapter 4). The procedure for the use of participatory heuristic evaluation is just as for the expert version, but the participants are involved as ‘work-domain experts’ alongside usability experts and must be briefed about what is required.



## Co-discovery

Co-discovery is a naturalistic, informal technique that is particularly good for capturing first impressions. It is best used in the later stages of design.

The standard approach of watching individual people interacting with the technology, and possibly ‘thinking aloud’ as they do so, can be varied by having participants explore new technology in pairs. For example, a series of pairs of people could be given a prototype of a new digital camera and asked to experiment with its features by taking pictures of each other and objects in the room. This tends to elicit a more naturalistic flow of comment and people will often encourage each other to try interactions that they might not have thought of in isolation. It is a good idea to use people who know each other quite well. As with most other techniques, it also helps to set users some realistic tasks to try out.

Depending on the data to be collected, the evaluator can take an active part in the session by asking questions or suggesting activities, or simply monitor the interaction either live or using a video recording. Inevitably, asking specific questions skews the output towards the evaluator’s interests, but it does help to ensure that all important angles are covered. The term ‘co-discovery’ originates from Kemp and van Gelderen (1996), who provide a detailed description of its use.

### BOX 10.3

#### Living Labs

Living Labs is a European approach to evaluation that aims to engage as many people as possible in exploring new technologies. There are a number of different structures for Living Labs. For example, Nokia has teamed up with academics and other manufacturers of mobile devices to hand out hundreds of early prototype systems to students to see how they use them. Other labs work with elderly people in their homes to explore new types of home technologies. Others work with travellers and migrant workers to uncover what new technologies can do for them.

The key idea behind Living Labs is that people are both willing and able to contribute to designing new technologies and new services and it makes sense for companies to work with them. The fact that the discussions and evaluation take place in the life context of people, and often with large numbers of people, gives the data a strong ecological validity.

## Controlled experiments

Another way of undertaking participant evaluation is to set up a controlled experiment. Controlled experiments are appropriate where the designer is interested in particular features of a design, perhaps comparing one design to another to discover which is better. In order to do this with any certainty the experiment needs to be carefully designed and run.

The first thing to do when considering a controlled experiment approach to evaluation is to establish what it is that you are looking at. This is the independent variable. For example, you might want to compare two different designs of a website, or two different ways of selecting a function on a mobile phone application. Later we describe an experiment that examined two different ways of presenting an audio interface to select locations of objects (Chapter 18). The independent variable was the type of audio interface. Once you have established what it is you are looking at, you need to decide how you are going to measure the difference. These are the *dependent* variables. You might want to judge which web design is better based on the number of clicks needed to achieve some task;

speed of access could be the dependent variable for selecting a function. In the case of the audio interface, accuracy of location was the dependent variable.

Once the independent and dependent variables have been agreed, the experiment needs to be designed to avoid anything getting in the way of the relationship between independent and dependent variables. Things that might get in the way are learning effects, the effects of different tasks, the effects of different background knowledge, etc. These are the *confounding* variables. You want to ensure a balanced and clear relationship between independent and dependent variables so that you can be sure you are looking at the relationship between them and nothing else.

One possible confounding variable is that the participants in any experiment are not balanced across the conditions. To avoid this, participants are usually divided up across the conditions so that there are roughly the same number of people in each condition and there are roughly the same number of males and females, young and old, experienced and not. The next stage is to decide whether each participant will participate in all conditions (so-called within-subject design) or whether each participant will perform in only one condition (so-called between-subject design). In deciding this you have to be wary of introducing confounding variables. For example, consider the learning effects that happen if people perform a similar task on more than one system. They start off slowly but soon get good at things, so if time to complete a task is a measure they inevitably get quicker the more they do it. This effect can be controlled by randomizing the sequence in which people perform in the different conditions.

Having got some participants to agree to participate in a controlled experiment, it is tempting to try to find out as much as possible. There is nothing wrong with an experiment being set up to look at more than one independent variable, perhaps one being looked at between subjects and another being looked at within subjects. You just have to be careful how the design works. And, of course, there is nothing wrong with interviewing them afterwards, or using focus groups afterwards to find out other things about the design. People can be videoed and perhaps talk aloud during the experiments (so long as this does not count as a confounding variable) and this data can also prove useful for the evaluation.

A controlled experiment will often result in some quantitative data: the measures of the dependent values. This data can then be analyzed using statistics, for example comparing the average time to do something across two conditions, or the average number of clicks. So, to undertake controlled experiments you will need some basic understanding of probability theory, of experimental theory and, of course, of statistics. Daunting as this might sound, it is not so very difficult given a good textbook. *Experimental Design and Statistics* (Miller, 1984) is a widely used text and another good example is Cairns and Cox (2008), *Research Methods for Human–Computer Interaction*. Statistical software such as SPSS will help design and analyze your data.

### Challenge 10.4

You have just completed a small evaluation project for a tourist information 'walk-up-and-use' kiosk designed for an airport arrivals area. A heuristic evaluation by you (you were not involved with the design itself) and a technical author found 17 potential problems, of which 7 were graded severe enough to require some redesign and the rest were fairly trivial.

You then carried out some participant evaluation. You had very little time for this, testing with only three people. The test focused on the more severe problems found in the heuristic evaluation and the most important functionality (as identified in the requirements analysis). Your participants – again because of lack of time and budget –



were recruited from another section of your organization which is not directly involved in interactive systems design or build, but the staff do use desktop PCs as part of their normal work. The testing took place in a quiet corner of the development office.

Participants in the user evaluation all found difficulty with three of the problematic design features flagged up by the heuristic evaluation. These problems were essentially concerned with knowing what information might be found in different sections of the application. Of the remaining four severe problems from heuristic evaluation, one person had difficulty with all of them, but the other two people did not. Two out of the three test users failed to complete a long transaction where they tried to find and book hotel rooms for a party of travellers staying for different periods of time.

What, if anything, can you conclude from the evaluation? What are the limitations of the data?

## 10.5 Evaluation in practice

A survey of 103 experienced practitioners of human-centred design conducted in 2000 (Vredenburg *et al.*, 2002) indicates that around 40 per cent of those surveyed conducted ‘usability evaluation’, around 30 per cent used ‘informal expert review’ and around 15 per cent used ‘formal heuristic evaluation’ (Table 10.2). These figures do not indicate where people used more than one technique. As the authors note, some kind of cost–benefit trade-off seems to be in operation. Table 10.2 shows the benefits and weaknesses perceived for each method. For busy practitioners, the relative economy of review methods often compensates for the better information obtained from user testing. Clearly the community remains in need of methods that are both light on resources and productive of useful results.

← See also Oli Mival’s guide to user research in Chapter 7

**Table 10.2** Perceived costs and benefits of evaluation methods. A ‘+’ sign denotes a benefit, and a ‘–’ a weakness. The numbers indicate how many respondents mentioned the benefit or weakness.

Benefit/weakness	Formal heuristic evaluation	Informal expert review	Usability evaluation
Cost	+ (9)	+ (12)	– (6)
Availability of expertise	– (3)	– (4)	
Availability of information			+ (3)
Speed	+ (10)	+ (22)	– (3)
User involvement	– (7)	– (10)	
Compatibility with practice			– (3)
Versatility			– (4)
Ease of documentation			– (3)
Validity/quality of results	+ (6)	+ (7)	+ (8)
Understanding context	– (10)	– (17)	– (3)
Credibility of results			+ (7)

Source: Adapted from Vredenburg, K., Mao, J.-Y., Smith, P.W. and Carey, T. (2002) A survey of user-centred design practice, *Proceedings of SIGCHI conference on human factors in computing systems*, MN, 20–25 April, pp. 471–478, Table 3. © 2002 ACM, Inc. Reprinted by permission

The main steps in undertaking a simple but effective evaluation project are:

- 1 Establish the aims of the evaluation, the intended participants, the context of use and the state of the technology; obtain or construct scenarios illustrating how the application will be used.
- 2 Select evaluation methods. These should be a combination of expert-based review methods and participant methods.
- 3 Carry out expert review.
- 4 Plan participant testing; use the results of the expert review to help focus this.
- 5 Recruit people and organize testing venue and equipment.
- 6 Carry out the evaluation.
- 7 Analyze results, document and report back to designers.

### Aims of the evaluation

Deciding the aim(s) for evaluation helps to determine the type of data required. It is useful to write down the main questions you need to answer. For example, in the evaluation of the early concept for a virtual training environment the aims were to investigate:

- Do the trainers understand and welcome the basic idea of the virtual training environment?
- Would they use it to extend or replace existing training courses?
- How close to reality should the virtual environment be?
- What features are required to support record keeping and administration?

The data we were interested in at this stage was largely qualitative (non-numerical), so appropriate data-gathering methods were interviews and discussions with the trainers.

If the aim of the evaluation is the comparison of two different evaluation designs then much more focused questions will be required and the data gathered will be more quantitative. In the virtual training environment, for example, some questions we asked were:

- Is it quicker to reach a particular room in the virtual environment using mouse, cursor keys or joystick?
- Is it easier to open a virtual door by clicking on the handle or selecting the 'open' icon from a tools palette?

Figure 10.7 shows the evaluation in progress. Underlying issues were the focus on speed and ease of operation. This illustrates the link between analysis and evaluation – in this case, it had been identified that these qualities were crucial for the acceptability of the



**Figure 10.7** A trainer evaluating a training system

virtual learning environment. With questions such as these, we are likely to need quantitative (numerical) data to support design choices.

## Metrics and measures

What is to be measured and how? Table 10.3 shows some common usability metrics and ways in which they can be measured, adapted from the list provided in the usability standard ISO 9241 part 11 and using the usability definition of ‘effectiveness, efficiency and satisfaction’ adopted in the standard. There are many other possibilities.

Such metrics are helpful in evaluating many types of applications, from small mobile communication devices to office systems. In most of these there is a task – something the participant needs to get done – and it is reasonably straightforward to decide whether the task has been achieved successfully or not. There is one major difficulty: deciding the acceptable figure for, say, the percentage of tasks successfully completed. Is this 95 per cent, 80 per cent or 50 per cent? In some (rare) cases clients may set this figure. Otherwise a baseline may be available from comparative testing against an alternative design, a previous version, a rival product, or the current manual version of a process to be computerized. But the evaluation team still has to determine whether a metric is *relevant*. For example, in a complex computer-aided design system, one would not expect most functions to be used perfectly at the first attempt. And would it really be meaningful if design engineers using one design were on average two seconds quicker in completing a complex diagram than those using a competing design? By contrast, speed of keying characters may be

**Table 10.3** Common usability metrics

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Overall usability	<ul style="list-style-type: none"> <li>Percentage of tasks successfully completed</li> <li>Percentage of users successfully completing tasks</li> </ul>	<ul style="list-style-type: none"> <li>Time to complete a task</li> <li>Time spent on non-productive actions</li> </ul>	<ul style="list-style-type: none"> <li>Rating scale for satisfaction</li> <li>Frequency of use if this is voluntary (after system is implemented)</li> </ul>
Meets needs of trained or experienced users	<ul style="list-style-type: none"> <li>Percentage of advanced tasks completed</li> <li>Percentage of relevant functions used</li> </ul>	<ul style="list-style-type: none"> <li>Time taken to complete tasks relative to minimum realistic time</li> </ul>	<ul style="list-style-type: none"> <li>Rating scale for satisfaction with advanced features</li> </ul>
Meets needs for walk up and use	<ul style="list-style-type: none"> <li>Percentage of tasks completed successfully at first attempt</li> </ul>	<ul style="list-style-type: none"> <li>Time taken on first attempt to complete task</li> <li>Time spent on help functions</li> </ul>	<ul style="list-style-type: none"> <li>Rate of voluntary use (after system is implemented)</li> </ul>
Meets needs for infrequent or intermittent use	<ul style="list-style-type: none"> <li>Percentage of tasks completed successfully after a specified period of non-use</li> </ul>	<ul style="list-style-type: none"> <li>Time spent re-learning functions</li> <li>Number of persistent errors</li> </ul>	<ul style="list-style-type: none"> <li>Frequency of reuse (after system is implemented)</li> </ul>
Learnability	<ul style="list-style-type: none"> <li>Number of functions learned</li> <li>Percentage of users who manage to learn to a pre-specified criterion</li> </ul>	<ul style="list-style-type: none"> <li>Time spent on help functions</li> <li>Time to learn to criterion</li> </ul>	<ul style="list-style-type: none"> <li>Rating scale for ease of learning</li> </ul>

Source: ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs), extract of Table B.2

crucial to the success of a mobile phone. There are three things to keep in mind when deciding metrics:

- Just because something can be measured, it doesn't mean it should be.
- Always refer back to the overall purpose and context of use of the technology.
- Consider the usefulness of the data you are likely to obtain against the resources it will take to test against the metrics.

The last point is particularly important in practice.

### Challenge 10.5

*Why is learnability more important for some applications than for others? Think of some examples where it might not be a very significant factor in usability.*



## People

The most important people in evaluation are those who will use the system. Analysis work should have identified the characteristics of these people and represented these in the form of personas. Relevant data can include knowledge of the activities the technology is intended to support, skills relating to input and output devices, experience, education, training and physical and cognitive capabilities.

You need to recruit at least three and preferably five people to participate in tests. Nielsen's recommended sample of 3–5 participants has been accepted wisdom in usability practice for more than a decade. However, some practitioners and researchers advise that this is too few. We consider that in many real-world situations obtaining even 3–5 people is difficult, so we continue to recommend small test numbers as part of a pragmatic evaluation strategy.

← Relevant characteristics of people are summarized in Chapter 2

### Engagement

Games and other applications designed for entertainment pose different questions for evaluation. While we may still want to evaluate whether the basic functions to move around a game environment, for example, are easy to learn, efficiency and effectiveness in a wider sense are much less relevant. The 'purpose' here is to enjoy the game, and time to complete, for example, a particular level may sometimes be less important than experiencing the events that happen along the way. Similarly, multimedia applications are often directed at intriguing users or evoking emotional responses rather than the achievement of particular tasks in a limited period of time. In contexts of this type, evaluation centres on probing user experience through interviews or questionnaires. Read and MacFarlane (2000), for example, used a rating scale presented as a 'smiley face vertical fun meter' when working with children to evaluate novel interfaces. Other measures which can be considered are observational: the user's posture or facial expression, for instance, may be an indicator of engagement in the experience.



**FURTHER  
THOUGHTS**

However, testing such a small number makes sense only if you have a relatively homogeneous group to design for – for example, experienced managers who use a customer database system, or computer games players aged between 16 and 25. If you have a heterogeneous set of customers that your design is aimed at, then you will need to run



3–5 people *from each group* through your tests. If your product is to be demonstrated by Sales and Marketing personnel, it is useful to involve them. Finding representative participants should be straightforward if you are developing an in-house application. Otherwise participants can be found through focus groups established for marketing purposes or, if necessary, through advertising. Students are often readily available, but remember that they are only representative of a particular segment of the population. If you have the resources, payment can help recruitment. Inevitably, your sample will be biased towards cooperative people with some sort of interest in technology, so bear this in mind when interpreting your results.

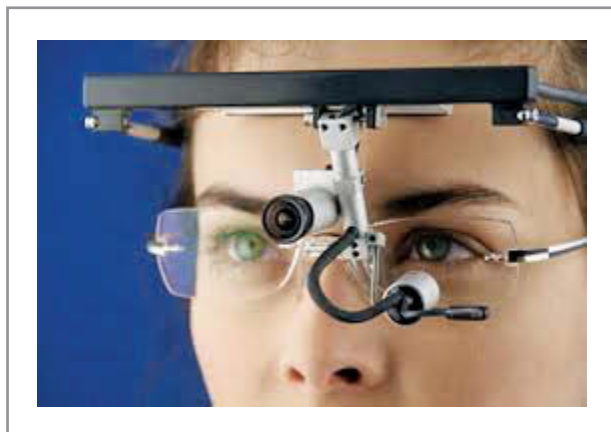
If you cannot recruit any genuine participants – people who are really representative of the target customers – and you are the designer of the software, at least have someone else try to use it. This could be one of your colleagues, a friend, your mother or anyone you trust to give you a brutally honest reaction. Almost certainly, they will find some design flaws. The data you obtain will be limited, but better than nothing. You will, however, have to be extremely careful as to how far you generalize from your findings.

Consider your own role and that of others in the evaluation team if you have one. You will need to set up the tests and collect data, but how far will you become involved? Our recommended method for basic testing requires an evaluator to sit with each user and engage with them as they carry out the test tasks. We also suggest that for ethical reasons and in order to keep the tests running you should provide help if the participant is becoming uncomfortable, or completely stuck. The amount of help that is appropriate will depend on the type of application (e.g. for an information kiosk for public use you might provide only minimal help), the degree of completeness of the test application and, in particular, whether any help facilities have been implemented.

## Physical and physiological measures

Eye-movement tracking (or ‘eye tracking’) can show participants’ changing focus on different areas of the screen. This can indicate which features of a user interface have attracted attention, and in which order, or capture larger-scale gaze patterns indicating how people move around the screen. Eye tracking is popular with website designers as it can be used to highlight which parts of the page visitors look at the most, so-called ‘hot spots’, and which are missed altogether. Measurements such as time to first fixation (TTFF), fixation length and duration can be taken to determine people’s first impressions (Lingaard, 2006) of what they see. Eye-tracking equipment is head-mounted or attached to computer monitors, as shown in Figure 10.8.

Eye-tracking software is readily available to provide maps of the screen. Some of it can also measure pupil dilation, which is taken as an indication of arousal – your pupil



**Figure 10.8** Head-mounted eye-tracking equipment

dilates if you like what you see. Physiological techniques in evaluation rely on the fact that all our emotions – anxiety, pleasure, apprehension, delight, surprise and so on – generate physiological changes.

The most common measures are of changes in heart rate, the rate of respiration, skin temperature, blood volume, pulse and galvanic skin response (an indicator of the amount of perspiration). All are indicators of changes in the overall level of arousal, which in turn may be evidence of an emotional reaction. Sensors can be attached to the participant's body (commonly the fingertips) and linked to software which converts the results to numerical and graphical formats for analysis. But there are many unobtrusive methods too, such as pressure sensors in the steering wheel of a games interface, or sensors that measure if the participant is on the edge of their seat.

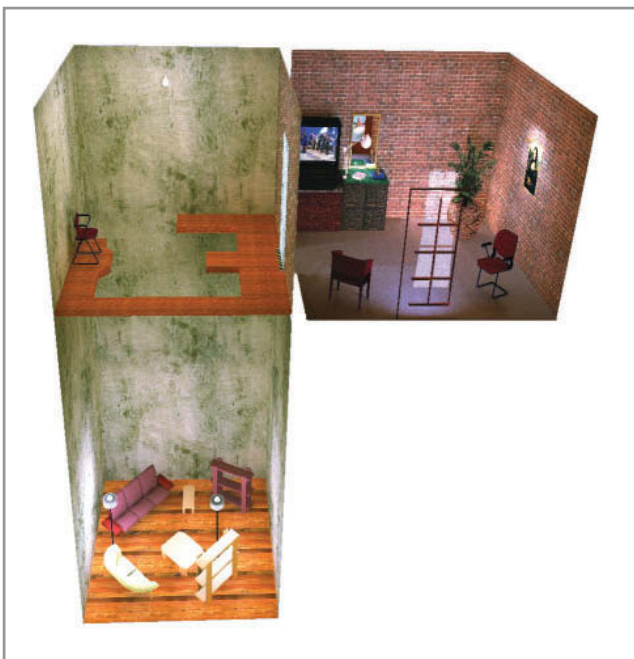
Which particular emotion is being evoked cannot be deduced from the level of arousal alone, but must be inferred from other data such as facial expression, posture or direct questioning. Another current application is in the assessment of the degree of presence – the sense of 'being there' evoked by virtual environments (see Figure 10.9).

Typically, startling events or threatening features are produced in the environment and arousal levels measured as people encounter them. Researchers at University College London and the University of North Carolina at Chapel Hill (Usoh *et al.*, 1999, 2000; Insko, 2001, 2003; Meehan, 2001) conducted a series of experiments when measuring arousal as participants approach a 'virtual precipice'. In these circumstances changes in heart rate correlated most closely with self-reports of stress.

Another key aspect of the evaluation is the data that you will need to gather about the people. Increasingly there is a host of measures that can provide real insight into UX. Galvanic skin response (GSR), for example, measures the level of arousal that a person is experiencing. A sensor placed on the user's skin will record how much perspiration there is and hence how aroused they are. Eye tracking can be used to see where people are looking. Face recognition can determine whether people are looking happy or sad, confused or angry. The Facial Action Coding System (FACS) is a robust way of measuring emotion through facial expression. Pressure sensors can detect how tightly people are gripping something. Of course, video can be used to record what people are doing. These various measures can be combined into a powerful way of evaluating UX.

→ There is more about the role of emotion in interactive systems design in Chapter 22

→ See Chapter 22, Affect, for more details on emotion in humans



**Figure 10.9** A 20-foot 'precipice' used in evaluating presence in virtual environments

(Source: Reprinted from *Being There: Concepts, Effects and Measurement of User Presence in Synthetic Environment*, Insko, B.E., Measuring presence. © 2003, with permission from IOS Press)

## The test plan and task specification

A plan should be drawn up to guide the evaluation. The plan specifies:

- Aims of the test session
- Practical details, including where and when it will be conducted, how long each session will last, the specification of equipment and materials for testing and data collection, and any technical support that may be necessary
- Numbers and types of participant
- Tasks to be performed, with a definition of successful completion. This section also specifies what data should be collected and how it will be analyzed.

You should now conduct a pilot session and fix any unforeseen difficulties. For example, task completion time is often much longer than expected, and instructions may need clarification.

## Reporting usability evaluation results to the design team

However competent and complete the evaluation, it is worthwhile only if the results are acted upon. Even if you are both designer and evaluator, you need an organized list of findings so that you can prioritize redesign work. If you are reporting back to a design/development team, it is crucial that they can see immediately what the problem is, how significant its consequences are, and ideally what needs to be done to fix it.

The report should be ordered either by areas of the system concerned or by severity of problem. For the latter, you could adopt a three- or five-point scale, perhaps ranging from ‘would prevent participant from proceeding further’ to ‘minor irritation’. Adding a note of the general usability principle concerned may help designers understand why there is a difficulty, but often more specific explanation will be needed. Alternatively, sometimes the problem is so obvious that explanation is superfluous. A face-to-face meeting may have more impact than a written document alone (although this should always be produced as supporting material) and this would be the ideal venue for showing *short* video clips of participant problems.

Suggested solutions make it more probable that something will be done. Requiring a response from the development team to each problem will further increase this probability, but may be counter-productive in some contexts. If your organization has a formal quality system, an effective strategy is to have usability evaluation alongside other test procedures, so usability problems are dealt with in the same way as any other fault. Even without a full quality system, usability problems can be fed into a ‘bug’ reporting system if one exists. Whatever the system for dealing with design problems, however, tact is a key skill in effective usability evaluation.

## 10.6 Evaluation: further issues

Of course, there are many specifics associated with evaluation and the contexts in which evaluation takes place. Many of these are covered in Chapters 14–20 on specific contexts of UX. In this section we look at a number of particular issues.

### Evaluating usability

There are several standard ways of measuring usability, but probably the best known and most robust is the System Usability Scale (SUS). Jeff Sauro presents the scale as

illustrated in Figure 10.10. He suggests that any score over 68 is above average and indicates a reasonable level of usability.

### The System Usability Scale

The SUS is a 10 item questionnaire with 5 response options.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The SUS uses the following response format:

Strongly Disagree 1	2	3	4	Strongly Agree 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Scoring SUS

- For odd items: subtract one from the user response.
- For even-numbered items: subtract the user responses from 5
- This scales all values from 0 to 4 (with four being the most positive response).
- Add up the converted responses for each user and multiply that total by 2.5. This converts the range of possible values from 0 to 100 instead of from 0 to 40.

**Figure 10.10** The System Usability Scale

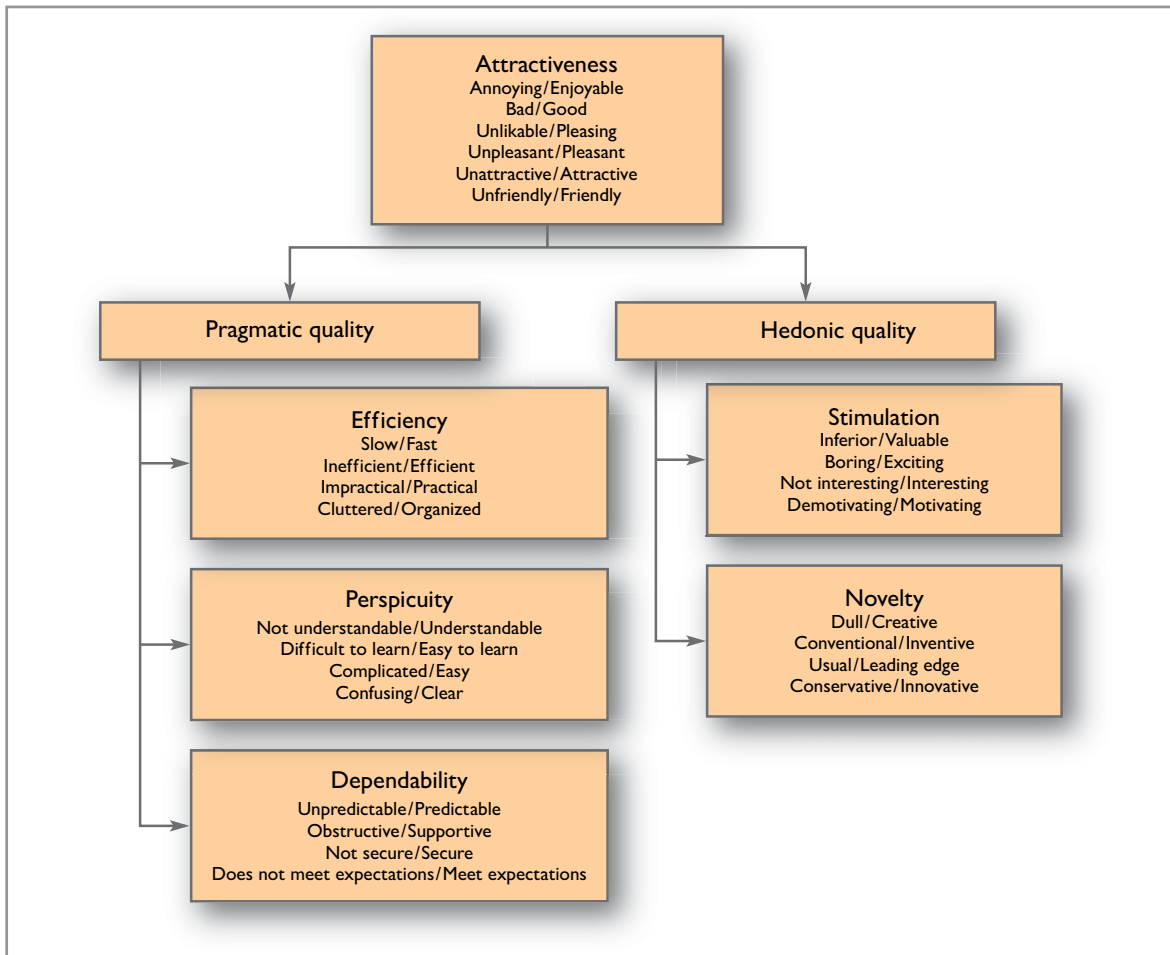
## Evaluating UX

There are a number of tools and methods specifically aimed at evaluating user experience. They differentiate between the pragmatic qualities of the UX and the hedonic qualities (Hassenzahl, 2010). The User Experience Questionnaire describes these qualities as illustrated in Figure 10.11. A 26-item questionnaire is used to gather data about a UX (Figure 10.12). Online spreadsheets are available to help with the statistical analysis of the data gathered.

An alternative is to use the AttrakDiff online questionnaire (Figure 10.13). This has a similar approach but uses different terms. Both of these questionnaires can be used as they are and this has the advantage that comparisons can be made across products and services. For specific evaluation, however, UX designers may need to change the terms used on the semantic differential scales.

## Evaluating presence

Designers of virtual reality (VR) and augmented reality (AR) applications are often concerned with the sense of presence, of being ‘there’ in the virtual environment rather than ‘here’ in the room where the technology is being used. A strong sense of presence is thought to be crucial for such applications as games, those designed to treat phobias,



**Figure 10.11** The User Experience Questionnaire


to allow people to ‘visit’ real places they may never see otherwise, or indeed for some workplace applications such as training to operate effectively under stress. This is a very current research topic and there are no techniques that deal with all the issues satisfactorily. The difficulties include:

- The sense of presence is strongly entangled with individual dispositions, experiences and expectations. Of course, this is also the case with reactions to any interactive system, but presence is an extreme example of this problem.
- The concept of presence itself is ill-defined and the subject of much debate among researchers. Variants include the sense that the virtual environment is realistic, the extent to which the user is impervious to the outside world, the retrospective sense of having visited rather than viewed a location, and a number of others.
- Asking people about presence while they are experiencing the virtual environment tends to interfere with the experience itself. However, asking questions retrospectively inevitably fails to capture the experience as it is lived.

The measures used in evaluating presence adopt various strategies to avoid these problems, but none is wholly satisfactory. The various questionnaire measures, for example the questionnaire developed by NASA scientists Witmer and Singer (1998), or the range of instruments developed at University College and Goldsmiths College, London (Slater, 1999; Lessiter *et al.*, 2001), can be cross-referenced to measures which attempt

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	2
creative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	supportive	11
good	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efficient	20
clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical	22
organized	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	innovative	26

Figure 10.12 Questionnaire used to gather data about a UX


Deutsch | **English**

---

**Assessment of [www.attrakdiff3.de](http://www.attrakdiff3.de)**

With the help of the word pairs please enter what you consider the most appropriate description for [www.attrakdiff3.de](http://www.attrakdiff3.de).

Please click one item in every line.

human*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	technical
isolating*	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	connective
pleasant*	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unpleasant
inventive*	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional
simple*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	complicated
professional*	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unprofessional
ugly*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	attractive
practical*	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	impractical
likeable*	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	disagreeable
cumbersome*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	straightforward

\* required field

Back
Continue

Figure 10.13 The AttrakDiff online questionnaire



to quantify how far a person is generally susceptible to being ‘wrapped up’ in experiences mediated by books, films, games and so on as well as through virtual reality. The Sense of Presence Inventory (SOPI) can be used to measure media presence. The Witmer and Singer Immersive Tendencies Questionnaire (Witmer and Singer, 1998) is the best known of such instruments. However, presence as measured by presence questionnaires is a slippery and ill-defined concept. In one experiment, questionnaire results showed that while many people did not feel wholly present in the virtual environment (a re-creation of an office), some of them did not feel wholly present in the real-world office either (Usoh *et al.*, 2000). Less structured attempts to capture verbal accounts of presence include having people write accounts of their experience, or inviting them to provide free-form comments in an interview. The results are then analyzed for indications of a sense of presence. The difficulty here lies in defining what should be treated as such an indicator, and in the layers of indirection introduced by the relative verbal dexterity of the participant and the interpretation imposed by the analyst.

Other approaches to measuring presence attempt to avoid such layers of indirection by observing behaviour in the virtual environment or by direct physiological measures.



### Challenge 10.6

*What indicators of presence might one measure using physiological techniques? Are there any issues in interpreting the resulting data?*

## Evaluation at home

People at home are much less of a ‘captive audience’ for the evaluator than those at work. They are also likely to be more concerned about protecting their privacy and generally unwilling to spend their valuable leisure time in helping you with your usability evaluation. So it is important that data-gathering techniques are interesting and stimulating for users and make as little demand on time and effort as possible. This is very much a developing field and researchers continue to adapt existing approaches and develop new ones. Petersen *et al.* (2002), for example, were interested in the evolution over time of relationships with technology in the home. They used conventional interviews at the time the technology (a new television) was first installed, but followed this by having families act out scenarios using it. Diaries were also distributed as a data-collection tool, but in this instance the non-completion rate was high, possibly because of the complexity of the diary pro forma and the incompatibility between a private diary and the social activity of television viewing.

An effective example of this in early evaluation is reported in Baillie *et al.* (2003) and Baillie and Benyon (2008). Here the investigator supplied users with Post-it notes to capture their thoughts about design concepts (Figure 10.14). An illustration of each different concept was left in the home in a location where it might be used and users were encouraged to think about how they would use the device and any issues that might arise. These were noted on the Post-its, which were then stuck to the illustration and collected later.

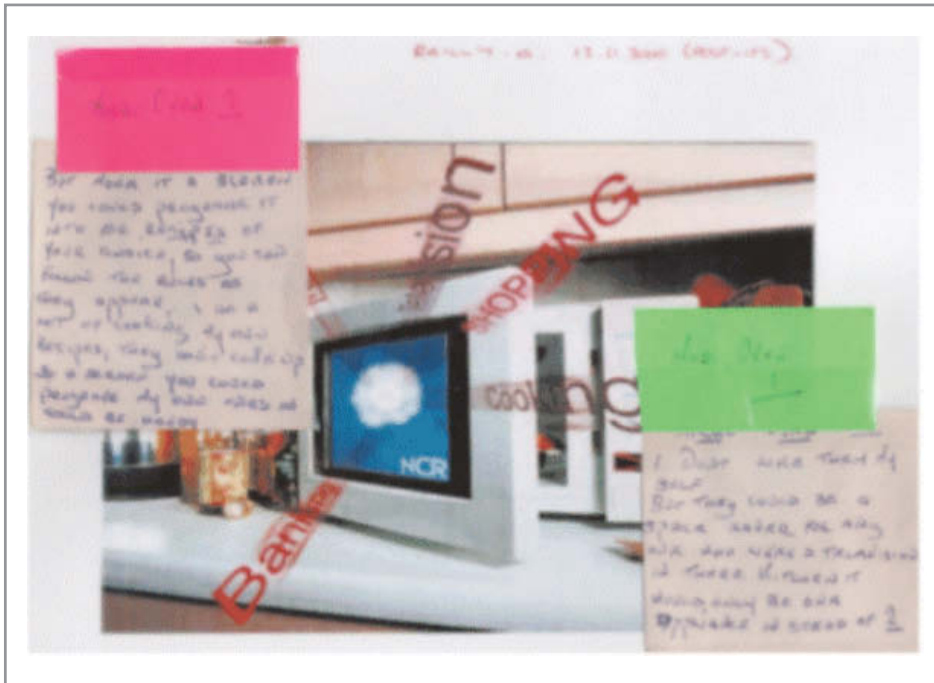
Where the family is the focus of interest, techniques should be engaging for children as well as adults – not only does this help to ensure that all viewpoints are covered but working with children is a good way of drawing parents into evaluation activities.

← Probes described in Chapter 7 are relevant here



### Challenge 10.7

*Suggest some ways in which 6–9 year olds could take part in evaluation activities situated in the home.*



**Figure 10.14** Post-it notes

(Source: David Benyon)

## Summary and key points



This chapter has presented an overview of the key issues in evaluation. Designing the evaluation of an interactive system, product or service requires as much attention and effort as designing any other aspect of that system. Designers need to be aware of the possibilities and limitations of different approaches and, in addition to studying the theory, they need plenty of practical experience.

- Designers need to focus hard on what features of a system or product they want to evaluate.
- They need to think hard about the state that the system or product is in and hence whether they can evaluate those features.
- Designers can gather and study data analytics on the performance of their service.
- There are expert-based methods of evaluation.
- There are participant-based methods of evaluation.
- Designers need to design their evaluation to fit the particular needs of the contexts of use and the activities that people are engaged in.

## Exercises

- 1 Using the list of heuristics from Section 10.3, carry out a heuristic evaluation of the features dealing with tables in your usual word processor and the phone book in your mobile phone.
- 2 Think about the following evaluation. A call centre operator answers enquiries about insurance claims. This involves talking to customers on the phone while



accessing their personal data and claim details from a database. You are responsible for the user testing of new database software to be used by the operators. What aspects of usability do you think it is important to evaluate, and how would you measure them?

Now think about the same questions for an interactive multimedia website which is an online art gallery. The designers want to allow users to experience concrete and conceptual artworks presented in different media.

- 3 (More advanced) Look at the terms used by the UX questionnaires and discuss how applicable they are. Do you interpret them in the same way as your colleague does?
- 4 (More advanced) You are responsible for planning the evaluation of an interactive toy for children. The toy is a small, furry, talking animal character whose behaviour changes over time as it 'learns' new skills and in response to how its owner treats it, for example how often it is picked up during a 24-hour period. The designers think it should take around a month for all the behaviours to develop. Children interact with the toy by speaking commands (it has voice recognition for 20 words), stroking its ears, picking it up, and pressing 'spots' on the animal's back which are, in effect, buttons triggering different actions. No instructions will be provided; children are intended to find out what the toy does by trial and error.  
Design an evaluation process for the toy, explaining the reasons behind your choices.
- 5 How do we know that the criteria we use for evaluation reflect what is important to users? Suggest some ways in which we can ground evaluation criteria in user wants and needs.
- 6 An organization with staff in geographically dispersed offices has introduced desktop video-conferencing with the aim of reducing resources spent on 'unnecessary' travel between offices. Working teams often involve people at different sites. Before the introduction of video-conferencing, travel was regarded as rather a nuisance, although it did afford the opportunity to 'show one's face' at other sites and take care of other business involving people outside the immediate team. One month after the introduction of the technology, senior managers have asked for a 'comprehensive' evaluation of the system. Describe what techniques you would adopt, what data you would hope to gain from their use and any problems you foresee with the evaluation.



## Further reading

Cairns, P. and Cox, A.L. (2008) *Research Methods for Human-Computer Interaction*. Cambridge University Press, Cambridge.

Cockton, G., Woolrych, A. and Lavery, D. (2012) Inspection-based evaluations. In Jacko, J.A. (ed.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, 3rd edn. CRC Press, Taylor & Francis, Boca Raton, FL, pp. 1279–1298.

Monk, A., Wright, P., Haber, J. and Davenport, L. (1993) *Improving Your Human-Computer Interface: A Practical Technique*. BCS, Practitioner Series, Prentice-Hall, New York and Hemel