Usability: Practical Methods for Testing and Improvement

Miles Macleod

National Physical Laboratory, Teddington, Middlesex, TW11 0LW, UK

Telephone: +44 81 943 6097
Fax: +44 81 977 7091
Email: miles@hci.npl.co.uk

Introduction

Most developers of interactive software and information systems want people to find their products effective, efficient and satisfying to use. People generally also want software to be easy to install, easy to learn (or to guess how to use), and usable with acceptable mental effort. Such usability issues are increasingly significant. They can be critical elements for the success of a product in a world marketplace where there are growing expectations of usability, and where many producers now give usability a high profile in their advertising. Major companies are investing in usability engineering and evaluation facilities. But what practical tools and methods are generally available to help test and improve usability?

This paper seeks to present usability evaluation and improvement in a general and business context. It gives an overview of some state-of-the-art methods and tools, which have been developed in collaboration with industry to meet commercial usability evaluation needs. It focuses on user-based and analytic methods, developed over the past four years by the MUSiC (Measuring Usability of Systems in Context) consortium, which are being taken up by many commercial organisations. It emphasises contextual aspects, with examples of how the methods are being taken up and incorporated into (iterative, user oriented) development processes. Successful commercial take-up means that a variety of people within an organisation – particularly users and managers with a stake in usability – become involved in the usability engineering and testing. Usability is the legitimate concern of many different people, and active participation by the various stakeholders is central to achieving improved usability.

Benefits of improved usability

As well as meeting the increasing expectations, there are many tangible benefits for purchasing organisations from improved usability of interactive systems. Work productivity and efficiency are higher when using IT systems with good usability, and there are fewer 'user errors'; less training of staff is required to enable effective and efficient use the system, users are more satisfied, and there may be lower staff turnover. There are benefits both for users and producers in that less support and documentation is required. Such benefits can be quantified, and incorporated into a business case for applying usability engineering in system development. Karat (1993) gives numerous examples.

A further incentive for companies marketing or using information systems in the European Union is the European Directive 90/270/EEC (CEC, 1990) concerning work with display screens, which has been implemented in national law in member countries. This is primarily concerned with the physical ergonomics of workstations, but makes some interesting demands concerning usability: “software must be suitable for the task ... software must be easy to use ... systems must display information in a format and at a pace which are adapted to users ... [and] the principles of software ergonomics must be applied”.

The scope of usability

Usability professionals frequently meet people who think usability can be added to a system just by providing a specific style of interface. However, usability is much deeper than the superficial features of the user interface. While user interface features are important in helping shape the usability of the system, simply employing a good set of widgets does not guarantee usability – interface components are the building blocks for constructing parts of a system, with the aim of adequately presenting to the user clearly understandable information and feedback, and providing an appropriate means of entering data and commands easily and efficiently. Adding, for example, a graphical interface – even one with easy-to-use objects – may not give a system an acceptable level of usability unless the design meets wider usability needs. Most style guides acknowledge this; for example, the Open Look™ graphical user interface application style guidelines (SUN Microsystems, 1990): “The presence of these graphical elements does not guarantee good application design; that depends on you”.

What then is usability? It can be thought of as quality of use, a quality of the interaction between user and system. Now, this is where usability is more difficult to reason about than some other qualities of software products. Usability depends upon the characteristics of the user as well as the software. A system can have excellent quality of use for some people and poor quality of use for others. For example, a graphical user interface may have simple, well structured menus – which novices can explore and use successfully and safely – but can be very frustrating for experienced, frequent users because it lacks keyboard short cuts.

Usability also depends on the specific tasks people want to perform. For example, a database may have adequate usability for repetitive data entry tasks, but poor usability for generating reports. Usability is also affected by environmental factors, from physical influences such as incident light and sounds, to organisational factors such as interruptions in mid task.

Hence we should think of usability in terms of the quality of use **of** an interactive system **by** its (intended) users **for** achieving specific work goals and tasks **in** particular work environments. This view is reflected in an emerging international standard, ISO 9241 - 11 (ISO 1993b), which provides guidance on assessing usability in terms of "the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in particular environments".

Design, testing and improvement

It is not the intention here to prescribe how to *design* usable systems. Some valuable relevant general texts are available (e.g. Nielsen, 1993; Norman, 1987); and copious guidance is provided in the various parts of ISO 9241 (ISO, 1993a). But it is worth considering several pieces of advice which are widely accepted. Firstly, follow good high-level design principles. A very clearly and concisely expressed set of such principles can be found in the opening pages of 'Human Interface Guidelines: The Apple Desktop Interface' (Apple Computer, 1987). Further principles, with examples of their use, are given in ISO 9241-10 (ISO 1993c). Secondly, if appropriate follow a specific style guide, which should enable the developer to provide consistency of interaction style, both within the application and with other applications familiar to the users. Thirdly, and most importantly, involve users in the process of design and development. To develop IT systems which adequately meet user needs requires the *participation* of (intended) users, along with other stakeholders, from requirements capture to acceptance testing or market release.

Before an IT system is implemented, most people are not very good at visualising from its specification how it will look, feel and perform, and how well the design will meet users' needs. [Note however that there are analytic methods, such as SANe (described below), which enable aspects of usability to be analysed from system specifications and models of tasks and users.] There is ample evidence from systems developed using one-shot (waterfall) development methods that even gifted designers following good human factors principles cannot anticipate how users will react to an implemented system. This is not to deny the value of early expert input from usability professionals, who can draw on their knowledge and experience of how designs can meet user needs, or the use of checklists (e.g. Ravden and Johnson, 1989) and heuristic evaluation (Nielsen and Molich, 1990; Nielsen, 1992). However, the early and continuing use of mock-ups and prototypes to gain user feedback stimulates constructive criticism and suggestions for improvement directly related to user needs, which can be fed back into development iteratively. In effect, this introduces user-based evaluation into the development process. Essential for its success are the choice of representative users, and tasks which are representative of intended system use.

Evaluation, validity and context

Methods for usability evaluation may have various purposes: for example, to shape design (or redesign) to meet user needs; to identify and diagnose problems; to evaluate implementation (for comparison with other designs and systems, and for acceptance testing). The data collected may be qualitative (for assessment of factors relevant to usability) or quantitative (measures which indicate usability). Note that quantitative data do not necessarily give a valid indication of the level of usability. For example, many approaches to usability evaluation focus specifically on user problems. Reducing problems is a desirable aim, and is very valuable for improving usability, but unless the problems are carefully graded for severity, a problem count can be a misleading indicator of overall usability. Most users would rather encounter several trivial problems than a single catastrophic problem which causes task failure. Evaluation can be applied at different stages during development: at specification (analytic and expert methods); when a mock-up or prototype is available (expert and user-based methods); and after implementation (user-based methods). In general, the earlier evaluation can be applied, the more cost-effective it is, so long as it is valid.

There are some underlying questions relating to validity which should be asked in planning any evaluation:

- Are we looking at the right things? This relates to the ecological validity of the evaluation: how representative of real-world use what is being evaluated is; how safe it is to generalise from the limited scope of the evaluation to the wider use of the product in the workplace.
- Are we collecting the right data? Is the specific data we are gathering relevant to usability and quality of use? It is all too easy to collect large quantities of data of little relevance to usability, obscuring relevant data, and making analysis difficult. This is particularly true of automatic system monitoring (see Maguire and Sweeney, 1989), which may also have dubious ecological validity unless its collection is carefully controlled.
- Are we analysing the data appropriately? The analysis of the raw data must reliably deliver valid indicators of usability.

As already discussed in this paper, usability depends not only on the characteristics of the product itself, but also the characteristics of users, the tasks they perform and their work environments. Together, these can be described as the 'context of use' of the product. For an evaluation to be ecologically valid, the contextual factors which bear on the evaluation (the 'context of evaluation') must reflect the intended context of use.

MUSiC methods for measuring usability in context

The need for practical, valid methods for the quantitative evaluation of usability in context – which can provide the usability measures required by management – underlay the work of MUSiC (ESPRIT Project 5429), a focus for European developments in this area from 1990-93. The MUSiC consortium is continuing to work together, and includes industrial and academic partners from six countries. After developing and verifying usability measures and evaluation methods, the major thrust of the MUSiC project's second phase was the use and refinement of the methods and tools in commercial contexts, and the development of training packages and support services for companies adopting the methods. The MUSiC consortium has been marketing the resulting tools and services since mid 1993. Their further development is continuing to shape them as closely as possible to current and anticipated commercial needs.

MUSiC has produced a set of evaluation methods – analytic and user-based – from which evaluators can choose to adopt methods individually or in combination to measure those aspects which they (or the developers or procurers) consider most important. Significant outputs of MUSiC include the usability context analysis method and guide; analytic measures supported by the SANe toolkit; and a set of user-based methods for measuring usability, including the core factors (effectiveness, efficiency and user satisfaction) and the cognitive workload imposed when using a system. MUSiC methods also enable measurement of the rate at which people can learn to use a system, and of its initial usability for first time users. As well as measures, the methods deliver diagnostic data which help identify usability problems and their causes.

MUSiC user-based methods include *performance measurement* – supported by the Performance Measurement Handbook and a software tool, DRUM, the Diagnostic Recorder for Usability Measurement – and *measurement of perceived usability* supported by a fifty item questionnaire, SUMI, the Software Usability Measurement Inventory. Work at the Technical University of Delft, Netherlands, has advanced methods for measuring the *cognitive workload* (mental effort) imposed upon users (Wiethoff et al. 1991). Objective measures are obtained by monitoring variations in heart rate; subjective measures by use of questionnaires. All these user-based methods require a working prototype or implemented system.

An analytic method, applicable early in design, has been developed at Westfälische Wilhelms Universität, Münster, Germany. This enables analysis of the specification of an interactive system, before any design decisions are committed to code, in order to predict aspects of usability, and identify areas where improvements can be made. It is supported by the SANe Toolkit: effectively a CASE tool which derives predictive analytic usability measures from device, task and simple user models (Gunsthövel et al., 1993). As with any CASE tool, use of the SANe toolkit requires training in the underlying method; the software itself is relatively easy to use. SANe has been successfully applied to early usability evaluation of designs in a range of major projects, including a complex workstation in the testbed for the European Space Agency Columbus Project.

Usability Context Analysis

Usability Context Analysis (UCA) helps developers and evaluators deal with contextual issues in usability. It firstly provides a simple, structured method for describing key features of the users, tasks and work-environments for which a system is designed: the context of use. It secondly gives evaluators a method for characterising the users, tasks and environments appropriate for *evaluation* of the system, and for documenting how accurately this context of evaluation matches the intended context of use. UCA employs a questionnaire format. This record provides an indication of ecological validity. An evaluation which studies how well atypical users perform unrepresentative tasks in highly artificial circumstances is unlikely to give a worthwhile prediction of the usability of the implemented system. The Usability Context Analysis method and guide (Macleod et al., 1993; Macleod, in press) were developed in collaboration with industry as part of MUSiC, by the UK National Physical Laboratory and HUSAT Research Institute.

A less obvious aim of UCA is to increase understanding of user needs among people responsible for system development, and to establish a shared understanding of contextual factors. This is facilitated by applying UCA cooperatively. The need for increased user involvement in system development is widely recognised, and there is an honourable history of user participation in Scandinavian information system development. However, it is still commonplace in Europe for IT development to be led from a software-centred perspective (Dillon et al. 1993). Even where companies employ human factors engineers to help match products and systems to users' needs, the findings of human factors evaluations may be ignored (e.g. Wasserman 1991), or misunderstood by managers.

Usability Context Analysis involves a range of people with a stake in the quality of use of the system. They may be drawn from: product managers; project managers (procuring or developing systems); designers; quality managers; users; user support managers ; documentation managers and technical writers; training managers; change management teams; work process analysts; people with responsibility for certification and audit; people with responsibility for health and safety; and human factors and usability professionals. Where UCA is applied cooperatively, much of the formal work is carried out in one or more group context meetings. The prerequisite is to identify and bring together a number of key people, typically six to ten.

In NPL's work with client organisations, we encourage the formation of a usability team including stakeholders representing different interests, early in the development of an IT system. Members of the usability team participate in the context study, and contribute to the planning and conduct of evaluations. They have a continuing role of liaison, informing and maintaining contact with other stakeholders throughout development. As a minimum the team should include someone with human factors knowledge, someone representing user interests and someone from the development team. We recommend a training course in the use of MUSiC methods, including UCA. The usability team is responsible for arranging participation in the context meeting. Participants must be provided in advance with a copy of the Context Questionnaire, and briefed adequately about the aims of UCA, and the information they should have available at the meeting. The characterisation of context of use can then be achieved cooperatively in a single group meeting. This involves working through the context questionnaire to record characteristics of users, tasks and environments, and agreeing under each heading a fairly high level description of that characteristic of the context of use.

The aim of the subsequent steps in UCA is to ensure that evaluation is carried out in a context, or in a specified subset of contexts, which fairly reflect real-life system use. We recommend that this part of UCA is performed separately from recording context of use. It requires at least one person with a background in human factors or HCI, working in consultation with other stakeholders and members of the usability team. The analyst considers each documented factor of the context of use, and assesses its relevance to usability. For each contextual factor identified as possibly affecting usability, the usability team must decide how to control or monitor that factor in the evaluation, to achieve the specific objectives of the evaluation. Key issues are the choice of tasks for evaluation (e.g. critical or frequently performed tasks), and the profile of users. The result is a summary of the context of evaluation, which specifies the conditions under which the evaluation should take place. The team derives an evaluation plan from this information, describing the practical details of how the evaluation will be carried out.

User satisfaction – perceived usability

One of the simplest means of testing usability is to ask users – to sample their subjective views. This can be achieved in a structured way by using a questionnaire. Properly conducted and analysed – with due consideration of contextual factors – this approach can provide valid and reliable measures of user satisfaction, which complement objective measures of performance. It is worth noting that user satisfaction is not necessarily closely correlated with performance: users may have positive perceptions of software applications which are not objectively particularly efficient to use, and vice versa.

The Software Usability Measurement Inventory, SUMI, (Kirakowski et al., 1992; Kirakowski and Corbett, 1993) is a fifty item, internationally-standardised questionnaire for quantitative measurement of how usable a product is, in the view of the user. SUMI was developed within MUSiC, by the Human Factors Research Group at University College Cork, Ireland. It is available in Dutch, English, French, German, Italian, Spanish and Swedish versions. SUMI gives a *Global* measure of usability, together with measure of five relatively orthogonal factors: *Efficiency, Affect, Helpfulness, Control and Learnability,* which have been empirically identified as dimensions of perceived usability. It also gives diagnostic data. The method is quick and straightforward to apply.

The validity and reliability of SUMI have been established internationally, and it offers convenient and inexpensive collection of trustworthy data about the usability of a product. A handbook (Porteous et al., 1993) guides the correct use of the questionnaire. Analysis of SUMI data is supported by the dedicated 'SUMISCO' software.

SUMI can be used in two ways: survey and controlled study. SUMI surveys measure the perceived usability of software systems already in use. For example, in response to the European Display Screens Directive, a company measured the perceived usability of software used by its employees – many hundreds of users. For several classes of software application, SUMI analysis gives a comparison with the expected level of usability for that particular class of application.

SUMI is widely used in controlled studies, where performance is also measured and problems identified and analysed. A minimum of about ten users is required. In usability evaluations at NPL we give users SUMI after they have completed – or attempted – the selected representative tasks (see performance measurement, below). It generally takes users between five and ten minutes to complete the SUMI questionnaire. Subsequent analysis using SUMISCO derives the measures, and gives diagnostic information by Item Consensual Analysis, which identifies specific questions where the responses are unusually negative (or positive). This can be used to identify areas of difficulty, and to guide follow-up interviews with those users about the nature of the difficulties.

Performance measurement

Performance measurement gives objective measures of usability, which complement measures of user satisfaction. It involves observing and analysing (preferably with the help of a video recording) the use of a system by representative users performing selected work tasks. The basic principles of usability testing are well established (e.g. Whiteside et al., 1988). A number of companies have developed in-house methodologies. However, there has not been a generally available, validated and well-documented method and set of measures, and there has been a lack of tools to support and improve the cost-effectiveness of such work.

The MUSiC Performance Measurement Method, developed at NPL in collaboration with industry, provides a validated method for performance-based usability measurement, involving observation of representative users performing selected tasks. The full method is video-assisted, with software support for management and analysis of data. The Performance Measurement Handbook (Rengger et al., 1993) introduces and explains the method, and provides step by step instructions for each stage, together with relevant background information. The method supports assessment of performance-related factors in the quality of use of a system, in context. It is designed to be applied in conjunction with Usability Context Analysis.

Performance measurement is typically applied in evaluating the usability of prototype systems during development, and in acceptance testing. It enables comparison with performance using other IT systems, or with other ways of achieving the same work goals. Studies are usually carried out in a usability laboratory. Alternatively, data can be captured in the workplace, for example when key factors in an information system or its environmental setting cannot adequately be replicated in a laboratory. Appropriate evaluation tasks and user profiles, to meet the specific objectives of the evaluation, will have been agreed as a result of the context study. Together with other stakeholders, the usability team defines assessment rules for task output, for determining correctness and completeness of task goal achievement.

A major practical consideration in planning an evaluation is the availability of users matching the required profile. Finding representative users is of great importance in any user-based evaluation, and may require careful pre-planning. It may, for example, be necessary to provide training in the use of a new or revised system, which matches the training users will have received on roll-out of the system. Our experience shows that usability testing can identify shortcomings in training and user support, as well as in the design of software. Typically, the usability team takes responsibility for user arrangements, and for preparing task instructions.

The MUSiC Performance Measurement Method gives measures of core indicators of usability: *Effectiveness* (how correctly and completely goals are achieved in context) and *Efficiency* (which relates effectiveness to cost in terms of time). It gives measures of *Productive Period'*(the proportion of time spent not having problems), and of *Snag, Search* and *Help* times: time spent overcoming problems, searching unproductively through a system, and seeking help. These problem-related measures are valuable sources of diagnostic data, and provide pointers to causes of problems.

It is possible to tailor the method to individual evaluation needs, for example by choosing not to analyse productive period – thus reducing analysis time – or by adding specific measures such as counts of errors. The Basic MUSiC Performance Measurement Method is a minimal version of the method, giving measures of effectiveness and efficiency without use of video. It employs manual timing of task performance, together with assessment of task outputs. This takes less effort than the full method, but gives reduced diagnostic data: it relies on accurate observation in real time, plus analysis of task output. The basic method may be employed in quick, small scale evaluations, or where the budget is very constrained.

Software support for video-assisted usability evaluation

User-based evaluations – when they take suitable account of contextual factors – can offer significant benefits over expert methods, in terms of the quality and validity of the usability data obtained. Although it is relatively costly to stage evaluations with representative users, tasks, and settings (compared with employing a usability professional to give an expert evaluation), the experience of many organisations shows that the extra cost is justified in many situations, by demonstrable additional benefits (Karat, 1993). In our work at NPL, we employ both expert and user-based approaches as appropriate, in combination or at different stages of development. One area in which cost can be reduced without prejudicing the quality of the evaluation is the management and analysis of data, by using suitable tools.

A major focus of MUSiC was the development of tools to support the analysis of data, as well as the correct application of the methods. For example, SUMISCO software facilitates the rapid analysis of data from the SUMI questionnaire. There is an even greater need for support for analysis of observational data, to derive objective performance measures and diagnostic information. While observation can yield rich usability data, analysis solely in real time may lose valuable data. Video recordings preserve data, yet can be very time-consuming to analyse, conventionally taking ten or more hours to analyse a single hour of video. The Diagnostic Recorder for Usability Measurement, DRUM (Macleod and Rengger, 1993), helps reduce analysis time to 2 or 3 hours per hour of recording, and supports first-pass analysis in real time.

DRUM provides support for many aspects of evaluation. It assists: management of data throughout evaluation; task analysis; video control; creation of an interaction log of each evaluation session; analysis of logged data; and derivation of the MUSiC performance measures and metrics, and user-defined measures. DRUM also supports the identification of analyst-defined critical incidents, and diagnosis of specific usability problems. Once logged, any observed event can be automatically found and reviewed on the video.

Iteratively developed over three years – in collaboration with a number of commercial organisations – to meet the needs of usability testing in development and procurement, DRUM is now in version 3.1. DRUM requires an Apple Macintosh II computer (e.g. FX, Centris, or Quadra) with a 14″ monitor (formerly marketed as 13′′) or larger , and an extended keyboard; System 7; and HyperCard 2.1 (allocated at least 1.8 MB RAM). DRUM drives the following VCRs: Sony U-Matic VO 7000 and 9000 series, with BKU 701 computer interface and FCG-700 frame code generator; and the Panasonic AG-7350 or AG-7355 SuperVHS with AG-IA232TC RS-232 board and time code generator/reader. It should be emphasised that the validity of the MUSiC usability measures derived by DRUM in any evaluation depends upon the contextual validity of the evaluation, and on the evaluators applying the MUSiC Performance Measurement Method for collecting and analysing the data.

Usability evaluation in practice

Usability work at NPL includes evaluation of a wide range of information systems and IT products for clients, and transfer of evaluation skills through training courses and collaboration. Our work involves us with a wide variety of organisations, both through commercial usability evaluations and through the activities of the NPL Usability Forum (NPL, 1994). [This aims to disseminate information and promote discussion on usability and its evaluation, with an emphasis on practical issues. At the time of writing, Forum events have involved 370 different participants from 160 organisations.] We find that many developers have in the past only given thought to usability late in system development. By that time there may be usability problems for which there is no immediate fix. We recommend user participation and usability input from the earliest stages. Participative design, following sound design principles and combined with early expert usability evaluation and advice, can help guide development of more usable systems. To refine systems, to meet user needs and capabilities more fully, requires user-based evaluation and testing.

The specific goals of usability testing vary considerably, depending upon what is being tested and why. For a shrink-wrap standalone software product, the goals may be specified in terms of performance and user satisfaction for particular classes of user carrying out selected benchmark tasks. The target may be to equal or exceed the performance of competing products. For other IT products, such as control systems, the targets may relate to the safe and efficient performance of specific critical tasks by operators of the system. For organisational information systems, targets typically relate to work using the system, compared with other ways of performing that work (e.g. an existing IT or paper-based system). For walk-up-and-use information systems and interactive consumer products, the concern is both initial usability for first time users, and efficiency and satisfaction for frequent users. Other systems assist in serving customers, for example in sales, in counter transactions, and in answering enquiries. Here the criteria relate to the efficiency and satisfaction with which the system supports an organisation and its staff in serving the customer.

At NPL, we have applied methods described in this paper (or advised on their use) in evaluating products of all these types. There are numerous constraints on usability testing. The work has to fit in with tight commercial development timescales. Prototypes systems available for usability testing may have limited functionality and reliability. While some systems support tasks which are simple and frequently performed (and easy to assess), other tasks are highly complex and specialised, requiring substantial task expertise to enable their assessment. The availability of users with the required skills and other characteristics may be very limited. The specific methods employed in an evaluation must accommodate such constraints.

As awareness of the benefits of usability evaluation and measurement has increased in the UK and across Europe, so has the demand for evaluation, and for access to practical methods and tools. The development of MUSiC methods and tools, and their refinement in commercial use – combined with training, support and evaluation services – has made available a flexible set of practical methods, which complement existing participative and expert usability methods. They are being adopted by organisations with existing human factors expertise, and by others where usability has previously only been informally addressed. Their refinement continues. Current developments include establishing an accreditation scheme for their correct application, as part of quality management, and setting up support services in more European countries. As IT systems reach an ever widening number of people, usability testing and improvement have increasing significance in helping to meet user needs.

Acknowledgements

NPL's work in the development of Usability Context Analysis and the MUSiC Performance Measurement Method was jointly funded by the Commission of the European Communities and the UK Department of Trade and Industry. The author is grateful for comments on drafts of this paper to Ralph Rengger, Cathy Thomas, Rosemary Bowden and Nigel Bevan (all of NPL) who have been co-responsible for much of this work.

References

Apple Computer (1987) Human Interface Guidelines: The Apple Desktop Interface. Wokingham, England, Addison-Wesley.

Bullinger H J (ed) (1991) Proceedings of the 4th International Conference on Human Computer Interaction, Stuttgart, September 1991. Elsevier.

CEC (1990) Minimum Safety and Health Requirements for Work With Display Screen Equipment Directive (90/270/EEC) Official Journal of the European Communities No L 156, 21/6/90.

Dillon A, Sweeney S and Maguire M (1993). A Survey of Usability Engineering within the European IT Industry – Current Practice and Needs. In: JL Alty, D Diaper, and S Guest (eds.), People and Computers VIII (Proc. of the HCI'93 Conf., Loughborough UK, September 1993), Cambridge, CUP, 81-94.

Gunsthövel D, Spiegel H et al., (1993) The SANe Basic Package Handbook. ACit GmbH, Münster, Germany.

ISO (1993a) ISO 9241: Ergonomic requirements for office work with visual display terminals.

ISO (1993b) ISO DIS 9241-10: Dialogue principles.

ISO (1993c) ISO DIS 9241-11: Guidelines for specifying and measuring usability.

Karat C-M (1993) Cost-Benefit and Business Case Analysis of Usability Engineering. INTERCHI'93 and NPL Usability Forum tutorial notes.

Kirakowski J, Porteous M, Corbett M (1992) How to Use the Software Usability Measurement Inventory: the user's view of software quality. Proceedings of European Conference on Software Quality, 3-6 November 1992, Madrid.

Kirakowski J and Corbett M (1993) SUMI: the Software Usability Measurement Inventory. British Journal of Educational Technology, **24**, 210-212

Macleod M, Thomas C, Dillon A, Maguire M, Sweeney M, Maissel J, Rengger R (1993) Usability Context Analysis: A Practical Guide, Version 3.1 NPL DITC, Teddington, UK.

Macleod M and Rengger R (1993) The Development of DRUM: A Software Tool for Video-assisted Usability Evaluation. In: JL Alty et al. (eds) People and Computers VIII (Proc of HCI'93 Conf., Loughborough UK, Sept 1993). CUP, 293-309.

Macleod, M (in press) Usability in Context: Improving Quality of Use. In: G Bradley and HW Hendricks (Eds.) Human Factors in Organizational Design and Management - IV (Proceedings of the International Ergonomics Association 4th International Symposium on Human Factors in Organizational Design and Management, Stockholm, May 29 - June 1 1994). Amsterdam, Elsevier / North Holland.

Maguire M and Sweeney M (1989) System Monitoring : Garbage Generator or Basis for Comprehensive Evaluation System. In: A Sutcliffe & L Macaulay (eds) People and Computers V (Proc. of HCI'89 Conf., Nottingham UK, 5-8 September), CUP, 375 - 394.

Nielsen J and Molich R (1990) Heuristic Evaluation of User Interfaces. In Chew J C & Whiteside J (eds), CHI'90 Conf. Proceedings: Empowering People, ACM, 249-256.

Nielsen J (1992) Finding Usability Problems through Heuristic Evaluation. Proceedings of ACM CHI'92 Conference (Monterey, CA, 3-7 May), 373-380.

Nielsen J (1993) Usability Engineering. San Diego, CA, Academic Press.

Norman D A (1987) The Psychology of Everyday Things. Basic Books.

NPL (1994) NPL Usability Forum, NPL, Teddington, Middx, TW11 0LW, UK. Email information: FORUM-info@hci.npl.co.uk

Porteous M, Kirakowski J & Corbett M (1993) SUMI User Handbook. Human Factors Research Group, University College Cork, Ireland.

Ravden S J and Johnson G I (1989) Evaluating Usability of Human-Computer Interfaces: A Practical Method. Chichester, Ellis Horwood.

Rengger R, Macleod M, Bowden R, Drynan A, Blaney M (1993) MUSiC Performance Measurement Handbook, V2. National Physical Laboratory, Teddington, UK.

Sun Microsystems, Inc. (1990) Open Look™: Graphical User Interface Application Style Guidelines, Wokingham, UK, Addison Wesley.

Wasserman AS (1991) Can Research Reinvent the Corporation? - A Debate, Harvard Business Review, March-April 1991, 175.

Whiteside J, Bennett J, Holzblatt K (1988) Usability Engineering: Our Experience and Evolution. In: M Helander (ed) Handbook of Human-Computer Interaction, Elsevier, 791-817.

Wiethoff M, Arnold A G, Houwing E M (1991) The Value of Psychophysiological Measures in Human-Computer Interaction. In: Bullinger (1991).