COMPARISON OF THE MAIN METHODS FOR EVALUATING THE USABILITY OF COMPUTER SOFTWARE

A thesis presented in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy in Psychology
at Massey University.

Ronald Derek Henderson

1992

Abstract

The aim of this thesis is to examine the dominant computer software usability evaluation methods. Four evaluation methods (logged data, questionnaire, interview, and verbal protocol analysis) were used to evaluate three different business software types (spreadsheet, word processor, and database) using a between groups design, involving 148 individuals of both genders. When each evaluation method was examined individually, the results tended to support findings from previous research. Comparisons were made to examine the efficiency of each evaluation method, in terms of its ability to highlight usability problems (both between and within the evaluation strategy). Here support for the efficiency of the verbal protocol analysis method was found. The efficiency of using two evaluation methods was also examined, where it was found that no significant improvement was obtained over the verbal protocol analysis used by itself. A comparison addressing the practicality of using these methods was also conducted. It seems that each method has differing strengths and weaknesses depending on the stage of the evaluation. From these results a theory for the effectiveness of evaluation strategies is presented. Suggestions for improving the methods commonly used, are also made. The thesis concludes by discussing the software evaluation domain and its relationship to the wider evaluation context.

Acknowledgements

There have been many individuals who have significantly influenced my thoughts and behaviour throughout the development and refinement of this thesis. Each deserves my heartfelt gratitude.

Special mention must go to my family. In particular, my wife Karen, who has encouraged, supported and helped me throughout the research process. My parents, Derek and June, and my parents-in-law, Ray and Patricia, have also provided immeasurable support and encouragement throughout this period.

Special mention must also go to my supervisors, Doctors Mike Smith, John Podd and Hugo Varela. All have influenced and supported me significantly throughout the development of this thesis. My chief supervisor, Mike, has continually supported and encouraged me throughout both a significant period before undertaking this dissertation, and through the dissertation process. Mike has also been party to innumerable discussions on the nature of applied research, stimulating thought provoking enquiry on my part. John, in turn, has provided thoughtful discussion on the research process, and his insights have been both stimulating and beneficial. Deep gratitude must be expressed towards Hugo, who patiently coaxed me through the pains of learning to program in PASCAL, the virtues of log linear analysis, and trout fishing. I am deeply indebted to all; thank you.

Not least, gratitude must be expressed to each and all of the research subjects who undertook the study and provided the wealth of information that form the basis of this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

Chapter One: Introduction.

1.1. The Software Industry.

Prior to the 1970s, computers tended to be the domain of engineers, mathematicians and other scientists. Since that time there has been an astounding proliferation of computer based technology. Panko (1988) reports that not only are computers being used more today, but they are being used differently, and by a more diverse user group, points also raised by Christie and McEwan (1985). This trend was first highlighted by Benjamin (1982), who had recently finished a major statistical study of computing in one of Xerox's operating units. Benjamin started by asserting that in 1970 end user computing had been negligible. However, by 1980 it had risen to almost 40 percent of the total central processing unit cycles. Despite initial scepticism, this general trend was quickly confirmed by other organisations and this reflected an early stage of end user computing before the Personal Computer (PC) explosion.

Nowadays, rather than data processing being the single dominant application, computers are used for a multitude of applications ranging from word processing to computer aided design. This trend has been vividly emphasised by DiNucci (1985) who has reported the revenue gained worldwide from the four main categories of business software types (see Table 1.1). These figures remind us that the personal computer software business is now a multi-billion dollar enterprise.

Nickerson (1986) has provided an impressive overview of present and anticipated uses and trends in the computing industry. Nickerson stressed that,

> "the computer is a new machine, a new tool, of enormous potential. It is perhaps the most awesome tool that has yet

been developed...It is imperative that we learn to use it well and for humane ends." (Nickerson, 1986, p.5-6).

Table 1.1. Revenue Gained Worldwide From the Four Main Categories of Business Software Used on Personal Computers (from DiNucci, 1985).

TYPE OF SOFTWARE	WORLDWIDE REVENUE (MILLIONS, U.S.A.)
Word Processing	$763
Spreadsheet	$741
Integrated Software	$361
Data Management	$311

Nickerson continued by stressing that computers are now being used in many diverse areas such as: farming, manufacturing, retailing, defence, general management, education, and training and research.

This computer based technological explosion has not gone unnoticed by the social science community, with Toffler (1980) having called this expansion the "Third Wave," following the agricultural and industrial revolutions. The information based expansion has also been emphasised by Parker (1976), who clearly illustrated the changes in the United States workforce from 1860-1980 (see Figure 1.1). Eason (1988) also noted that the information based revolution is an insidious revolution, as computer based technologies have now infiltrated almost all components of the "Westernised" individual's life. Such technologies are now found from the board room to the bathroom.

1.2. Problems in the Software Industry.

The increase of interactive computer based technology has been parallelled by an increase in problems reported by individuals using this

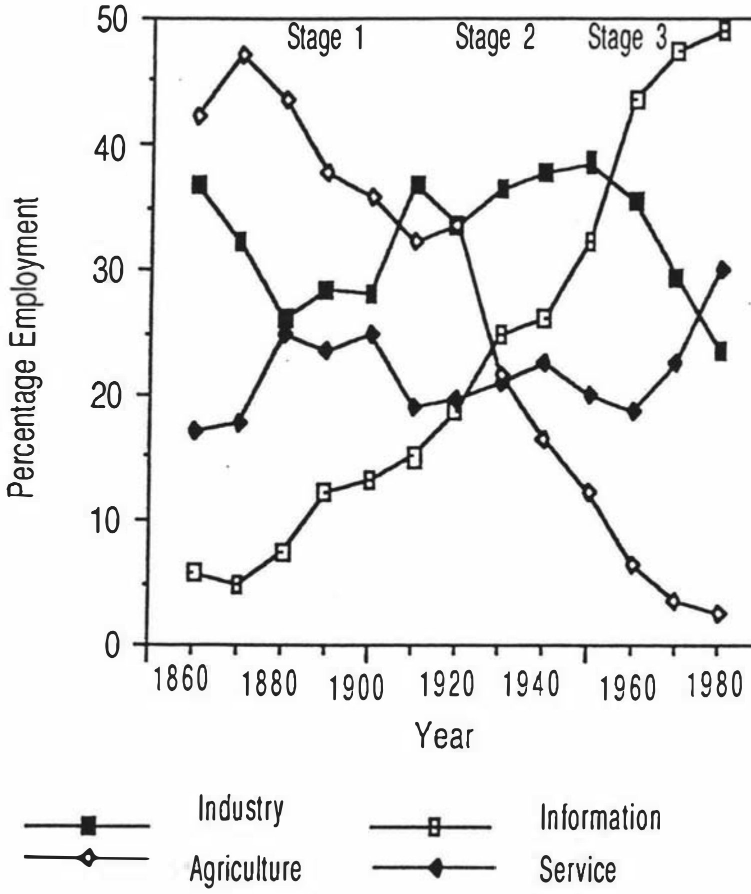


Figure 1.1. Four-Section Aggregations of the US Workforce 1860-1980 (from Parker, 1976).

technology. Salvendy (1987), when citing a survey of fifteen hundred adults, reported that 67 percent of individuals using computer technology in the work place experienced problems, 47 percent had problems in banking, 42 percent had problems with computerised home appliances, and 40 percent have had glitches involving credit cards. The problems associated with using computer based technology have also been stressed by Bjorn-Anderson (1988). Bjorn-Anderson reported that in a recent U.S.A. study it was found that 60 percent of the more than five million private citizens who owned a home computer, did not use it. Chapanis (1982) also described factor analytic work that suggests that,

> "people appear to be ambivalent about computers. They have some strong positive and some strong negative attitudes about them." (Chapanis, 1982, p.108).

These problems have reached the point such that Hartson and Hix (1989) have asserted,

> "the possibilities of this amazing machine are now limited not by its power to compute, but rather by its power to communicate with its human users." (Hartson and Hix, 1989, p.6).

Oborne (1985) also highlighted the present and future impacts of computers on organisations and society, illustrating how computer based technology not only alters individual group practices, but also the organisational structure. Furthermore, there is a wealth of evidence that computer based technology may affect both the psychological and physiological well being of the individual worker (Frese, Ulich, and Dzida, 1987; Salvendy, Sauter, and Hurrell, 1987).

The statistics and trends are clear. People are using computer based technology, and all indicators suggest that it will become more prominent in our day to day lives. Nevertheless, many individuals have problems using this technology, and this may affect both their psychological and physiological wellbeing.

1.3. The Reasons for Human Computer Interaction Problems.

Computers are designed by people. It should follow, therefore, that they should be designed with people in mind. Shackel (1986a) reported, however, that the overriding criterion in the computer industry, traditionally, has been the functionality of the system. It was perceived that all customer problems could be overcome by faster more powerful systems. Moran (1981a) also noted that the main interest of the computer science profession lies in the core of computer systems. Interest wanes as one moves away from the core towards the periphery. Traditional software development approaches and quality metrics have therefore emphasised factors such as reliability, maintainability, complexity and portability, with usability being a relatively minor component. Grimes, Ehrlich, and Vaske (1986) also reported that in the past, due to emphasis on efficient systems that could optimise costly hardware, human factors tended to be treated after everything else had been done.

Edmonds (1987) believed this was a narrow and deficient focus, drawing a parallel with architectural design by examining the notion of architectural totality. Edmonds emphasised that,

> "A building is given as a totality, and the architects aim at creation of such totalities. It is not correct to see technical realisation as the "real" result . . . The form, as well as the building task and the technical aspects are abstractions from the work of architecture, which we designate as an

> "architectural totality." The totality is a building task realised technically with style." (Edmonds, 1987, p.334).

He went on to say that,

> "Good software design should exhibit a concern with totality." (Edmonds, 1987, p.334).

Recently, there has been an evolutionary trend that has seen the primacy shift from hardware to software, and presently to the user. Shackel (1986a) reported that, within the International Business Machines Corporation (IBM), usability is now seen as important as functionality. The impetus for this shift in focus has been multidimensional, with such factors as the tipping of the hardware/software cost ratio, expansion of interactive capabilities, and the changing composition of the user group, all interacting. Similarly, Benbasat, Dexter, and Masulis (1981) suggested that, for more than 95 percent of human-computer interactions, human costs are greater than machine costs. Therefore, methods of simplifying the interface, and thus lowering human costs, would have a large impact.

It has also become apparent that in highly competitive situations where many alternate products have little difference in functional capacity, it may be the user interface that is the deciding factor, and therefore,

> "end-user satisfaction is becoming not only a productivity factor, but a decisive competitive factor" (Branscomb, 1982, p.72).

These points have also been made by Otte (1984), Shneiderman (1984), and Sutcliffe (1988) with Johnson and Anderson (1981) having described three development projects scrapped largely because of inappropriate dialogue techniques. This suggests the over-riding importance of the interface in certain situations, and adds credence to Wexelblat's (1981) comment that,

> "If a system is difficult to learn or difficult to use, people will not use it." (Wexelblat, 1981, p. 260).

Shneiderman (1984) discussed three further reasons for the increasing concern over human factors in interactive systems. First, Shneiderman argued that there is an increasing number of companies that are becoming heavily or totally dependant on interactive computing. Furthermore, the demanding conditions of many such companies make high error rates and operator confusion no longer tolerable. Shneiderman cited one example of a United States telephone company, which estimated that one second off the average time for a directory call would save the firm $24 million American dollars annually. McDonald, Stone, Liebelt, and Karat (1982) argued that in such circumstances it is more efficient to design the system to be easy to use, rather than altering human behaviour to fit the system.

Secondly, Shneiderman (1984) argued that some new systems, such as those used in the military, medical, air traffic or nuclear reactors, require extremely high performance from users. This means an absolute minimum of errors.

Finally, through the expansion of interactive computer systems, there has been a change in the user population. This new population of users is largely dominated by non-computer based professionals. Such professionals see the computer as a tool or resource, and do not wish to spend a large portion of time learning how to operate the user interface. This gives rise to the notion of a transparent system.

Furthermore, the uses for computers are becoming more complex. Gardiner (1986), for example, pointed out that communication across the human-computer interface nowadays, is not necessarily between the human and computer, but can now be between one human and any other. This may be in either single or multiple modes. Now a human(s)/ computer(s)/ human(s) interface problem can arise.

Sutcliffe (1988) remarked that poor interface design can result in increased mistakes, user frustration, and poor system performance; while Hodgson and Ruth (1985) suggested that unusable software may result in,

> "employee dissatisfaction, high staff turnover, less willingness to pursue company objectives, absenteeism and tardiness." (Hodgson and Ruth, 1985, p.16).

Cohill, Gilfoil, Pilitsis, and Carey (1988) showed that the benefits of user interface evaluation may result in increased user satisfaction, increased sales, decreased development costs, increased productivity, decreased product returns and decreased training costs. Ravden and Johnson (1989) suggested benefits of reduced training time, a reduction in support costs, a reduced need for amendments, perhaps increased sales, and greater acceptance, efficiency, and awareness.

Mantei and Teorey (1988) have pointed out that these possible benefits are moderated by such factors as the type and number of system users, the complexity of the user interface being built, and the amount and type of human factors stages that are included in the life cycle. Therefore, the human factors considerations should not be considered a panacea, but must be tempered by other considerations.

However, the evidence seems compelling. System design, with the end user in mind, benefits the software designer and producer, the organisation, and the individual. Despite this realisation there seems a gulf between the recommended user based software development techniques, and those used in practice (Gould and Lewis, 1985; Smith and Mosier, 1985; Hannigan and Herring, 1986).

1.4. The Evaluation Process.

Evaluation is a mechanism that may enhance interactive computer system design. History has recorded that the desire to assess the effectiveness of human endeavour is not new. Isaac and Michael (1981) have related that the evaluative process seems to have functioned informally since the beginning of time. This notion is supported by Mosteller (1981), who referred to one of the earliest evaluation studies that used a quasi-experimental design to examine the effectiveness of a vegetarian diet compared with a diet containing meat. The results of this study conducted by Daniel, Shadrach, Meshach, and Abednego are recorded in the Old Testament of the Bible.

Riecken (1977) suggested that the modern history of evaluation research can be traced back to 1950 when post-war institutions were entering a period of critical self-examination. Concomitantly, Isaac and Michael (1981) suggested that the more formal version of evaluation research coincided with the advent of the computer to,

> "give rise in the 1950s to the 'man-machine systems' movement, and currently to the 'systems 'approach." (Isaac and Michael, 1981, p.2).

Suchman (1967) has provided an expanded overview of the evaluation mechanism and enumerates seven steps to the evaluation process (see Table 1.2).

Table 1.2. Seven Step Evaluation Process (from Suchman, 1967).

1. Identification of goals to be evaluated.
2. Development of measurable criteria specifically related to these goals or objectives.
3. Analysis of the problems with which the activity must cope.
4. Description and standardisation of the intervention procedure.
5. Measurement of degree of change that take's place by means of setting up a controlled situation to determine the extent to which these objectives and any negative side effects are achieved.
6. Determination of whether the observed change is due to the intervention activity or to some other cause.
7. Some indication of the durability of the effects.

Anderson, Friedman, and Murphy (1977) provided a more succinct account of the evaluation process through an overview of definitions of evaluation, ranging in complexity and comprehensiveness. They asserted that, taken as a whole, all definitions describe an approach to evaluation that incorporates three essential components: the formation of criteria, assessment of the attainment of criteria, and the utilisation of results. It would appear, therefore, that the evaluation process is essentially linear in nature. However, the process, although appearing rudimentary, is more complex than initial impressions would suggest.

Evaluations generally fall into the two stages of formative and summative evaluation (Scriven, 1972). Edwards, Guttentag, and Snapper (1975) reported that many social programmes often need continuous feedback to permit wise programme management and adaptation to either correct errors or to adapt to changing circumstances. This type of evaluation feedback is known as formative evaluation, whereas summative evaluation describes a final verdict on the programme. These ideas highlight the fact that evaluation is not a static, one-off, linear process, but a fluid and open process, incorporating both feedback and outcome measures.

Wortman (1975) has produced an "ideal" model that embroiders these concepts into a working whole (see Figure 1.2). The model emphasises the experimental design characteristic of the scientific hypothesis-testing approach, including internal and external validity, randomisation, and the use of control groups. Wortman emphasised that the evaluation process occurs both in an applied setting and a political setting, and such considerations will impact on the evaluation process. Therefore, organisational components, theoretical components, and the evaluation process itself must be considered. That is, evaluation researchers have to not only meet required research standards, but must do this within a set context. Accordingly, the Wortman model incorporates both formative and summative components, while also addressing validity issues.

This basic model of evaluation seems not only to apply to the social research setting, but to most evaluative settings. Meister (1986) outlines an ergonomic test plan for conducting an evaluation that is remarkably similar (see Table 1.3).

Furthermore, Hamblin (1974), in the training context, remarked that one of the purposes of evaluation is the control of training by a process of collecting, analysing and evaluating information, leading to decision-making and action. The cyclic process also underlies the work of Rackman, Honey, and Colbert (1971) who described the purpose of evaluation as that of creating a feedback loop, a self correcting system, a

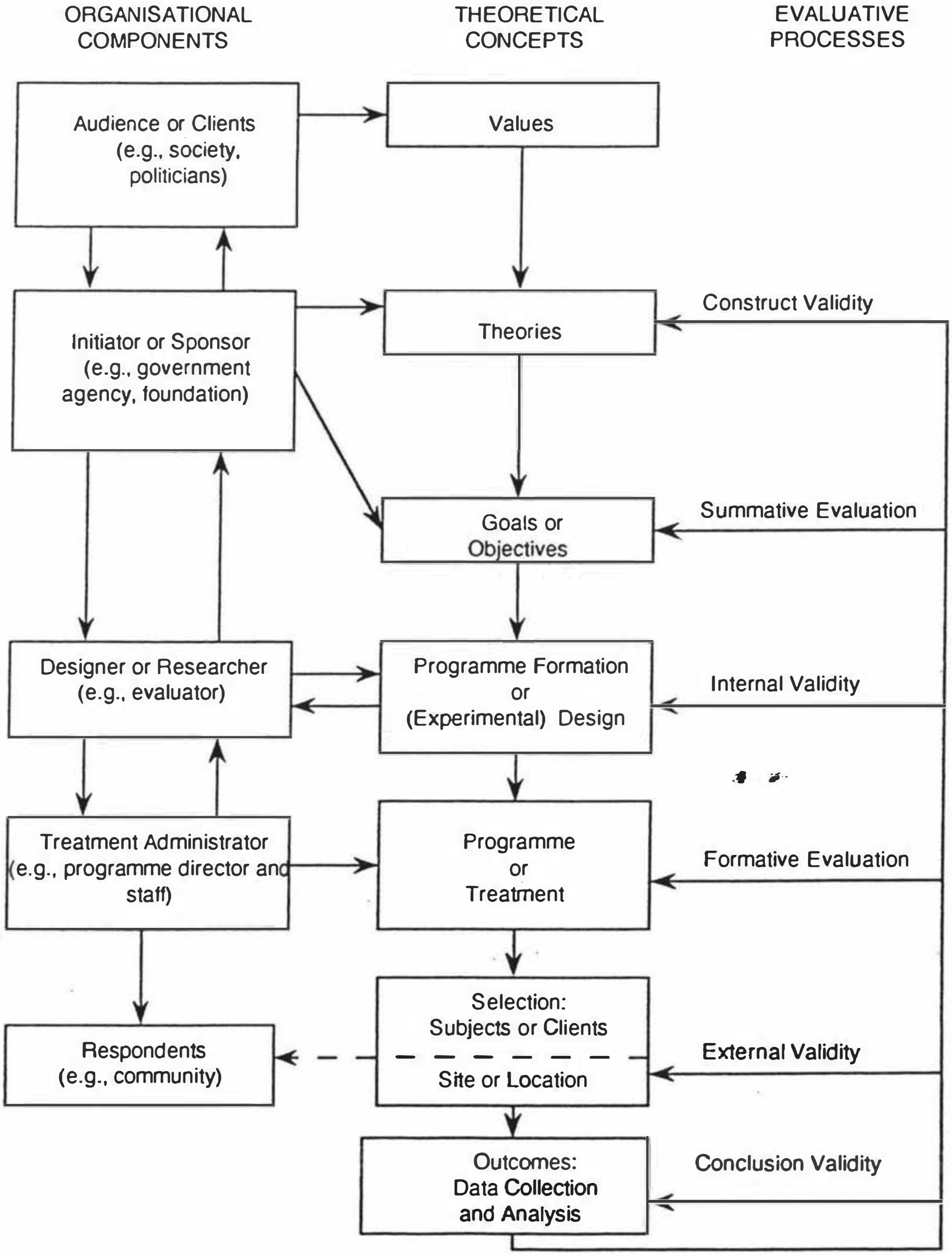

ORGANISATIONAL COMPONENTS
THEORETICAL CONCEPTS
EVALUATIVE PROCESSES
Audience or Clients (e.g., society, politicians)
Values
Initiator or Sponsor (e.g., government agency, foundation)
Theories
Construct Validity
Goals or Objectives
Summative Evaluation
Designer or Researcher (e.g., evaluator)
Programme Formation or (Experimental) Design
Internal Validity
Treatment Administrator (e.g., programme director and staff)
Programme or Treatment
Formative Evaluation
Respondents (e.g., community)
Selection: Subjects or Clients
Site or Location
External Validity
Outcomes: Data Collection and Analysis
Conclusion Validity

Figure 1. 2. A Model of Evaluation Research (from Wortman,1975).

Table 1.3. Test Plan for Conducting Ergonomic Evaluations (from Meister, 1986).

1. Analyse system to determine those variables that probably affect system and operator performances. Examine predecessor system, talk to subject matter experts, review documents, test reports, etc.
2. Specify precisely why the test is to be performed and what its outputs will be.
3. Determine criteria, measures and standards.
4. Develop experimental design (if appropriate), statistical analysis and data collection methodology.
5. Select subjects (if it is relevant and possible to do so).
6. Have test plan reviewed and approved by management.
7. Try out data collection procedures.
8. Revise test plan and data collection procedures.
9. Commence data collection.
10. Analyse data.
11. Make recommendations resulting from test data.
12. Write interim and final test reports.

notion also described by Lea (1988), concerning software evaluation in the computing industry. Such models also seem strikingly similarto the concepts of action research (Cohen & Manion, 1980), that are prominent in the educational setting.

It becomes apparent that this generic process is fundamental to evaluation studies, whether they are educational, social, psychological or industrial. In all cases criteria must be established, assessment of the criteria must be undertaken, and results must be reported and presumably acted upon. During initial development, the evaluation may be in a mainly formative mode, and therefore, internal feedback is paramount. At some stage, however, the outcome must be assessed and reported. This report should then be utilised in some way. Furthermore, it should also be noted that evaluation research is set in both applied and political settings. Ideal circumstances do not exist; funding, time, and resources may be limited, but a good evaluation is still required.

Chapter Two: Developing and Evaluating the Human-Computer Interface.

2.1. User Based Development and Evaluation Strategies.

When developing computer systems, Gould and Lewis (1985) originally advocated four critical concepts. They were, an early focus on users, empirical measurement to assess learnability, usability of the human computer interface, and iterative design. Through refinement and development, Gould (1987,1988) also stressed integrated design. This resulted in a focus on design concepts which encompasses users, integrated design, user testing, and iterative design. Gould argued that each factor must be considered as an integral component of the design process. Although these concepts are not as immutable as implied by Gould (see Olson and Ives, 1981; Maguire, 1982), it appears as if they are broad, robust, assertions. Unfortunately, when examining the design process it has been found that, in general, these steps are not followed. A rather informal approach to the design of the user interface is more common (Farooq and Dominick, 1988).

The reasons for such an approach tend to be multifaceted; for example, Landauer (1988) suggested that system designers may fall into the trap that he coins the "egocentric intuition fallacy." Landauer intimated that software designers are often seduced by the compelling illusion that one knows the determinants of one's own behaviour and satisfaction. Consequently, we tend to overestimate the degree to which what is true for us will be true for others. This results in designers greatly underestimating the variability in performance and preference. The resulting outcome Landauer suggested made it,

> "hard for programmers and system development managers to appreciate the need for more rigorous methods for ensuring usability." (Landauer, 1988, p.906).

Bjorn-Anderson (1988) also noted that many problems, in relation to the introduction of office automation, are related to the fact that experts are designing something for others to use. Sanders and McCormick (1987) have also pointed out that human factors are not just common sense, and that a lay approach may not result in the desired benefits anticipated. Edmonds (1987) suggested that,

> "The pain felt by ergonomists when they read the advertisements or are given a demonstration of the latest "ergonomic" product is surely great. The problem is not simply the misuse of the word, it is also the extreme simplification of the notion of good design." (Edmonds, 1987, p.334).

Landauer (1988) believed that designers may be amazed by how their systems are used. This may be totally different to how the interface of their "baby" was expected to be used, a point also made by Karat (1988). Landauer suggested that it was impossible for system developers to adequately evaluate their own software, because after hundreds of hours of use and development, they have a "tainted" view and cannot place themselves in the position of a naive, or new, user of the product. These factors suggest that the individuals developing the software are not the best people to evaluate it, and reinforce the need for user based testing (Gould, 1987, 1988).

Additionally, Spinas (1989) has presented the results of 32 case studies designed to examine user participation within the development process. Spinas concluded that there is,

> "strong evidence that users are not only interested in

> participating in software development but are also able to contribute substantiallyto the improvement of a system." (Spinas, 1989, p.205).

Nevertheless, problems did occur. In particular, it was found that user participation was started too late in the design process. At times there was also a decrease in motivation within large projects of long duration, due to a lack of feedback. Designers also tended to underestimate user contributions. Moreover, communication also became a problem, particularly because of different language usage between users and designers. Users also had difficulty understanding system specifications or pencil and paper versions of the system. Despite these problems, the results reinforced the idea of a continual and early focus on the user population.

2.2. Models for the Development of Usable Software.

In an attempt to clarify and synthesise interactive design concepts Williges, Williges, and Elkerton (1987) reviewed and organised various software design methods into a three stage iterative software design process. This encompassed the initial design, formative evaluation, and summative evaluation, and is shown in Figure 2.1.

Stage one of the Williges et al. (1987) model encompassed the initial design, the planning, specification and basic configuration of the software interface. This is the stage that will result in an initial prototype of the interface, that should then be iteratively tested. They reported that it must be carefully constructed, because a good initial design will result in fewer redesign iterations. Approaches that may help in the design process at this stage of development include theoretical models, such as the Command Language Grammar (CLG) (Moran, 1981b), Backus Naur Form (BNF) (Reisner, 1984), Goals Operations Methods Selection

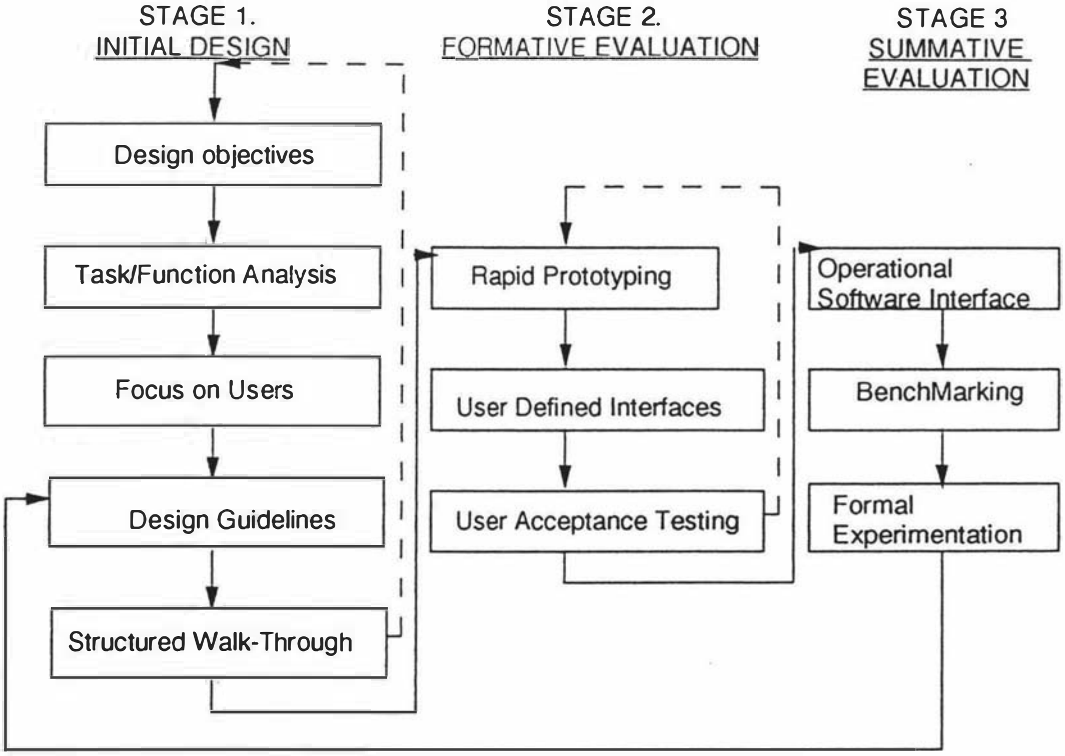


Figure 2.1. Iterative Software Design Process (from Williges, Williges, and Elkerton, 1987).

(GOMS) (Card, Moran, and Newell, 1983), and Production System Analysis (PSA) (Kieras & Polson, 1985).

Proponents of such models suggested that they constrain design at an early stage and therefore, block expensive usability failures and consequently reduce the need for usability testing. For example, Card et al. (1983) claimed that their model is sufficiently accurate, robust and flexible to be applied to practical design and evaluation (see Roberts and Moran (1983) for an implementation of this type of model). In contrast, Allen and Scerbo (1983) suggested that such claims are premature and provide empirical and theoretical evidence suggesting that the model still needs further development and refinement. Christie and Gardiner (1990) also pointed out that the simplifications needed to be made to the GOMS model may restrict the value of the approach in practical situations. Furthermore, such systems tend to model that of an expert user and may be limited when dealing with the new or naive user. Dzida (1984) also has shared a concern for the application of the model, arguing that it may be applied using misleading measurement concepts, resulting in inappropriate development criteria being used and strived for.

Design principles and guidelines have also emerged that can be used to guide the developer and eliminate obvious interface problems (Brown, 1986; Smith and Mosier, 1986; Frese, 1987). Gardiner and Christie (1987) suggested that a distinction can be drawn between principles, guidelines and standards. Gardiner and Christie have related that design principles are generally taken to be more abstract recommendations, often posed in conceptual terms rather than a design action. Examples include Shneiderman's (1984) concept of direct manipulation, and Thimbleby's (1984) concept of generative principles. In contrast, Gardiner and Christie (1987) proposed that compared to guidelines, principles require more interpretation within the context of a particular problem to support decision making. Standards in turn are intended to be as unambiguous as possible.

Problems with such approaches do exist, however. Smith and Mosier (1985), in a study involving 130 professional software developers, found that although guidelines were useful in establishing requirements before design, as a decision aid, and for evaluation purposes, users reported long search times to find applicable guidelines. This was on average 14 minutes. Furthermore, when the appropriate guideline was located, they were sometimes irrelevant, too general, and occasionally too specific. Also, the guidelines were perceived as not sufficiently helpful in supporting decisions about new technology. Smith (1986) also argued that it is too early to develop standards, as they might lead to inflexible use.

Landauer (1988) made the point that although there are competently compiled guidelines they are,

> "mostly based on good current practice or expert opinion" (Landauer, 1988, p.915).

Furthermore, such guidelines can appear contradictory, resulting in confusion and debate between the developer and evaluator (Maguire, 1982; Gardiner and Christie, 1987). Problems such as these have caused Gardiner and Christie to suggest that guidelines may be better used for physical aspects and simple presentation issues of the interface, rather than the cognitive aspects of interaction. They concluded that,

> "Guidelines are only one of the tools in the human factors tool kit and, just as one would not build a house using only one tool, so too interfaces need to be built through the combined use of different tools. This means that guidelines may have their own important role to play in specific facets

> of the design process, and should be used within that application boundary." (Gardiner and Christie, 1987, p.41-42).

In essence, the designer does not have to present a user interface representing totally random usability attributes.

Nevertheless, as highlighted by Carroll (1989),

> "Direct empirical measurement is still the only adequate means of assessing the usability of software techniques and artifacts" (Carroll, 1989, p.49-50).

The reasons appear multifaceted and have been summarised adequately by Brooks (1987) who pointed out that inherently, everything interacts with everything else.

During the second, formative evaluation stage of the Williges et al. (1987) model, the interface that was specified in stage one is implemented, evaluated, and redesigned in an iterative fashion. This process should continue until the design objectives are satisfactorily achieved. Such concepts as object oriented programming and rapid prototyping can be used as a means to carry out iterative design during this stage of the process. It is during the third stage that the final summative evaluation is conducted. Here Williges et al. (1987) reported that the evaluation may consist of comparing the final design with other design alternatives and / or previous versions of the interface. Also, comparisons may be made with commercially comparable alternatives.

Christie and Gardiner (1990) have provided an overview of the design process highlighting the nature of evaluation as experienced in commercial software design. Christie and Gardiner suggested that,

> "Each company has its own way of doing things, and to some extent each product development project has its own unique requirements." (Christie and Gardiner, 1990, p.282).

Despite this, five main stages to the design process can be identified,

1. Pre-design information gathering.
2. Design.
3. Design review.
4. Implementation.
5. Fine tuning.

Although this process seems strikingly similar to that advocated by Williges et al. (1987), Christie and Gardiner (1990) placed more emphasis on the commercial implementation of evaluation, stressing that it will be limited by budget restrictions, by the time scale, and by the relative lack of awareness of the interface as a specialist area, notions supported by Lea (1988). Shneiderman (1988), in turn, highlighted "three pillars" of user interface development. These consisted of guideline documents, user interface management systems and usability laboratories, and iterative testing. Sutcliffe (1988), on the other hand, highlighted user-participative design, user-centred design, and iterative design.

Several dominant themes for interface design emerge. Guidelines and principles can be used in the initial specification of the user interface. These may be implemented by user interface management techniques, such as rapid prototyping and object oriented principles. The interface may even be evaluated by using formal theoretical models, such as GOMS (Card et al. 1983), and then undertaking iterative user testing. The politics of the organisation will also moderate this evaluation and iterative process. All this is remarkably similar to the original evaluation stages suggested by Wortman (1975).

2.3. The Aims and the Individuals Included in the Evaluation.

Draper and Norman (1985) suggested that there are,

> "two quite different broad aims for an interface: Achieving speed and convenience of use (power) for the practised user and achieving ease of learning and use." (Draper and Norman, 1985, p.253).

In turn, Root and Draper (1983) suggested that there are two kinds of concerns that a software developer may have,

> "to identify problem areas that may not even have been suspected before, and to get more information on existing hypotheses about problem areas." (Root and Draper, 1983, p.84).

This has been expanded by Karat (1988) who suggested that the reasons for evaluating a system can be divided into several broad classes. These include assisting in design decisions, measuring quality, evaluation to select between two or more alternatives, the intention of understanding how well a system design works, and to decide whether a system meets some acceptable criteria. Lea (1988) also reported that three broad categories of objectives can be identified:

1. The assessment of the capabilities of the design.
2. The assessment of the impacts of design decisions.
3. The diagnosis of problems with the design.

Yoder, McCracken, and Akscyn (1985) reported that at differing stages of the design various individuals will require information on how the system is operating and being used,

> ". System developers need to know what problems and inefficiencies are plaguing the system so they can improve the design and implementation.
>
> . Managers who are introducing the system into their organisation need to know how people feel about the system and how it is affecting their job performance and social environment.
>
> . System evaluators are typically concerned with job performance may also be evaluating the system's potential for use at other sites or in other applications.
>
> . Researchers studying human-computer interaction need detailed information about the user's interaction with the system." (Yoder et al., 1985, p.907).

The types of evaluation may also vary. Specifically, Howard and Murray (1987) asserted that the evaluations may be subject based, expert based, theory based, user based, and market based.

As a means to implement such evaluation strategies, software developers are beginning to employ the analytic skills of ergonomists and psychologists (Richardson, 1987, Shneiderman, 1988). The utilisation of ergonomists and psychologists within the design team would seem a natural phenomenon. Professionals in these fields have been examining human-machine interaction issues and the evaluative process for some time. Chapanis (1976), for example, reported that the primary aims of ergonomics, or human factors engineering as it is known in North America, is the discovery and application of information about human behaviour in relation to machines, tools, jobs, and work environments. As far back as 1955 ergonomists have been asserting that if the point of contact between the product and the user becomes a point of friction, then the designer has failed. Conversely, if people are made more comfortable, more eager to purchase, more efficient - or just happier - the designer has succeeded (Dreyfuss, 1955).

It seems prudent to draw on the knowledge and experience of ergonomists and psychologists and direct some of their expertise specifically towards problems associated with human-computer interaction. This is happening with a rapid expansion and redirection of resources to research on human-computer interaction. Several new academic journals have been developed and special interest research groups established, leading Shneiderman (1988) to suggest that,

> " While it may be too bold to declare a "golden age of ergonomics" it seems clear that human factors issues and the politics of "user-friendly" are fashionable topics in professional journals and popular magazines."
> (Shneiderman, 1988, p.699).

The role of human-computer interaction specialists may vary depending on the design stage of the software and the individuals requiring the information. Eberts (1987) has suggested that the ergonomist can play many roles within the design process and can contribute at all phases of the design of the human-computer interface.

Carroll (1989) also reported that the role of human factors specialists may vary within each organisation and that in the past the basic assumption held was that,

> "psychology operates <u>outside</u> the development process, outside even the research prototyping process. They assume that the role of psychologists in human computer

> interaction is to offer commentary." (Carroll, 1989, p. 62).

However, this view is changing, with Carroll (1989) having advocated that human factors professionals should be placed directly into development groups and that they should manage the developers. Usability consultants from outside the organisation should also be used (Carroll, 1989). The incorporation of the usability consultant into the development process has given rise the notion of usability innovated invention where,

> "usability is seen as connecting the invention of HCI (Human Computer Interaction) artifacts to user needs no less essentially than nerves connect organs and muscle tissues to sensory and motor brain centres." (Carroll, 1989, p.61).

Although some software designers are sceptical of the importance of human factors evaluations, research by Marshall, Mcmanus, and Prail (1990) suggested that improvements and suggestions made by human factors specialists should be taken seriously. They were in the position of having examined a commercial product that had undergone several human factors evaluations. It was then released without modification. Marshall et al. observed that,

> "Circumstances dictated that no remedial action was taken, so it was possible to track these potential usability defects to customer sites, where it was found that the most important problems did indeed occur." (Marshall et al., 1990, p.243).

This adds credence to the validity of the human factors evaluation in the software development setting.

2.4. The Lack of Criterion Development in Software Evaluation Methods[1].

Although much attention is being directed at the development of user interface management tools that enable models of an interface to be quickly developed and tested, less energy appears to have been directed towards the development and validation of evaluation tools to test the various prototypes. There is little research into the efficacy of the various software evaluation techniques. This is a serious problem because iterative empirical testing seems so prominent in the software evaluation design philosophies. Yet, without empirical data on the appropriateness of the evaluation methods, the construct validity of the software evaluation design philosophy must be in doubt.

This is the case because the quality of refinements made to software will be directly related to the quality of the information elicited during an evaluation. Poor evaluation information that does not highlight potential problem areas, or misdirects the attention of the developers, is of little use for refining the user interface. Such poor evaluation information may even allow poor design elements to go undetected.

To study the usability evaluation methods the construct of usability must be understood. From this point, operationalisations can be examined, and measures used to test usability attributes can be considered. The usability construct will be addressed in the next chapter.

[1] Henceforth the term evaluation methods will be taken to refer specifically to user based evaluation and it therefore, excludes theoretically based approaches.

Chapter Three: The Usability Construct and the Problems of Operational Definitions and Measures.

3.1. The Usability Construct.

The usable system is a popular topic of research in recent literature. There has been a proliferation of design guidelines and suggestions, which have ranged from the highly technical specifications and theoretical guidelines, to guesswork.

The idea of the usable system is not new, because industrial designers have been concerned with the usability of products for some time (Dreyfuss, 1955). As far as software is concerned, ideal design concepts are still in the early stages of development. This is reflected by the gulf between ideal software development techniques and those used in practice (Gould and Lewis, 1985; Smith and Mosier, 1985; Hannigan and Herring, 1986).

A construct can be defined as,

> "A concept based on relationships between empirically verifiable and measurable events or processes." (Goldenson, 1984, p.175).

Goodwin (1987) stressed that the usability construct is not easily defined and may change depending on the context and user groups. Therefore, definitions may range from the technically dominated approach of Barnard, Hammond, Morton and Long (1981) of,

> "To be truly usable a system must be compatible not only with the characteristics of human perception and action, but, and most critically, with a user's cognitive skills in

> communication, understanding, and problem solving."
> (Barnard et al., 1981, p. 229),

to the more popular approach of Meads (1985) of,

> "A friendly system has three important aspects. It is cooperative, preventative, and conductive." (Meads, 1985, p. 97).

Richardson (1987) further stressed that the usability construct relates to the extent to which the system matches the user's characteristics, and skills for the task concerned. This latter definition implies that the construct is similar to the ergonomic concept of compatibility, which implies that things do what they are expected to do. Moran (1981a) noted that there is no one dimension of goodness of behaviour, but behaviour is multi-dimensional and may encompass such facets as, functionality, learning time, errors, quality, robustness, and acceptability. Furthermore, trade-offs between these dimensions usually need to be made.

Dzida, Herda, and Itzfeldt (1978) reported the results of a factor analytic study examining the dimensionality of the usability construct. They identified seven orthogonal factors (self-descriptiveness, user control, ease of learning, problem adequate usability, correspondence with user expectations, flexibility in task handling and fault tolerance), which accounted for 44 percent of the total variance. In contrast, Shackel (1986b) distinguished four components in his operational definition of usability: effectiveness, learnability, flexibility, and attitude. Foley, Wallace, and Chan (1984) on the other hand highlighted what they term "ergonomic quality" which seems to equate to usability. They identified three primary criteria for ergonomic quality: the time any user must spend to accomplish a particular task, the accuracy with which the user can accomplish the project, and the pleasure the user derives from the process. They then went on to identify secondary criteria, which elaborate on these primary criteria.

Frese (1987) highlighted the difference between the broad construct of usability and the more narrow construct of user friendliness. A definition of user-friendly has also been proposed by Edmunds (1985), who defined and helped to operationalise the process. Edmunds stated that user-friendly is,

> "A term that describes computer hardware or software products or a computer system that is easy for a person (normally a "non-technician") to understand and use. Various techniques are utilised (particularly with on-line systems) to make a computer system more "user-friendly." Examples of such techniques are on-line help facilities, tutorials, screen formats that use "plain english" as opposed to "computerise," and well-written users' manuals." (Edmunds, 1985, p.468).

This later definition is more explicit about the attributes of the user friendly system, but is still rather vague concerning the precise dimensionality of the construct. It does, however, highlight its nebulous nature. The construct itself relates not only to the software presented, but also to such factors as support, functions, and system performance. This was also examined by Gould (1987), who presented the components of usability (see Table 3.1) and highlighted how many factors integrate to compose the construct.

The concepts of functionality and acceptability seem an integral component of the usability construct. Richardson (1987) noted that they can sometimes take precedence over the usability criterion.

Table 3.1. Suggested Components of Usability (from Gould, 1987).

- System performance.
  - Reliability.
  - Responsiveness.
- System functions.
- User interface.
  - Organisation.
  - Input/output hardware.
  - For end-users.
  - For other users.
- Reading materials.
  - End-user groups.
  - Support groups.
- language translation.
  - Reading materials.
  - User interface.
- Outreach program.
  - End-user training.
  - On-line help system.
  - Hot-lines.
- Ability for customers to modify and extend.
- Installation.
  - packaging and unpacking.
  - Intra field maintenance and serviceability.
- Advertising.
  - Motivating customers to buy.
  - Motivating user to use.
- Support-group users.
  - Marketing people.
  - Trainers.
  - Operators.
  - Maintenance workers.

Eason (1988) added the notion of organisational acceptability, which is defined as the extent to which the system is able to support organisational objectives. Notions of this kind have also been highlighted by Rushinek and Rushinek (1986), who found in their study that system response time was the main predictor of satisfaction with a system. Goodwin (1987) also noted the integral relationship between functionality and usability by stating that,

> "it is important that a system provide the functions that a user needs to accomplish a task or set of tasks. However, it is a mistake to suppose that design features intended to enhance usability are niceties to be provided at the designers convenience . . . There is increasing evidence that the effective functioning of a system depends on its usability." (Goodwin, 1987, p.229).

Goodwin (1987) also noted that although these definitions may be quite valid they do not direct software developers towards specific guidelines. This means that they do not help to operationalise the construct to an adequate degree.

3.2. Evaluation Relevance, Deficiency, Contamination, and Redundancy.

To ensure the acceptable development of the usability construct, strong operational definitions must be present. Operational definitions provide the link between the construct and the measurement process, which at times may be tenuous. In particular, the concepts of relevancy, contamination, and deficiency must be addressed. These concepts and relationships may be illustrated by using an approach outlined by Blum and Naylor (1968).

Start by conceptualising the ultimate usability evaluation. This is an evaluation that encompasses all the components of usability. In Figure

3.1 this is represented by space "A." the actual operational definition of this concept may be, relevant, deficient or contaminated. Space "B" represents the actual evaluation. A portion of the space "B" overlaps space "A," and this portion is known as relevance. The actual evaluation is only encompassing a portion of the entire construct, and is therefore deficient. When considering this relationship contamination also needs to be addressed. Contamination occurs when a construct that is not relevant is measured. This model is useful because it highlights issues that impact on the effectiveness of an evaluation strategy.

Based on this model, the refinements made to software will be directly dependent on the quality of the information elicited from the evaluation. To the extent that the evaluation is relevant, deficient, or contaminated, so too will any refinements made on the basis of the evaluation be relevant, deficient or contaminated.

Furthermore, when multiple evaluation strategies are used, the problem of redundancy of information also becomes important. Figure 3.2 illustrates this problem. It should be noted that in Figure 3.2, the two evaluation methods overlap with the ultimate evaluation. The extent of this overlap may vary. For example, in Figure 3.2 there is some overlap between the two evaluation methods and the ultimate evaluation. They appear to be tapping the same portion of the ultimate evaluation.

The evaluation strategy taps more of the ultimate evaluation using the second method, but the usefulness of this second method is moderated by the overlap it has with the first method. If there is low overlap between the methods, and high overlap with the ultimate evaluation, useful extra information will be gained. This extra information will outweigh the extra cost, time, and effort associated with using the second method. On the other hand, if there is high overlap between each evaluation method, then the extra information gained may not warrant the

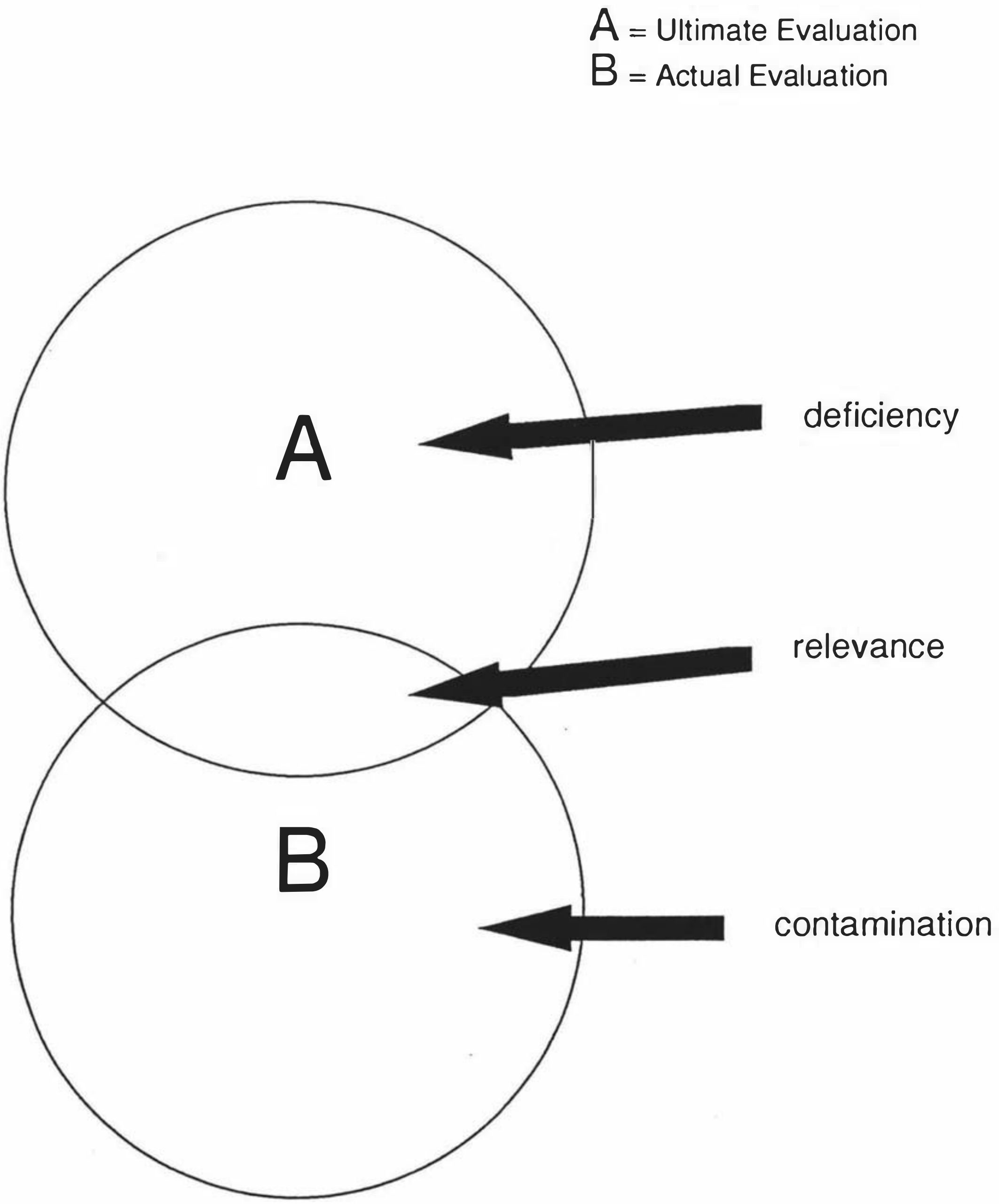


Figure 3.1. The Deficient and Contaminated Evaluation.

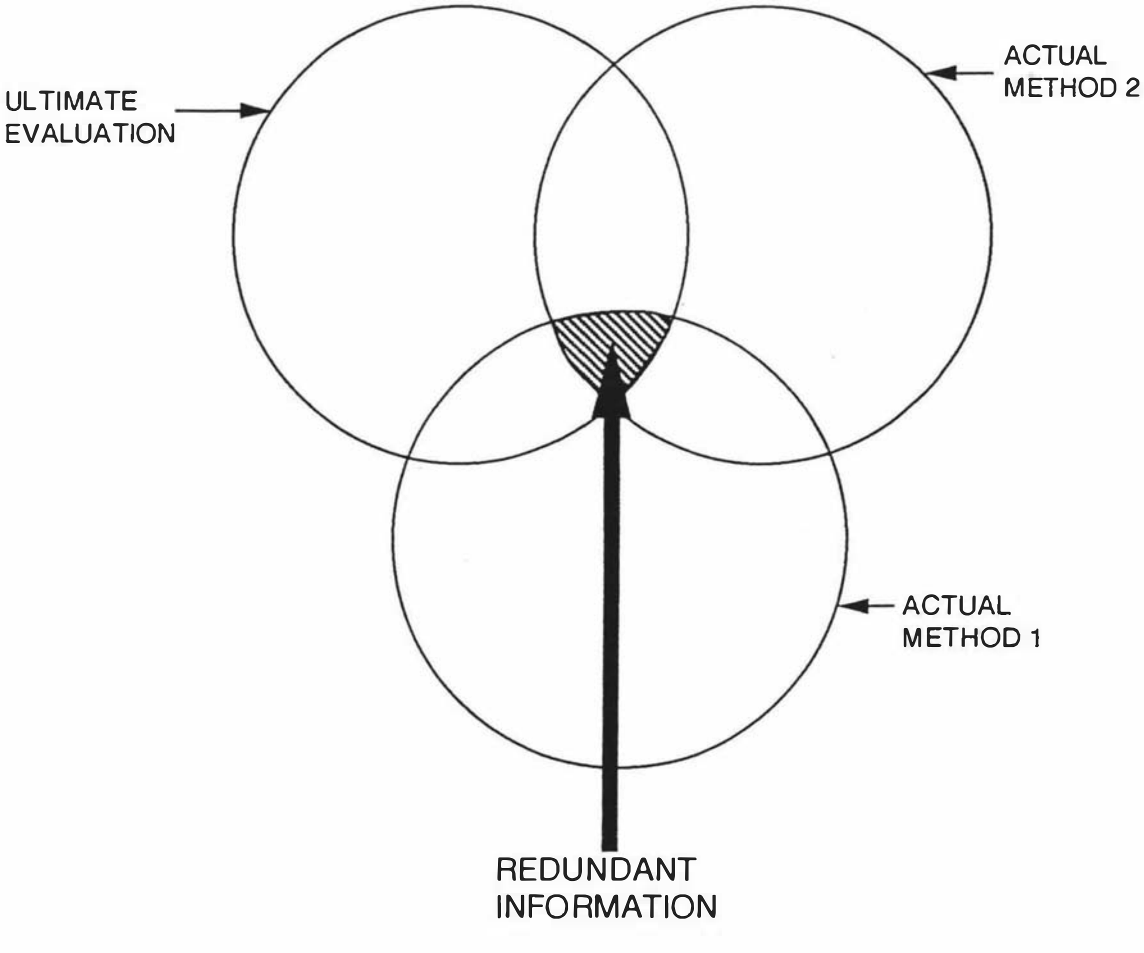


Figure 3.2. Potential Relationships Between the Ultimate Evaluation and Two Actual Evaluations (adapted from Blum and Naylor, 1968).

extra expense, time, and effort of using two evaluation methods. It is therefore vital to examine the information gained by each evaluation method before suggesting the use of composite evaluation strategies.

3.3. Operational Definitions.

Penniman and Dominick (1980) having conducted a review of the literature stated that the interview, questionnaire, logged data, secondary data analysis, and controlled experiments are methods that have been used to assess usability. Sweeney and Dillon (1987) reported that developers may use standard performance measures, interactive error analysis, subjective ratings/questionnaires, verbal protocol analysis, and system monitoring. Sutcliffe (1988) adopted a similar view suggesting that, system logging, video recording, direct observation, protocol analysis and questionnaires are options. Yamagishi and Azuma (1987) stated that logged data, the interview, protocol analysis, and evaluation questionnaires are the dominant methods. Karat (1988) suggested that the questionnaire, verbal report, controlled experiments, design reviews, formal analysis, and production systems analysis are options. Furthermore, each option will produce differing information and therefore, will be appropriate at differing stages of the development cycle. After examining the software evaluation methods Meister (1986) asserted that,

> "although automated methods of recording computer events and operator actions can perhaps be considered a distinctive feature. . when the methods used to evaluate computer systems and software are considered in their generic form. . these methods are essentially the same as those used in non-computer systems and software situations." (Meister, 1986, p. 268).

Maguire and Sweeney (1989) have put software evaluation methods into a coherent form through a taxonomy of evaluation methods. The taxonomy was based upon, whether the method was user based, theory based, or expert based, the intrusiveness of the data collection process, and the type of data available (see Table 3.2). Maguire and Sweeney also described the relationships between the evaluation metric and the data capture method (see Table 3.3).

Although these are the dominant methods other methods have also been advocated. In particular, the psychophysical method (Grudin and Maclean, 1985), the critical incident approach (del Galdo, Williges, Williges and Wixon, 1987), the impact analysis table (Gilb, 1985), and the randomly sampled self report method (Tynan, 1985), to name but a few. Furthermore, syntheses of the base methods have also been advanced (Neal and Simons, 1984a, 1984b; Kopp, 1988; Morris, Theaker, Phillips and Love, 1988; and Theaker, Phillips, Frost and Love, 1989), which seem to be based around the playback method, incorporating elements of logged data, video analysis, protocol analysis and the use of questionnaires.

Although the Maguire and Sweeney (1989) taxonomy highlights whether the method will elicit subjective or objective data it does not tell the researcher what to measure. However, many measures have been suggested (see Table 3.4).

3.4. Analysis of Measures.

In the past, evaluations of software usability data have traditionally been of a univariate and bivariate nature (Shneiderman, 1987). This has perhaps been a function of the type of data elicited and the knowledge of those performing such usability evaluations. However, the development of the usability construct has been parallelled by a development in

Table 3.2. Taxonomy of Evaluation Methods (from Maguire and Sweeney, 1989).

GENERAL APPROACH	TYPE OF DATA CAPTURED	SPECIFIC METHOD	
User-based evaluation	Performance (based on user interaction) (Objective measure)	Live observation (manual) -observation journal	(intrusive)
		Video and audio recording of user interaction	(intrusive)
		-recording of interaction data	(non intrusive)
		System monitoring (automatic)	
	Behavioural (Objective measure)	Recording of non-verbal gestures	(intrusive)
	Psychophysical (Objective measure)	Galvanic skin response Heart rate Eye movement	(intrusive)
	Affect or Attitudinal (users' attitudes and opinions) (Subjective measure)	Replay of interaction and post hoc comment Follow-up interview Rating scale Questionnaire Group Survey	(non intrusive)
	Cognitive (users' understanding and knowledge of system) (Objective measure)	Verbal protocal anaysis (VPA) Interaction replay & post hoc comment	(intrusive)
		Content analysis	(non intrusive)
		Comprehension questionnaire	(non intrusive)
Theory based	Performance (predictions of usage) (Objective measure)	Formal modelling -grammatic techniques e.g. CLG, GOMS -diagrammatic techniques e.g. CCT	
Expert-based evaluation	Performance (expert appraisal of system perfomance) (Objective measure)	Comparison checklist with: -guidelines and standards -design criteria -general fitness of purpose System 'walk throughs'	
	Expert opinion (Subjective measure)	Rating scales	

Table 3.3. Relationships Between Evaluation Metrics and Data Capture Methods (from Maguire and Sweeney, 1989).

Ticks √ indicate possible data capture methods to support each metric assessment.

EVALUATION DATA TYPE	RELATED METRICS	METHODS OF DATA CAPTURE										
		Live observation of user during interaction	Video & audio recording of user interaction	System monitoring of user interaction	Session replay and recording of users' post hoc comments	Audio recording of user verbal protocols	Psycho-physical recording equipment	User Question-naire interview survey	Rating scale	Formal modeling	Compre-hension questions	Check-list inventory walk-through
PERFORMANCE (based on user interaction)	Task times, error rates, Coverage of facilities, % task completion, Duration of time in HELP, Continuance of usage	√	√	√								
BEHAVIOURAL	Non-verbal behaviours, Visual direction, duration and frequency of document usage, emotional expression	√	√									
PSYCHO-PHYSICAL	Stress levels				√		√					
AFFECT or ATTITUDINAL	Users' opinion, Preference, Satisfaction, Acceptance				√			√	√			
COGNITIVE	Level of users' knowledge and understanding of system Accuracy of mental model					√					√	
PERFORMANCE (Usage prediction)	Task times Levels of system complexity									√		
EXPERT PERFORMANCE APPRAISAL	Level of conformance with guidelines, standards and design criteria											√
EXPERT OPINION	Level of satisfaction of the expert based on his or her existing knowledge or experience								√			

Table 3.4. Some Suggested Measures of the Usability Construct.

Tyldesley (1988). Time to complete task, percentage of task completed, percentage of task completed per unit time, ratio of success to failures, time spent in errors, percentage number of errors, percentage or numbers of competitors that do this better than current product, number of commands used, frequency of help or documentation use, time spent using help or documentation, percentage of favourable/unfavourable user comments, number or repetitions of failed commands, number or runs of successes and of failures, number of times the interface misleads the user, number of good and bad features recalled by the user, number of available commands invoked, number of regressive behaviours, number of users preferring your system, number of times users need to work around a problem, number of times the user is disrupted from a work task, number of times the user loses control of the system, number of times the user expresses frustration or satisfaction.

Shneiderman (1988). Time for a user to learn to accomplish a task, speed of performance on bench mark tasks, rate and distribution of errors, subjective satisfaction, users' retention of syntax and semantics over time with intermittent use.

Carey (1988). Information accuracy, information timeliness, information relevancy, correct information content, appropriate level of information scope, decision support capability, speed of performance, reduced error rates, user satisfaction, ease of learning, system responsiveness, enhancement of communication, system reliability.

Johnson, Clegg, and Ravden (1989). Visual clarity, consistency, informative feedback, explicitness, appropriate functionality, flexibility and control, error prevention and control, user guidance and support.

McClelland (1990). Time, speed of response, activity rate, etc., accuracy and errors, convenience and ease of use, comfort, satisfaction, physical health and safety, physical fit, physical effort and workload, stress and mental workload.

methodology. Hanson, Kraut and Faber (1984) have provided an interesting approach to the multivariate analysis of performance based data, and Dzida (1984) examined the dimensionality of attitudinal data using a multivariate perspective.

Such approaches have presented the development of the usability construct so that it is now not seen as a one-dimensional entity, but as multifaceted and of considerable complexity.

Ideally, an evaluation method should have perfect overlap with the ultimate evaluation. However, due to the complex nature of the usability construct, it may be unrealistic to expect such a situation to exist in the real world. Instead, software developers may need to use more than one evaluation method to gain the maximum relevant information about the usability of the software. In doing so, the software developer needs to pay attention to the overlap between the various evaluation methods and the ultimate evaluation.

The number of possible relationships is disconcerting. In contaminated evaluation methods, adjustments may be made to software that may not be related to the usability of the system. Also, in deficient evaluation methods, changes may need to be made, but may not be highlighted.

It is therefore important to decide that portion of the evaluation methodology which is related to the ultimate evaluation. It is then important to establish whether other evaluation methodologies can elicit information that the original methodology could not, thus contributing to general knowledge of the usability of the software. This can be done in a cumulative way so that each portion of the evaluation adds new and relevant information to the software designer's knowledge, and consequently reduces its deficiency.

An in-depth examination of user based evaluation methods, their assumptions, strengths and weaknesses may be of some help when examining evaluation relevance, deficiency, contamination, and redundancy of data. Such an examination is undertaken in the next chapter.

Chapter Four: Software Evaluation Methods.

There are many methods for evaluating the human-computer interface, which can be broadly divided into objective and subjective methods. Lea (1988) reported that some evaluators suggest that the subjective responses of the user are more important than objective data. Such evaluators have argued that knowing what the user thinks about system response time is more revealing than objectively measuring it. Although this approach seems feasible for discretionary software, it has dubious merit for life critical systems, such as air traffic control.

Another issue is the criteria that should be used to assess the effectiveness of the evaluation method. In this context, Johnson et al. (1989) proposed that methods used for evaluating user interfaces should be: systematic, based on existing criteria, iterative, general, participative, sensitive, simple to use by people who are unfamiliar with usability issues, face valid, related to the realistic usage of the system, and reasonably exhaustive.

The delineation of such attributes may be premature, however. At this stage more emphasis should perhaps be placed on fundamental base aspects of criterion development. Landy (1989), for example, has pointed out that criteria need to be reliable, valid, and practical. He has further asserted that there are substantial advantages to be gained by reducing criterion requirements to the three categories of reliability, validity and practicality,

> "First and foremost, it helps to point out that in spite of its unique position . . . criterion data must satisfy the same requirements as all other forms of data. If inferences are to be drawn based on criterion data, those data must be reliable and representative. In addition, there must be a

> practical scheme for gathering the data so that the cost does not greatly exceed the potential benefit. Great advances in the area of criterion measurement and development can be made if these three requirements are kept in mind." (Landy, 1989, p.151).

It is useful, therefore, to examine the current software evaluation methods this way. In particular, there are four major methods used: logged data, questionnaires, interviews, and verbal protocol analysis (Hietala, 1985; Yamagishi and Azuma, 1987). Little information is known about their strengths and weaknesses. Furthermore, although these methods are sometimes used alone, they are often used in combination (Neal and Simons, 1984a, 1984b; Hietala, 1985; Lea, 1988). This leads to problems of redundancy and contamination of information. As a means of clarifying the efficacy of these methods an examination of their theoretical and empirical properties is important.

4.1. System Monitoring, or Logged Data.

System monitoring, or logged data, can be viewed as the epitome of objective behaviourial data. It involves the unobtrusive collection of keystrokes, with accompanying time stamps, of the user interacting with the computer.

Drury (1987) reported that it has only been with the advent of modern microprocessor based computer systems, that psychologists have been able to unobtrusively, and relatively inexpensively, collect and analyse large accurate data sets of human performance. Penniman and Dominick (1980) has supported this view by stating that,

> "Prior to the advent of on-line interactive systems, the application of unobtrusive measures of user behaviour was

> most difficult. Now it is perhaps the easiest of techniques, using monitor programs built into the information system." (Penniman and Dominick, 1980, p.19).

Penniman and Dominick (1980) have suggested a methodology for the evaluation of system usage. As shown in Table 4.1 and Figure 4.1 this process is similarto the general evaluation methodology; that is, define objectives, collect data, and report data. This suggested methodology is also of an iterative nature and therefore in line with current software development philosophies (Gould, 1987, 1988; Williges et al., 1987).

Collection of logged data can generally be accomplished in one of three ways (Kirakowski and Corbett, 1990). The first is the software oriented approach, in which a data collection routine is embedded within the software package to be evaluated. Such an approach has been described by Yoder et al. (1985). To use this method access to source code is necessary, which limits the application of the approach to mainly in-house evaluations. Even when access to the source code is available, Kirakowski and Corbett (1990) have suggested that some changes in the operating system may need to be made, which rapidly becomes fraught with problems.

The second approach is an on-line tap, as described by Neal and Simons (1984a, 1984b). With this approach a direct hardware tap is placed between the central processing unit (CPU) and the keyboard (see Figure 4.2). The advantage of the on-line tap approach is that evaluations are not limited by access to source code, which removes the necessity for imbedding data collection sub-routines in the software to be evaluated. By not requiring access to the source code, using the on-line tap method makes comparisons between software using logged data feasible. However, the drawback of such an approach is the necessity for sophisticated hardware.

Table 4.1. General Methodology for Monitoring and Evaluation of On-line Information System Usage (from Penniman and Dominick, 1980, p.24).

1. Determine the monitoring / evaluation objectives.

2. Determine the specific parameters to be monitored initially,based upon the overall objectives.

3. Design and implement the monitoring facility into the system.

4. Design and implement the data analysis tools to be used in analysing the monitored data, if such analysis tools are not already available.

5. Design and conduct the monitoring experiment to collect the data to be analysed.

6. After the experiment has been completed perform the data analysis, making evaluations and drawing conclusions, as appropriate.

7. Identify system improvements and enhancements as implied by the results of the analysis.

8. Identify monitor improvements and enhancements as implied by the results of the analysis. This may involve adding new parameters that were found necessary, deleting existing parameters that were found not necessary, or modifying existing parameters to collect more detailed or more aggregated data.

9. Identify experimental design improvements and enhancements.

10. Apply the results of phases 7 through 9 to implement the identified improvements and enhancements to the system, to the monitor, and to the design of the data collection experiment.

11. After a period of time which depends upon the initial objectives, cycle back through phases 5 through 10.

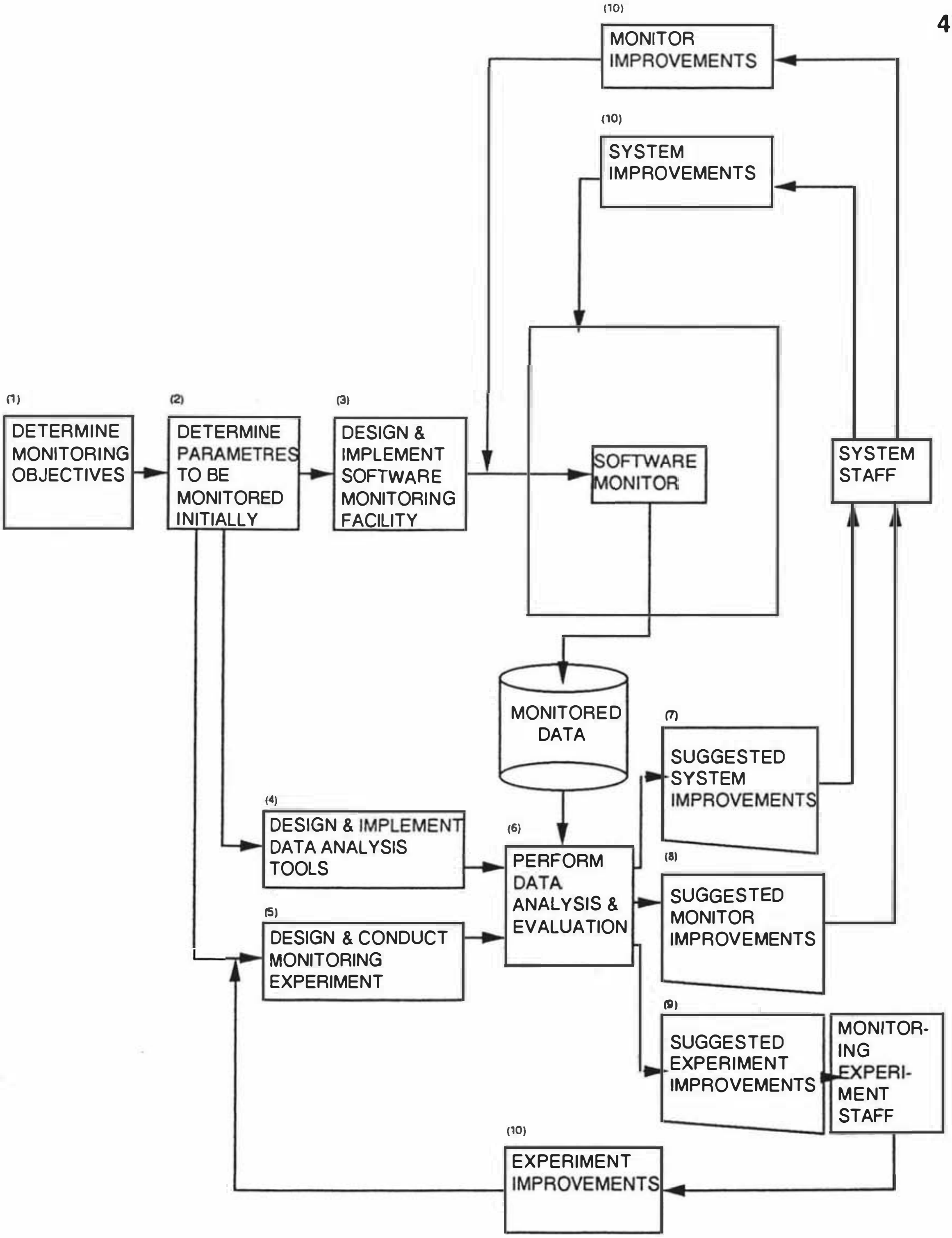
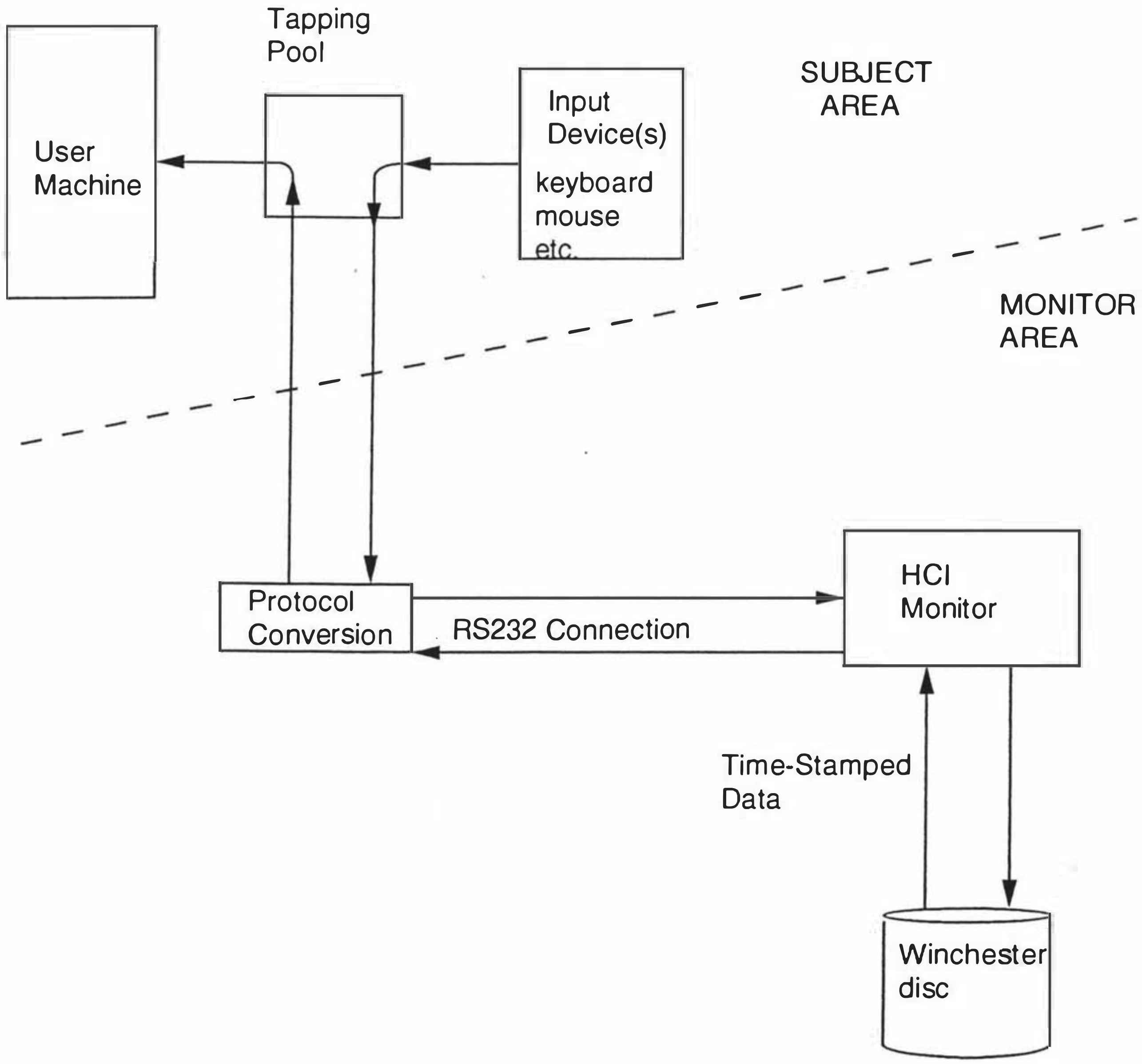

(10)
MONITOR IMPROVEMENTS
(10)
SYSTEM IMPROVEMENTS
(1)
DETERMINE MONITORING OBJECTIVES
(2)
DETERMINE PARAMETRES TO BE MONITORED INITIALLY
(3)
DESIGN & IMPLEMENT SOFTWARE MONITORING FACILITY
SOFTWARE MONITOR
SYSTEM STAFF
MONITORED DATA
(7)
SUGGESTED SYSTEM IMPROVEMENTS
(4)
DESIGN & IMPLEMENT DATA ANALYSIS TOOLS
(6)
PERFORM DATA ANALYSIS & EVALUATION
(8)
SUGGESTED MONITOR IMPROVEMENTS
(5)
DESIGN & CONDUCT MONITORING EXPERIMENT
(9)
SUGGESTED EXPERIMENT IMPROVEMENTS
MONITOR-ING EXPERI-MENT STAFF
(10)
EXPERIMENT IMPROVEMENTS

Figure 4.1. A Suggested Methodology for the Monitoring and Evaluation of On-Line Information System Usage (see Table 4.1) (from Penniman and Dominick, 1980).



'e 4.2. The Direct Hardware Tap Method for Collecting On-Line Information System Usage (adapted from Theaker, Phillips, Frost and Love, 1989).

The third, and most basic, approach to logged data collection is to make a copy of the computer screen. A copy of the screen can be obtained by either directly feeding the video signal into a video recorder in parallel with the real screen, or by making a video copy of the screen. The disadvantage of this approach is that the video recording subsequently requires transcription when performance measures are to be used, procedures that require both time and effort (Kirakowski and Corbett, 1990). Furthermore, Laws and Barber (1989) report problems with such an approach when the refresh rate of the computer screen is greater than 50 Hz.

Another problem when using logged data are what measures should be recorded. Table 4.2 outlines some information that can be recorded using the logged data approach to system evaluation. Penniman and Dominick (1980) suggest a "minimal data set" of logged data, which consists of three basic categories: system usage profile and database usage profile data measures, user error and error recovery data measures, and user success and user satisfaction data measures.

Yoder et al. (1985) have suggested that when using the logged data method, evaluators should iteratively develop the statistics over time, have all the instrumentation programming done by one person, record intermediate statistics which summarise sessions in progress, devise a method for validating the instrumentation, expect to devote much effort to post-processing the data, and provide easy access to the statistics. Furthermore, it can also be suggested that times should be recorded to 100th of a second. This level of accuracy is required because some streams of interaction by skilled users may be very fast. As can be imagined, a very large data file quickly develops, making data reduction a critical factor in the utility of the logged data method.

Table 4.2. Some of the Suggested Information that can be Recorded Using the Logged Data Approach to System Evaluation.

Penniman and Dominick (1980).

User's name and affiliation, data of interactive session start and end time of session in both real and CPU time, real and CPU time durations for major phases of system processing, operation execution counts, full text of the operations considered, detailed context-dependent statistics for those operations of primary importance, user ratings, user comments, session cost, error recovery, error frequency counts, and error context, user success and satisfaction.

Neal and Simons (1984a, 1984b).

Time required to perform selected tasks, success or failure in completing the task, frequency of use of various commands or language features, measure of user problems similar to those used to measure learning difficulties.

Kretz (1985).

Error measures, time measures, goal completion measures.

Meister (1986).

Number/type of errors made under individual conditions, Number/type of keyboard input errors, number of requests for assistance, menu choices made, number/type of errors made while learning to use the computer system, reaction time to system display conditions, total time to perform individual tasks, such as completing forms, filing etc; total time spent operating the system, number of trials to learn to operate the system to some proficiency criterion, time required to learn to operate the system to criterion, number of successful task completions.

Kopp (1988).

Time spent to learn product, time taken to complete task, error recovery time, number of times help screens used, number and types of errors encountered, number of users who complete task successfully.

Rubin (1988).

Task completion (number of tasks correctly completed, number of tasks completed in a given time, time taken per task), command usage (frequency of use of different commands, use of command sequences, use of special commands), command abbreviations (use of abbreviations for particular commands, occurrence of mistyped command names), use of visual display (time spent looking at display, comparative data for different display formats), use of keyboard (time taken to execute command, comparisons with other devices) user errors (classification of error-types, frequency of error types across tasks, time spent in error situations, time taken to correct errors).

Problems stemming from the resolution of the data set has led Yoder et al. (1985) to report that analyzing such data is,

> "akin to archaeology, because one must infer behaviour patterns from low-level data 'artifacts'." (Yoder et al., 1985, p.907).

In support of this view Penniman and Dominick (1980) stated that,

> "On-line monitoring provides a means of collecting more data than we have ever had before on information system user behaviour. The problem quickly becomes one of data reduction and analysis - not data availability."(Penniman and Dominick, 1980, p.29.).

Logged data in its standard form is interpreted without the aid of the user who performed the original actions. This allows subjectivity and perhaps bias and error to enter the interpretation phase of the evaluation. Behaviour has a multiplicity of causal agents, but, when analysing logged data the evaluator must attempt to infer causality. This can cause problems. For example, in a situation where there was slow response time, was it due to the user not comprehending the user interface, disturbance from situational factors, or thinking what to do about tea? Was the wrong menu option chosen simply because of a slip of the hand, cognitive overload, or was the user just exploring the menu system? The alternative possibilities can be tested when a large sample is used, and behaviour patterns have the opportunity to emerge.

Hanson et al. have described such a process (1984). Hanson et al. used a variety of multivariate statistical analyses to examine the commands people used and the errors they made in office work. They showed that command profiles can be developed and clusters of commands established. These, in turn, may be used to produce a command lay-out based upon frequency of key usage. However, commercial developers do not have the resources to use large sample sizes to reduce such ambiguity. An examination of the literature shows that samples used in studies are often small, typically no larger than six to ten subjects.

Lea (1988) stated that system monitoring does have several methodological strengths. In particular, the data are accurate and are collected automatically, reliably and unobtrusively. Also, large quantities of cumulative time related data can be gathered to describe the interactions and the method can be used longitudinally.

Laws and Barber (1989) have noted that the use of automated event-logging reduces data extraction time and effort, and yields rich data for analysis. Furthermore, the data set comprises a complete history of interaction. Nevertheless, Laws and Barber report that problems of interpretation still exist. In particular,

> "they all pose problems to the resolution of the data set: the information may be too detailed for the majority of the experimental session, pre-specified grammars of actions are required to assist the automated extraction of valuable and meaningful data, and finally, it is often costly in terms of storage space . . . and preparation time . . . before any analyzable data can be obtained. Furthermore, such real-time data capture devices in isolation fail to provide contextual information to enable interpretation of interaction behaviours." (Laws and Barber, 1989, p.1306).

The large size of the data set would also suggest that using the logged data method would result in a large amount of redundant data of little use to the evaluator. One could suggest that if target, or critical aspects, can be identified before the evaluation, such aspects could be focused on, resulting in a reduction of the amount of redundant data.

Some problems with the use of performance data as indicators of usability have also been noted. Specifically, the assumption of maximisation of efficiency may not hold. In this context MacLean, Barnard and Wilson (1985) have noted that users do not always choose the most efficient means to accomplish a task, a point also noted by Eason (1984) who cites the work of Zipf (1965) in this context. Zipf proposed the law of "Least Effort," under which individuals strive to minimise effort. This causes Eason (1984) to suggest that,

> "It appears that, faced with a problem, people initially check whether they have a tried and trusted method which will or might work (a form of "inner search") and only when this fails do they turn to ("outer search")." (Eason, 1984, p.136).

This suggests that the validity of some performance measures may be suspect.

Mais and Giboin (1989) have also expressed concerns with the maximisation approach to system design. Here, in the context of "Help" facilities they argue that most help systems are designed with the notion that users wish to maximise their efficiency. However, evidence from their work tends not to support this notion. Rather, they introduced the economic concept of "satisficing" as a likely paradigm for user interaction. The satisficing notion stems from the original work of Simon and Stedry (1969). In the satisficing view of motivation, an individual is not viewed as a blindly maximising animal, but as a being with limited capacity for information. Choices are viewed not only in information and search costs, but also in the cognitive limitations of the individual. Furthermore, satisficing recognises that the aspirations of individuals are variable.

Yamagishi and Azuma (1987) remarked that in their study, frequency of command use did not reflect the importance of the command nor the subject's satisfaction. Furthermore, Kondakci (1985) has stated that various measures of effectiveness often gave directly conflicting results. Such results suggest further difficulties with the construct validity of these measures. That is, with such ambiguity it is hard to establish which, if any, should be taken as the main predictor of usability.

Furthermore, there appears to be little information which addresses the reliability of the logged data approach. Intuitively, automated data collection would seem the epitome of reliability, and in one sense it is. During the data collection phase of the evaluation, information is recorded precisely and, excluding technical difficulties, can be regarded as reliable. However, during the data reduction and interpretation phases of the evaluation, subjective assessment, which could lead to error and bias, may enter the process. Little research has focused on the reliability of the inferences made when using logged data.

Conceptual problems arise when addressing the appropriate cutoff points for utilising performance times. For example, it can be argued that an individual formally begins a sub-task when the first key is activated. Another view may be that an individual begins a sub-task at the completion of the previous sub-task, and that the time between finishing one task and beginning the next is cognitive processing time. This leaves the actual key activation times as a relatively unimportant time reference.

There appears, in the past, to have been a fixation on the technical and not the psychological attributes of this method. Just because the information is accurately stored does not necessarily mean that such information has been accurately reduced, analysed, or interpreted. Nor does it imply it is a valid measure of the construct about which inferences are to be made.

4.2. The Questionnaire Method of Software Evaluation.

The questionnaire is a self report technique which has the property of rigidity. Meister (1986) stated the outstanding characteristics of questionnaires are the fixed order and format in which the questions are asked, and that they are in a written form requiring a written response. Lea (1988) noted that the method can be used to elicit a user's cognitions about interfaces, or self reports about their own behaviour, when interacting with the interface. Meister (1986) stated both positive and negative aspects of the questionnaire approach (see Table 4.3), and suggested that the written process is not a natural interaction like verbal communication. This could cause some data loss in the evaluation context.

Table 4.3. Advantages and Disadvantages of the Questionnaire Approach (from Meister, 1986, p. 167).

ADVANTAGES	DISADVANTAGES
Group administration (more respondents available more quickly).	Almost impossible to clarify obscurities if questions misinterpreted.
Remote administration (can be mailed).	Less motivating to respondents than interviews.
No variations possible.	No opportunity for analyst to explore missing details.
Requires less time / or fewer personnel to administer.	Speaking more natural to most respondents than writing.
More rapid responses and more data available in shorter time.	Little opportunity for respondent to explain response.

For pragmatic reasons the questionnaire can be considered an attractive choice of evaluation method. The questionnaire can be a quick, standardised, cheap, means of collecting large quantities of attitudinal data, points made by Karat (1988) in the context of computer user evaluation. Furthermore, it seems inherently sensible when addressing usability attributes to ask individuals how usable they perceived the product to be.

To overcome some of the problems associated with the questionnaire, Bouchard (1976) focused on the design process, and suggested that it is wise to subject any questionnaire to criticism from professional colleagues, and pre-test it on a new sample of respondents. Further to this, Bouchard outlined 21 considerations that should be used when designing a questionnaire (see Table 4.4). Bouchard insists that this list should not be used exclusively, but should be modified as necessary. However, many factors must be simultaneously considered if a good questionnaire is to be designed. This emphasises the complex nature of questionnaire construction.

The response format of a questionnaire can be open or closed. The open format permits respondents to write answers in their own words, which means that they are able to communicate in an unstructured, direct way. Closed items use established response alternatives (Meister, 1986). Rust and Golombok (1989) asserted that the most common form of closed items in the questionnaire are the alternative choice, the multiple choice and the rating scale. Of particular interest is the rating scale, which Rust and Golombok suggest allows respondents to express themselves more precisely than with alternate choice items. This may mean that the rating scale format is appropriate for the collection of user based attitudinal data. However, Rust and Golombok have pointed out that there are problems with the rating scale format. In particular, respondents will differ in their interpretations of the response options, suggesting that

Table 4.4. Considerations that Should be Used when Designing a Questionnaire (derived from Bouchard, 1976, p.381-382).

1. Is the question necessary?
2. Is the questionnaire repetitious?
3. Could the answer be obtained more easily elsewhere - by simple observation or from records?
4. Does the question contain more than one idea? Is it double barrelled?
5. Is the question adequate as it stands, or should complementary questions be asked?
6. Can the respondent answer the question?
7. Can the question embarrass the respondent?
8. Could it be made more specific or more concrete?
9. Is the question clear?
10. Would a memory jogger help?
11. Is the question too indirect?
12. Is the response format adequate from a theoretical point of view? Or from the respondents point of view?
13. If prerecorded answers are given, will they yield far more accurate answers than open ended questions?
14. Is the questionnaire susceptible to an order effect?
15. Can the items be arranged so that particular answers preclude the need to answer others?
16. Is an item likely to bias those following it?
17. Is ordering of the questions natural or reasonable?
18. Does the sequence maintain motivation?
19. Is the opening appropriate?
20. Will the respondent be able to read and understand the question?
21. End the questionnaire with a courteous Thank You or similar note of appreciation.

anchoring may be needed. Also, some respondents tend to choose either the extreme or middle scores. The tendency of respondents to choose extreme or middle scores has caused Meister (1986) to assert that although rating scales are generally more reliable than dichotomous multiple choice items, they are more prone to biases and errors than other types of items.

In turn, Landy (1989) suggested that these biases can be broken into the three major categories of leniency errors, halo effects and central tendency effects. Although a problem, Landy proposed that procedures can be undertaken to limit the impact associated with each of these errors. Specifically, leniency errors can be addressed by using a forced distribution format, or reducing the ambiguity of scales, as can the central tendency effects. Halo effects can be addressed by clear definition and the anchoring of dimensions.

Fiske (1982) also commented that memory, motivation, communication, and knowledge, all affect response accuracy. In practice, respondents may have forgotten information or may remember it incorrectly; respondents may be motivated not to tell the truth; they may not understand the question; and they may not know the answer. This suggests that, although practical in nature, the questionnaire may have reliability and validity problems.

Root and Draper (1983) have noted the potential for validity problems with the questionnaire, by remarking that although inexpensive and easy for software teams to use, little is known about how effective the approach is at identifying the good and bad aspects of a user interface. Consequently, Root and Draper examined the effects of different types of questions, user experience, and the method of administering the questionnaire. They concluded that questions using checklists about specific features of a system yield findings that are robust across methods of questionnaire administration, and across the amount of user experience. They are also reasonably consistent within subjects. Furthermore, comparable non-checklist questions had the same properties. Root and Draper reported, however, that internal consistency checks gave less then perfect results, suggesting that internal consistency measures should be built into questionnaires.

Root and Draper (1983) also stated that there was an effect for time delay, suggesting that administration should take place immediately after using the target software, and that the user's experience changes the discriminability of the responses, but not the overall results. This would suggest that this method is appropriate for a usability instrument for naive or inexperienced user groups. They also concluded that the checklist format was successful in identifying areas that need attention. However, this only applied to existing features and not proposed changes. Open ended style questionnaire items were also useful in identifying "sins of omission" within a software package.

Yang (1989) also examined the use of an evaluation questionnaire, concluding that there were advantages and disadvantages with the approach. Yang suggested that the questionnaire did highlight unanticipated problem areas. Also, it was suggested that the questionnaire gave the respondent more time to think about the issues compared to an interview, and was less threatening. Yang asserted, however, that although the questionnaire can provide useful information, and was inexpensive, it did not provide complete information due to recall problems.

Yang (1989) reported other problems with the questionnaire. These included users' understanding of the terms, problems with the use of quantitative data, and problems with regards to asking an opinion about new or unfamiliar concepts, points also raised by Root and Draper

(1983). Yang therefore suggested that wording should be as clear and simple as possible, qualitative data may be more appropriate than quantitative data, and that definitions should be provided for terms and concepts. Yang concluded that checklist questions are easily filled in and provide precise information, but they depend on a skilful anatomizing of all possibilities worth consideration. Furthermore, open-ended questions are difficult to complete, but can provide valuable information. They are suitable for professional users, but not typical users.

It has been suggested that ratings can be used to provide quantitative data that may be used to solve software usability issues (Shneiderman, 1987; Yamagishi and Azuma, 1987). This implies that data may be used in a composite fashion. That is, either all attributes are of equal importance, or some weighting system can be derived. Johnson et al. (1989) have argued, however, that attributes are not of equal importance, and moreover, the use of data in a composite fashion requires the development of cumbersome quasi-statistical weighting's. Johnson et al. argued that when using such an approach data analysis became,

> "cumbersome and that more energy was invested in the abstract weighting of priorities than the debate about user preferences." (Johnson et al. 1989, p.259).

There are similar problems when comparing software packages. By examining each attribute on a pair-wise basis the problems associated with minimal levels of acceptable usability, or interactional properties, are not being addressed. If one attribute is poor, it can affect performance on other dimensions. As a consequence, results can become indecipherable. Moreover, such statistical combination fails to address the problems of associated variance within each attribute. Knowledge that attribute 1 has a mean rating of 4 on version A and a mean of 5 on version B is meaningless if the variance around the associated means is not examined. The score of 4 on A may have a standard deviation of 1, whereas the score of 5 on B may have an associated standard deviation of 6. This suggests that a large portion of the sample is in agreement in the case of version A, but not in version B. Furthermore, standard deviations are particularly problematic when small sample sizes are used, as would seem to be the case for much of the software evaluation literature.

Anderson (1989), on the other hand, described the development of a statistically based methodology previously used in capital rationing and tournament ranking. The method itself considers three measures of quality,

> "the frequency with which the attribute ratings of one package exceed those of another, the presence of outlier's, and the cumulative magnitude of attribute ratings on one package that exceed those on others" (Anderson, 1989, p.707).

Such a system requires some numerical sophistication and therefore would seem more useful as a tool for human factors specialists working in an advisory role. Such an approach may not be practically feasible for most "in house" evaluations. Although his work is conceptually appealing, Anderson (1989) did not provide any concurrent, or predictive information about the validity of the findings.

The questionnaire can also be labelled retroactive. Facets of the user's impressions that are tapped will be a function of the comprehensiveness of the evaluation form, and the comprehensibility of each evaluation question. In essence, the response of the user is dictated by the investigator's set of questions, which in turn makes content validity a particularly pertinent issue. The problem of comprehensibility and comprehensiveness is also further confounded by recall problems. If the questions do not adequately stimulate the memory with regards to the attribute questioned, problems with validity of responses become paramount. These validity problems are further compounded by memory decay, demand characteristics, sensitisation, and order effects of the questions. The prominence of the attributes may also cause problems. If the attributes questioned are peripheral to the user's attention they may have not adequately focused on the issue, causing distortion and bias.

Karat (1988) has noted the restricting nature of the questions and the ex-post facto nature of the responses. But despite these problems Karat asserted that,

> "For many situations these concerns are not serious enough to outweigh the economy of collection, scoring and analysis. In other situations the additional detail in more complete verbal reports describing the experience are called for."
> (Karat, 1988, p.896).

Although several questionnaires have been developed (Dzida et al., 1978; Root and Draper, 1983; Simes and Sirsky, 1985; Shneiderman, 1987; Yamagishi and Azuma, 1987; Ravden and Johnson, 1989) few have been described in a systematic way. Little is known about their development and empirical properties.

Dzida et al. (1978) have produced an empirically based usability questionnaire, that has a theoretical foundation. They used a classic approach to criterion development by using a large sample to describe the relevant system properties related to human factors. Next, subjects rated a total of 100 derived user requirements with regards to their relevance for user perceived quality. Data were then submitted to factor analytic procedures resulting in a reduced set of 57 requirements. These requirements resulted in seven orthogonal dimensions. Dzida et al. labelled these dimensions; self-descriptiveness, user control, ease of learning, problem adequate usability, correspondence with user expectations, flexibility in task handling, and fault tolerance. However, it should be noted that the resulting questionnaire only accounted for 44 percent of the total variance.

Simes and Sirsky (1985) used a theoretical approach to the development of a questionnaire that incorporated theoretical psychological issues in the usability of computer systems. This approach resulted in the identification of 15 dimensions: tailorability, control over amount of information, manipulation shortcuts, types of human-computer dialogues, state, translations, user overrides, information density, categories of information, coding, command uniformity, performance reliability, system responses, messages, and attention/notification. Although this theoretically based approach to criterion development is preferable to a purely empirical approach, Simes and Sirsky provided no information with regards to the psychometric characteristics of the questionnaire. As well, reliability and validity issues were not addressed.

Ravden and Johnson (1989) described their questionnaire in some depth. However, little actual empirical information is presented in their paper. The checklist sub-headings of, visual clarity, consistency, informative feedback, explicitness, appropriate functionality, flexibility and control, error prevention, and user guidance support, seem similar in nature to the factors described by Dzida et al. (1978).

The questionnaire is a promising evaluation tool. Of the possible formats, the open ended statements and rating scales seem most appropriate. However, the utilisation of quantitative information elicited from attribute ratings should be treated with some caution. The open ended format may also be used to supplement the quantitative rating system, and therefore provide a more qualitative aspect to the evaluation information. Of particular concern, as the reliability and validity of the questionnaire, simply being a practical method of evaluation is not sufficient.

4.3. The Interview as a Software Evaluation Method.

The interview is a self report technique that utilises the medium of verbal communication as a means of information gathering. Interviews for software evaluation generally involve a face to face, one to one, situation where an interviewer orally presents questions to the interviewee. The interviewee in turn responds using oral communication. Shouksmith (1978) commented that, in a sense, there is no such thing as the interview, but there are many interviews that can be used in many different ways and for many different ends. Sinclair (1990) suggested that the interview has eight general characteristics; these highlight some of the strengths and weaknesses associated with the interview approach (see Table 4.5).

Meister (1986) asserted that the interview is a generic technique that has both advantages and disadvantages. In particular, the interview can be as natural as conversation; so, apparently, it is quite easy to develop and use. However, this naturalness does conceal its complexity, a problem noted by Bouchard (1976) who suggested that,

> "the great power of the interview and apparent ease with which it is applied has led to a large number of abuses and misuses." (Bouchard, 1976, p.368).

As an aid to reduce the frequency of such abuses, Bouchard (1976) addressed such topics as the development and setting up of the interview, the types of interviews, interviewer selection, training, and monitoring. The central thrust of Bouchard's work was that the interview

Table 4.5. General Characteristics of the Interview (from Sinclair, 1990, p.84).

1. The use of an interview can serve to direct and accelerate the information flow.

2. The interviewer can explore unexpected information, or unexpected occurrences.

3. A well trained interviewer will be sensitive to individual needs of the respondents, and will adjust his or her behaviour accordingly, thereby improving the quality of the information flow.

4. Interviewers can help to motivate respondents to give more information about the topic during the interview.

5. For the advantages above to occur, the interviewer must be well-trained in interview technique, should have at least some knowledge of the topic areas (the more the better), and must be sensitive to people. Collectively these criteria are not easy to meet.

6. It can be difficult to find and schedule people for the interview session.

7. Interviewers bias may creep in; this might be due to the interviewer's own knowledge of the topics, interpersonal relationships between the interviewer and the respondent, or to more mundane things such as fatigue, and so forth. This constitutes an extra source of error.

8. Systematic recordings of data are difficult, and sometimes impossible. In certain instances it may take up to three times as long to sort and assimilate the data as it took to obtain it.

must be approached and implemented with the same rigour as any other psychological technique, and one must be aware of the possibility of complex interactions impacting on the interview situation and consequently affecting the validity of the approach.

One way of examining possible interview formats is to distinguish between the possible structure of the question and response format (see Table 4.6) (Bouchard, 1976). The Type I interview is described as totally structured, in that the respondent replies to standard questions using a specified set of responses. This is different to the Type II interview in which the questions are specified, but the response format is open. Bouchard reported that there has been much debate over the relative merits of each format, with no clear support for any particular position emerging. Bouchard did report, however, that in both the Type I and Type II interview, the nature of the question may be specified, but not the wording. This allows the interviewer to alter the wording to best suit the interviewee, and therefore enhance rapport. However, when using such practices, lower reliability may occur due to the lack of standardisation and the possibility of interpretation errors. On the other hand, Bradburn and Sudman (1979) cited evidence suggesting that such manipulations generally result in small non-significant effects.

Table 4.6. Four Types of Interviews Classified According to Type of Question and Type of Answer Required (from Bouchard, 1976).

	Specific Question	Unspecific Question
Specific Response	I	III
Unspecific Response	II	IV

As can be seen from table 4.6 the Type III and IV interviews differ from the Type I and II interviews. The Type III uses a specified response, but an unspecified question format (Bouchard reported that at that time no one had appeared to use this format). The Type IV interview in contrast uses both unspecified questions and responses. This latter type is traditionally known as the clinical, or nondirective, interview and seems best suited to exploratory situations (Bouchard, 1976).

As an overall strategy Bouchard (1976) recommended the use of a "funnelled" interview format with feedback loops. With this approach the interview starts by addressing broad issues of a general nature. Next, the interview focuses on each issue in more depth. If further information on a broad issue, or an alteration to a broad issue, is needed, the interview feeds back to the broad issue.

When addressing the single dimension of the specificity of the questions, Meister (1986) suggested a quasi-formal approach, in which the interviewer covers set topics, yet is also free to ask further questions as a result of the interviewee's responses. It is suggested that such an approach will allow the interviewer to secure more detail than when using an interview schedule which is completely structured. Within the software evaluation context, Lea (1988) has taken a similar stance and suggested that,

> "The advantage of structure is that it ensures that the topics are covered . . . however, these advantages are achieved at the cost of possibly restricting the scope of the interview . . . Unstructured interviews are the most useful in this latter respect, and semi-structured interviews represent a compromise between the two approaches." (Lea, 1988, p.165).

Yamagishi and Azuma (1987) reported using a quasi-formal, funnelled, interview technique where users were first asked to offer their general comments about the system, and then to answer itemised questions. Critiques obtained were summarised and compared to the other methods used in their study. In this case it was found that the interview proved a good means of eliciting information that was of a global nature.

When addressing the analysis of interview data for software evaluation, Meister (1986) noted that the purpose of the interview is generally not to predict behaviour of large groups of people, but more for examining obscurities of, and understanding, objective test data. Consequently, Meister stated that the analysis of interview data tends to be a qualitative, content-oriented approach.

The validity of the interview is also of concern when evaluating the approach as a usability evaluation method. Essentially, the interview assumes that what the respondent says is true. Without reference to performance data the interviewer has no way of determining the validity of the information. Meister (1986) presented some work on the reliability and validity of interview information; however, the results seem inconclusive. In line with this finding, Landy (1989) remarked that when information can be verified it tends to be more reliable than information that cannot. It would appear, therefore, that the reliability and validity of the more objective information, for example, the interviewee's biographical data, would be more reliable and valid than their subjective opinions, which will be affected by a host of factors including perceptual errors and demand characteristics. Lea (1988) also noted that,

> "Details of interviews are rarely supplied in reports of evaluation studies. This makes it difficult to assess the significance of the information they turn up." (Lea, 1988, p.165).

Weiss (1975) suggested that there are many potential sources of error within the interview situation. In particular, Weiss noted that the predisposition of the respondent and the interviewer, the procedures used in the study, and the interaction between the respondent and the interviewer, may all be sources of error. Furthermore, he went on to say that several factors will also impinge on the validity of the responses.

Here, social desirability, acquiescence, and deference may all impact on the validity of responses, making the poorly designed and administered interview error prone.

Problems of accurately recording and transcribing the data elicited during the interview are also a potential source of unreliability. Tape recording the interview may be one way around these problems. Bouchard (1976) stated, however, that although tape recordings can be made successfully in group settings, they have a strong differentiating effect in the diadic situation. That is, some individuals talk to excess, while others "clam up." Secondly, the existence of a tape which must be transcribed, poses a threat to the confidentiality which respondents expect. Careful coding of information may help to overcome the confidentiality problem, however.

Despite the shortcomings of the interview, Bainbridge (1979) concluded that,

> "a carefully organised interview is probably the best technique to use when time and equipment are limited, as it should give information on general principles of behaviour and plant." (Bainbridge, 1979, p.435).

4.4. Verbal Protocol Analysis for Software Evaluation.

Bainbridge (1990) remarked that there are many complex jobs in which the outcome of thinking does not emerge in observable action. Bainbridge argued that to be able to train and support these types of jobs, we need knowledge of the cognitive processes involved. One way to obtain such information is to ask people to "think aloud" while undertaking such tasks. These reports are known as "verbal protocols" and are essentially reports of the mental processes used during the task.

However, the validity of such reports has been under discussion for some time. The validity problem primarily revolves around the information tapped during such exercises. In particular, does the individual who undertakes such a process have access to the higher order, or "meta-cognitive," thought processes? Some would argue not.

Ericsson and Simon (1980,1984) have noted that, for some time, there has been a trend within psychology to view verbal reports as suspect data, asserting that,

> "behavioursim and allied schools of thought have been schizophrenic about the status of verbalisations as data." (Ericsson and Simon, 1980, p.216).

Problems with verbal reports seem to stem from the original work of Boring (1953) which discredited the practice of classical introspection as a valid psychological technique. More recently the influential work of Nisbett and Wilson (1977) has also posed problems for the potential use of such data. Nisbett and Wilson conducted an in-depth review of the verbal protocol research and concluded that subjects have no access to their own "higher mental processes" and therefore cannot reliably, or correctly, report on them.

Nisbett and Wilson's (1977) work has, however, been criticised (Ericsson and Simon, 1980; Hoc and Leplat, 1983; Praetorius and Duncan, 1988). Praetorius and Duncan suggested that Nisbett and Wilson have made inaccurate extrapolations from their results, and that it is quite natural under some circumstances for individuals to be unable to report,

> "how we do what we do, or why we think or do as we do." (Praetorius and Duncan, 1988, p.310).

Specifically, it is argued that in many cases subjects have not been supplied with appropriate media, or means of expression, adequate for eliciting the information the investigator is seeking.

In an attempt to clarify the validity of verbal reports as data, Ericsson and Simon (1980) provided an in-depth review and theoretical amalgamation of the knowledge of the verbal report method. They presented a generalised processing model to aid in the theoretical and empirical examination of verbal reports as data.

Ericsson and Simon (1980) began with two base assumptions. First, that a cognitive process can be seen as a sequence of internal states, which are successively transformed by a series of information processes. Secondly, that information is stored in several memories that have different capacities of storage and accessability. The broad distinction is between what can be referred to as short term memory (STM) and long term memory (LTM). The short term component is seen as having a limited capacity with, and/or, intermediate duration and long term memory as having a large capacity and relatively permanent storage, but with slow fixation and access times. Within this model it is assumed that information recently acquired by the central processor is kept in the STM and is directly accessible for further processing, whereas information from LTM must first be retrieved before it can be reported.

The important hypothesis advanced by Ericsson and Simon (1980) was that,

> "due to the limited capacity of the STM, only the most recently heeded information is accessible directly. However, a portion of the STM is fixated in the LTM before being lost from the STM, and this portion can, at later points in time, sometimes be retrieved." (Ericsson and Simon, 1980, p. 223).

As a derivative of this hypothesis Ericsson and Simon (1980) made two major distinctions,

> "First, the time of verbalisation is important in determining from what type of memory the information is likely to be drawn. Second, we make a distinction between procedures in which the verbalisation is a direct articulation or explication of the stored information and procedures in which the stored information is input to intermediate processes, such as abstraction and inference, and the verbalisation is a product of this intermediate processing." (Ericsson and Simon, 1980, p.223).

Using this processing model, Ericsson and Simon (1980) have been able to predict when verbal reports will and will not be valid. Of perhaps more importance is the finding that by using this model they have produced data that are consistent with the experimental findings reported by Nisbett and Wilson (1977).

Ericsson and Simon (1980) concluded that evidence of inconsistency can only be found under two possible conditions. First, when cues used to access the LTM are too general, which can result in information related to, but not identical to, the information sought to be retrieved. Secondly, when subjects use intermediate processes to infer missing information, which is then used to fill out, and generalise, incomplete memories before responding.

Discussion aside, Nisbett and Wilson (1977) concluded that individuals do have access to specific data,

> "The individual knows a host of personal historical facts: he knows the focus of his attention at any given point of time; he knows what his current sensations are and has what almost all psychologists and philosophers would assert to be "knowledge" at least quantitatively superior to that of observers concerning his emotions, evaluations, and plans." (Nisbett and Wilson, 1977, p.255).

It would seem that it is this information that software developers would want access to in the evaluation context. Therefore, Ericsson and Simon's (1980) comment that,

> "For more than half a century, and as the result of an unjustified extrapolation of a justified challenge to a particular mode of verbal reporting (introspection), the verbal reports of human subjects have been thought suspect as a source of evidence about cognitive processes." (Ericsson and Simon, 1980, p.247),

has particular merit. One obvious area of contention pertains to access to higher order, or meta-cognitive, processes. This may, however, not be of primary interest to the software developer. Simply put, the designer wants reliable, valid, and practically obtainable information about the usability for their product. They may not be interested in the exact nature of the mental models elicited by the interface. They want to know just how easy it is to use, where problems occur and how to fix them.

In support of the verbal protocol methodology Kirakowski and Corbett (1990) observed that human computer interaction is primarily stepwise, making it well suited for concurrent verbalisations. Also, Robson and Crellin (1989) stated that protocol analysis has the advantage of being a convenient method for collecting a rich form of data. In the methodology, data and theory are separated, resulting in no value judgements imposed by any particular theoretical perspective adopted, and the data can be readily analysed in a number of ways. The problems with the method include identifying the correct level of analysis, loss of detail through data compression techniques, time and effort in analysis, and problems with interpretation during transcription.

Bainbridge (1979, 1990) has also documented problems specific to verbal protocols. Operators may not document what is "obvious" to them; most people think more quickly than they can talk. Furthermore, some practical problems also may arise, such as a long period of recording may be necessary to obtain a representative sample of activities. It is not possible if the task is verbal, and analysing data is both time consuming and difficult. Sweeney and Dillon (1987) referred to the time consuming nature of verbal protocol analysis, suggesting that analysis time on average will take ten times the data capture time. Sweeney and Dillon also commented on the reliability issues surrounding verbal protocol analysis data suggesting that,

> "True protocol analysis requires the use of independent raters to score the data in terms of an agreed upon rating procedure, from which reliability measures of any conclusions drawn from the data can be obtained . . . However, the principle of the technique can be more loosely used to provide a record of interaction from the users perspective and thus offer an insight into the effect of particular interface features on interaction." (Sweeney and Dillon, 1987, p.369).

Karat (1988) noted that in practice one rarely finds a subject who gives a quality report with little prompting, concluding that,

> "a significant number of subjects will simply not provide very useful verbal reports. It is generally the case that under half of the subjects drawn from a typical undergraduate subject pool will provide good protocols." (Karat, 1988, p.898).

Lund (1985) reported the use of an aided subsequent approach to verbal protocol analysis. Here the user generated a protocol while viewing themselves undertaking a computer based task. The advantage of this approach is that the process of generating the protocol does not interfere with the task, but there may be a cost in the reliabilityof the now retrospective verbal data, caused by bias, and after the event rationalisation of behaviour.

Hoc and Leplat (1983) have addressed the reliabilityproblem by evaluating the different modalities of verbalisation related to the "thinking aloud" kind of sorting task. In particular, Hoc and Leplat examined the efficiency of simultaneous verbalisations and the unaided, and subsequent aided, verbal protocol analysis procedures. They found that simultaneous verbalisation slowed the process of automation of the activity and produced some disturbances in the execution of the task. They therefore recommended that this procedure should not be used outside problem-solving activities.

Hoc and Leplat (1983) have also recommended that unaided subsequent verbalisation should be avoided because,

> "it produces too much distance from the task and there is a risk of obtaining data which are not very valid for the activity being studied." (Hoc and Leplat, 1983, p.302).

They stated that for a logical task, aided subsequent verbalisation was the most favourable, concluding,

> "Although under these conditions a slight slowing down in the stabilisationon a procedure is noted, the data obtained are similarto simultaneous verbalisation without perturbing the execution of the task (and therefore the process being studied)." (Hoc and Leplat, 1983, p.302).

It would seem, therefore, that the subsequent aided verbal protocol analysis is the most appropriate verbal protocol analysis technique when examining human-computer interaction. This form of protocol analysis may be accomplished by video taping the user interacting with the target system, and then playing this tape back in real time to the user with the "think aloud" instruction.

Video taping human-computer interaction does, however, pose both psychological and technical difficulties. In particular, there is the problem of reactivity caused by the obtrusiveness of the video equipment. Furthermore, the refresh rate of the screen can create difficulties when filming the computer screen. Particularly so, if the refresh rate is much greater than 50 Hz. This is because a flicker results on the video image reducing the resolution of the events being observed (Laws and Barber, 1989).

Despite the controversies associated with verbal protocol analysis, Bainbridge (1979) remarked that,

> "Preliminary interviews and observation would indicate the problems and areas of interest. Static simulation with careful interviewing would give information about both general and specific knowledge, while verbal protocols and associated observation would show the details of behaviour in real conditions of complexity and time." (Bainbridge, 1979, p.435).

4.5. Other Evaluation Techniques.

del Galdo et al. (1987) commented on the use of the critical incident approach as a possible means for evaluating the usability of an interface. Essentially, this method entails users undertaking specified sub-tasks and reporting critical incidents. While reporting the critical incidents the user is also asked to rate the perceived severity of the incident. Incidents are then classified to identify common causal elements. Next, the incidents are ranked in order by frequency and severity, which can then be fed back to the software designers. del Garbo et al. conclude that this method was successful for collecting user input about an interface. This input can then be translated into clear cut problems. Concomitantly, the method can also be used to identify good aspects of the interface.

del Galdo et al. (1987) stated that further research still needs to be conducted on the efficacy of the critical incident approach. In particular, although alterations can be made based upon the reported critical incident, the relationship between such alleviations and a better interface has not been ascertained. Nevertheless, this method does have some potential, for the same questions posed by del Galdo et al. can be justifiably addressed to all usability evaluation methods. Of concern is the problem of the representativeness of the critical incidents cited. If the method is used without memory aids, the approach may again be limited by the memory of the user and therefore may be deficient.

Praetorius and Duncan (1988) described research analysing video tapes of subjects performance using computers. Subjects were required to comment on their performance in fault diagnostic tasks. By using this technique, many potential misunderstandings and false interpretations of performance were avoided during the analysis of the video record. Lea (1988) seemed to support this kind of process suggesting that evaluators can conduct an interview after videotaping the subject and therefore the user can describe the experience and explain interactions.

Along similar lines is the video analysis methodology outlined by Laws and Barber (1989). With this method users' are video taped while using the software that is to be evaluated. Next, the analysable data are extracted from the video tape in one of two forms: viewing the tape and scoring the intensity of behaviourial reactions on appropriately devised rating scales, or inserting a time base which normally constitutes an electronic frame code. The time base can be inserted on the video so that events of interest may be logged in association with the number of the frame in which it occurred. Laws and Barber argued that this system can be used in a similar fashion to the conventional logged data approach for evaluating the user interface. However, due to the nature of the tape, many of the problems associated with data interpretation are eliminated. Apart from the technical problems of videotaping a monitor screen, Laws and Barber explained that items of interest must be identified and logged from the video image, which makes this approach more time consuming than logged data collection during the data collection phase of the evaluation. However, they argue that these problems are more than offset by the interpretability and utility of the method.

Tynan (1985) reported on a randomly generated self report technique as an adjunct to conventional methods used to evaluate the human computer interface. With this technique, users are signalled to note their behaviour by some external event that is not under their control. Tynan observed that to use this technique one must make the method as non-intrusive as possible. One way this can be done by using small portable hand held computers to produce a random signal, which can be produced through a remote piezoelectric device that can be worn on a lapel. At the onset of the signal, users note their behaviour. Reporting of behaviour, in turn, must be as convenient as possible.

Tynan (1985) asserted that the randomly generated self report method has the advantage of being able to be used within the information processing environment. By being able to be used in the information processing environment, the method may be more useful in the field study setting, as opposed to the more intensive methods that may be used during the initial development of the user interface.

Similar to the randomly generated self report, is the prompted diary described by Kirakowski and Corbett (1990). Kirakowski and Corbett take a developmental view of software development and suggest that the real issues of interest emerge over a span of time. Therefore, a diary is used to supplement other measures. With this approach a diary is kept by users, which contain prompts on how they are developing by having them compare their experiences and feelings with those arising from the previous session and to project how they expect to get on in the future. Furthermore, specific prompts may be used to ask about features of direct concern to the developer. Although Kirakowski and Corbett felt that developing the prompts may be difficult they suggested that there are significant advantages to be gained by using this approach, particularly in the form of sensitive, and context-focused information. However, the validity of this approach in terms of the effects of demand characteristics and a possible skew in information is not addressed by the authors.

Another method known as "impact analysis" has been described by Gilb (1985). Gilb claimed that this approach enables multiple viewpoint analysis of any system design. When using this method designers assess the impact of design attributes in the form of a percentage of the target level for a particular attribute. Gilb stated that the accuracy of these estimates is not the primary objective, but to identify weak patches in the design. Although this method has been described with reference to designers, the approach could equally be applied to users. The literature suggests, however, that it is not a widely used method.

One other approach that would seem to have potential as a software evaluation method is the psychophysiological approach. The approach seems useful because it employs both psychological and physiological criteria. Gale (1985) provided an in-depth discussion of the potential of the psychophysiological approach in office system design. Gale advocated using an experimental office system which permits both office and research functions, and suggests that behaviourial, subjective experiences, and physiological responses should all be recorded using a longitudinal design. Gale argued that by using such an approach large quantities of useful experimental data may be gathered and examined.

Grudin and Maclean (1985) have reported an adaptation of the psychophysical approach for examining performance and preference trade-offs in human-computer interaction. In one experiment they manipulated two variables in a data entry task by manipulating the size and method of entry of a code string the user had to enter. It was found that the preferred methods of entry altered as the length of entry string altered. By using psychophysical methodology, they were able quickly to focus on the trade-off point, when a user would choose one method of entry over the other. Although the method still needs further refinement the approach holds promise. They suggested that the method has a variety of uses particulary in the field of early comparisons between interfaces prior to between-subject testing in the laboratory or field. Grudin and Maclean remarked that the method can be used to,

> "balance our general reliance on performance measures with simultaneous measures of preference." (Grudin and Maclean, 1985, p.741).

However, the psychophysical procedure has several limiting factors, including the within-subjects design, and the repetitious nature of the tasks. Furthermore, experimental sophistication on the part of the evaluator is required, further limiting the potential use of the method.

Along similar lines to the psychophysical method, Krueger (1989) described using eye movement analysis to evaluate the user interface. Krueger made the distinction between voluntary and involuntary eye movements and argued that good interfaces should cause a minimum of involuntary eye movement. From the results Krueger commented that the search time and the total number of eye fixations was highly correlated, and that the number of voluntary eye movements allows for an evaluation of cognitive workload. Further to the technical restrictions of using this system within the commercial environment, more research may need to be conducted examining the relationships between such objective measurements, and the subjective impressions of effort and difficulty, as expressed by the user.

4.6. Comparisons of Evaluation Methods.

When considering the evaluation methods, it is important to consider research that has specifically compared the information gained by using each method, and those studies that have compared evaluation methods.

Sweeney and Dillon (1987) compared standard performance measures, interactive error analysis, subjective ratings / questionnaires, and verbal protocol analysis. They said that,

> "performance measures are quick, objective and easy to capture. However, they provide no indication of how an interface can be improved or why a particular interface leads to faster/slower, or accurate/inaccurate performance."
> (Sweeney and Dillon, 1987, p.396).

In contrast, interactive error analysis data were more difficult to gather but yielded information on the effectiveness of dialogue design and the level of knowledge of the user. This method was, however, unable to pinpoint areas of the interface with which users had difficulty. Verbal protocol analysis was time consuming and the data were difficult to collect. Also, verbal protocol analysis sometimes required expert analysis. Nevertheless, the data obtained were the most detailed of all the methods.

Yamagishi and Azuma (1987) have compared the logged data, the questionnaire, the interview, and the verbal protocol analysis methods of software evaluation. In their study a counter-balanced, within subjects, design was used to evaluate two versions of a software development support system.

The analysis consisted of comparing the information elicited upon a number of dimensions. In particular, a comparison between information elicited during the interview and protocol analysis suggested that there was substantial agreement. However, the retrospective interview yielded more global, or high level, issues than the concurrent verbal protocol analysis. Automated data could be used to see if the designer's assumptions of user behaviour matched reality, and for providing backup information for the interpretation of verbal data. Automated data were of limited value, because they could not be used to indicate user satisfaction. The questionnaire provided a straight forward quantitative score, but it provided little information about what and how the interface could be improved.

One dependant variable chosen by Yamagishi and Azuma (1987) was the time required to conduct the evaluation. It was observed that the protocol analysis was the most time consuming, with an average of 6.3 hours per 0.5 hour experiment time, with the bulk of this time being consumed by the post experiment analysis (a figure similarto that cited by Sweeney and Dillon, 1987). They also noted that complete dictation and encoding was not required for the case where specific problem areas for a system could be identified. Table 4.7 presents the comparative times devoted to each evaluation method reported by Yamagishi and Azuma. Although of heuristic value, such a comparison is misleading. The time required to conduct the evaluation will be a function not only of the experimental procedure but also the data analysis. That is, the quantitative score is easily calculated by means of a standard data analysis package, but such a package was not used for the protocol analysis. Such techniques are available, and may have reduced data analysis time. Furthermore, it could be argued that if the questionnaire had been designed to yield more qualitative data, or open ended responses, data analysis time would have been increased. The comparative time approach used by Yamagishi and Azuma could therefore be confounded by such problems. It may well be that rank ordering, or paired comparisons, such as described by Blum and Naylor (1968), may have been a more useful approach.

Table 4.7. Comparative Time Required to Conduct the Evaluation (from Yamagishi and Azuma, 1987, p.172).

Total 1223.5 Hours	
Protocol analysis	71.0%
Questionnaire	16.1%
Logged data	7.6%
Interview	5.3%

Furthermore, there are problems with the applicabilityand generalisabilityof the Yamagishi and Azuma (1987) study. Specifically, a small sample size was used (seven subjects) which limits the generality of their findings. Also, subjects consisted only of software designers, which may limit the generalisabilty of their findings to other groups.

Yamagishi and Azuma (1987) concluded that,

> "both the strengths and weaknesses of each technique were isolated. Protocol analysis and interviews are time consuming, but useful in identifying problem areas for a system. Questionnaire and logged data analysis, both of which can produce some sorts of quantitative results rather easily, are useful only for limited purposes, such as for comparative analysis among systems, versions or categories of users." (Yamagishi and Azuma, 1987, p.167).

4.7. Composite User Based Methods.

Neal and Simons (1984a, 1984b) described the development and use of a composite usability evaluation methodology known as the playback method. They stated that,

> "The central idea is that while a user is working with a system, the keyboard activity is timed and recorded by a second computer. This log of stored activity is later played back through the host system for observation and analysis." (Neal and Simons, 1984a, p.79).

The method therefore seems to be an elaborate logged data approach to usability evaluation. However, Neal and Simons explained that the system can be altered so that video cameras can be used to observe peripheral activity, such as the user's interaction with documentation. Also, the interaction can be played back in the presence of the user,

> "in order to obtain supplementary information about the user's thoughts or reasons for particular actions while performing the task." (Neal and Simons, 1984a, p.80).

In essence, then, the method has the ability to encompass components of the logged data, interview, and subsequent aided verbal protocol methods of user based evaluation.

Morris et al. (1988) reported the use of an on-line logged data tool using "playback." It involves interposing a laboratory computer between the keyboard and the host so that all user keystrokes can be time-stamped and captured in a file store. Morris et al. reported that,

> "Quite often only selected intervals of a session need to be analyzed" (Morris et al., 1988, p.438).

During replay the observer can switch between modes to reduce analysis costs. Four modes existed; synchronised replay of digital and video recording, replay of just video recording, replay of just the digital recording, and replay of just the journal notes.

Morris et al. (1988) concluded that the method is valuable for generating feedback to designers, which may make a positive contribution to achieving a good level of usability. They pointed out, however, that the method is very labour intensive and further work needs to be done to improve this aspect of the method.

Hietala (1985) has taken a similar stance to Neal and Simons (1984a, 1984b) by promoting the use of a composite method that is a combination of system logging, playback, and verbal protocols. Hietala argued that,

> "the emphasis on verbal protocols in connection with logging and playback can provide benefits that have not been recognised hitherto." (Hietala, 1985, p.100).

In particular, the method is unobtrusive, can be inexpensive and easy to realise, can support the extraction of relevant evaluation information by reproducing the actual work situation, and is capable of bringing out knowledge of users' mental models, and problem-solving processes. Hietala did suggest, nevertheless, that further work is needed to make the approach more versatile.

It would appear that many approaches are being advocated as possible means for evaluating the human-computer interface. These methods are varied, some requiring advanced technical and theoretical knowledge, and others relying on intuition. Little comparative research addressing the strengths and weakness of the approaches has been conducted. This is a serious omission, because iterative refinement of software rests upon the assumption of reliable and valid evaluation information. Furthermore, there appears to be a trend towards amalgamating some of the common methods in the hope that the interface will be adequately evaluated. Lea (1988) suggested that evaluators should use their knowledge and appreciation of methodology to select methods which provide complementary data types, and therefore utilise a triangulation approach.

When using multiple evaluation methods the evaluator is ensuring that a broader range of information is available. Nevertheless, in doing so, the evaluator may be duplicating information, which although adds credence to the obtained results, may be a waste of the commercial developers' scarce resources.

There is also a trend towards more complex evaluation systems. This trend must be seen as positive because the usability construct is multidimensional in nature. However, such systems must still be accessible and understandable by the practitioner. To be truly usable, the evaluation essentially must be reliable, valid, and practical. It is therefore important to focus attention on the development and utilisation of such methods. At present, however, little information is available even on the efficacy of the main software evaluation methods. It may therefore be premature to advocate more complex evaluation techniques when so little is known about the current strengths and weaknesses of the main methods in use. The work of Sweeney and Dillon (1987) and Yamagishi and Azuma (1987) have addressed some of the issues. However, information regarding the reliability, validity, and practicality of the methods is scarce.

Following on from Yamagishi and Azuma (1987), this research sought to compare and contrast four software evaluation methods. These were, logged data, the questionnaire, the interview and subsequent aided verbal protocol analysis. In particular, the efficacy of these evaluation methods were examined on several dimensions:

1. The degree to which information elicited by one evaluation method matched the information elicited from other evaluation methods.

2. The ability of each method to highlight problem areas in software.

3. The rate at which an evaluation highlighted user problem areas, within a set evaluation method.

4. The cross software robustness of the effectiveness of the methods.

5. The feasibility of using each evaluation method in a commercial environment.

Chapter Five: The Evaluation Study.

5.1. Experimental Design.

The present study examined the information obtained from the main software evaluation methods. Being a multiple method examination, either a between or within groups design could have been used. The within groups design poses problems with order effects, resulting in the necessity of using counter-balancing techniques. In contrast, the between groups design has problems with differential treatment effects, but these can generally be controlled by standardised conditions.

The study also examined evaluation information in relation to different software, because the type of software used, may have affected the information obtained from each of the evaluation methods. Three different sorts of business software (spreadsheet, word processor, and database) that represent the three major software types used in commerce and industry were used. The experimental design is shown in Table 5.1.

Table 5.1. Design Used in the Present Study.

	Evaluation method			
Software	Logged data	Questionnaire	Interview	Verbal protocol
Spreadsheet				
Word Processor				
Database				

5.2. Experimental Tasks.

The experimental tasks were designed to be representative of the work that a new, or naive, user would undertake. This is important for both the face validity and external validity of the study. Of importance when developing the tasks was the notion of the discretionary user. A discretionary user is one who chooses to use a system, rather than one who is required to use a system. Examples of discretionary users include people using a word processor for home use, and people purchasing a spreadsheet package to keep accounts for businesses. On the other hand, non-discretionary users include air traffic controllers who have set systems, or people moving into a setting where the software has already been decided upon and implemented.

As the functional properties of much software are becoming increasingly similar, attention is turning to the role of subjective impressions in the software purchasing process. If several packages are of similar price, and can perform similar functions, the deciding factor may be how much an individual likes the package. This is particulary so for discretionary users who have a choice of software.

Initial impressions have a disproportionate impact on the process of impression formation. This is known as the "primacy effect" and has been displayed by Luchins (1957). It can therefore be inferred that the first contact individuals have with a software package will have a major impact upon their impression formation. Thus, there is some (indirect) evidence to suggest that for example "first impressions count" when it comes to purchasing software.

The tasks used in the present experiment therefore were made similar to the tasks an individual starting on a system would face. These are entering data, saving data, recovering data, deleting data, and printing data. From these tasks, initial impressions of discretionary users may be obtained.

Although it would be advantageous for subjects to be thoroughly familiar with the software before evaluating it, researchers are confronted with the problem of what is the minimum number of instances needed to conduct informed software evaluation. Here Meister (1986) suggested that,

> "An absolute minimum is three trials, based on the need to secure some sort of variance estimate." (Meister, 1986, p. 45).

It therefore seemed appropriate to ensure that each subject was exposed to at least three instances of each of the core tasks. This enabled some impressions to form, a variance estimate to be established and some learning and familiarisation with the package to occur. (See Appendices 3 - 12 for the introductory material and tasks used in the study.)

5.3. Sample Related Issues.

Problems associated with the representativeness of the sample also needed to be addressed. Internal validity was achieved through the random assignment of the subjects to the different evaluation methods.

5.4. Software.

Software used in the study consisted of demonstration programmes developed and marketed by Borland International (1987). These programmes were chosen for two reasons:

The programmes are robust operationalisations of the software domain that they represent, both in terms of functionality and interaction style. It was intended that this would enhance the face validity of the study and improve the generality of the findings.

The programmes were also available in source code. This allowed for the modifications necessary to record the logged data. In each case the only modifications made to the programmes was the embedding of two routines (see Appendix 13).

One routine was positioned at the beginning of the programme, and requested the subject to answer questions about their perceived confidence when using general application programmes. The second routine was unobtrusive, recording, and time stamping, ASCII codes of the activated keys to 100th of a second.

5.4.1. Spreadsheet.

The spreadsheet used, was Microcalc (Turbo Pascal, version 4, 1987). The main menu system was started by pressing the "/" key. When this was done a set of nine options appeared at the bottom of the screen. These options were activated by pressing the first letter of the option desired and then the "Enter" key. Once activated, sub-menus appeared; these were started in the same fashion as the main menu system. Appendix 10 outlines the menu options in Microcalc.

The system used the conventional "cell" approach adopted by commercial spreadsheet packages. Microcalc included such features as formatting, calculations, insert and delete, editing and a utilities system. Microcalc appeared similarto commercial spreadsheet programmes.

5.4.2. Word Processor.

The word processor used, was Microstar (Turbo Pascal, Editor Toolbox, 1987). Microstar was a programme that incorporates a pull-down menu interface. The main menu was activated by pressing the "F10" function key, and was presented at the top of the screen. Movement between menu options was by using either the left or right arrow key and then pressing the "Enter" key, or pressing the first letter of the desired option. Once an option was activated, the sub-menu of the chosen option would appear. The sub-menu system was activated in a similar way to the main menu system. See Appendix 11 for an outline of the complete menu and sub-menu system used in the Microstar programme.

Microstar incorporated the features expected from a modern word processor. These included spelling-checking, font display, block command, windows, DOS shell, help screens, and macros. Furthermore Microstar incorporated a "What You See Is What You Get" (WYSIWYG) printer approach.

5.4.3. Database.

The database package used was another Borland International (Turbo Pascal, Database Toolbox, 1987) programme and was a dedicated database. It was similarto a database used in an office environment to control a company's client listings. Menu options were presented at the bottom of the screen. Options are chosen by using either the left or right arrow key or pressing the first letter of the desired option, and then pressing the "Enter" key. See Appendix 12 for an outline of the menu system used by the database.

To enter information into the database a "form filling" approach was used. Features such as update, delete, search, and report output were supported by the system. All options behaved in a manner similarto commercially available database packages.

5.5. Operational Definitions and Evaluation Method Development.

To ensure the generality of findings, and thus enhance the external validityof the study, it was important that all evaluation methods used were adequate operationalisationsof the methods that could be used in the commercial sector. Care was taken, therefore, to ensure that they were as robust as possible.

5.5.1. The Logged Data Collection Procedure.

The logged data method was an internal unobtrusive software oriented method. This procedure recorded the ASCII codes of all keys activated with the appropriate time stamps. This record could later be examined or transformed as necessary. See Appendix 13 for the base source code used to gather the logged data.

5.5.2. The Questionnaire.

Although several evaluation questionnaires were found in the literature (Dzida et al. 1978; Simes and Sirsky, 1985; Shneiderman, 1987; Yamagishi and Azuma, 1987) only one was empirically derived (Dzida et al., 1978). Therefore, a questionnaire was developed which incorporated all the best aspects of existing questionnaires. The structure of the Dzida et al. questionnaire was used, however, as the base instrument. The general strategy involved:

1. Obtaining questionnaires and ergonomic checklists.
2. Conceptually allocating each statement into one of the factors suggested by Dzida et al., and adding any other

factor/s that seemed necessary.

3. Eliminating duplication of statements.
4. Ensuring that, where possible, each question was jargon free.
5. Embedding each question in a seven point (low/high) rating scale, while also adding the two further response categories of Not Applicable (N.A), and Don't Understand (D.U).
6. Using open ended questions for each factor.
7. Piloting the questionnaire and altering as necessary.

The questionnaire that resulted was an amalgamation of three questionnaires (Simes and Sirsky, 1985; Shneiderman, 1987; Dzida et al. 1978) and one ergonomics checklist (Brown, 1986). The questionnaire developed used the original seven factors as described by Dzida et al. and one further factor, "formatting," that was derived from the Shneiderman questionnaire.

The questionnaire was initially examined separately by four individuals, who commented on its wording and style. After alteration, the questionnaire was piloted with four different individuals undertaking a word processing task and a further two different individuals undertaking a spreadsheet task. Comments on the form were collected, with particular attention being paid to the appropriateness and comprehensibility of the statements. Further refinements were then made to the questionnaire based on this information.

The resulting questionnaire had eight factors using a total of 117 statements. The factors and number of statements in each factor were: Programme Self-Descriptiveness (19), User Control of the Programme (14), Ease of Learning the Programme (20), Completeness of the Programme (11), Correspondence with User Expectations (15), Flexibility in Task Handling (7), Fault Tolerance (17), and Formatting (14).

To gain more qualitative information, at the end of each factor section, an open ended question was inserted that asked subjects about specific problems that arose and any suggestions for improvement to the software. One final open ended question was also inserted at the end of the questionnaire which requested any further comments or suggestions for improvement. The developed questionnaire appears in Appendix 14.

5.5.3. Interview Procedure.

The interview incorporated both structured and unstructured components as suggested by Bouchard (1976) and Meister (1986). This type of quasi-formal format meant that the interviewer could ask further questions as a result of the subject's statements, while also covering a formal set of topics. As a consequence, the interviewer was able to secure more detail than would have been the case using an interview schedule which was completely structured (Meister, 1986).

Subject's initially were asked to explain how they approached a specified sub-task. This was intended to put interviewee's at ease and to try to elicit indicators of the cognitive model they held. Next, subject's were asked to outline any problems they encountered while undertaking that task. Finally, subjects were asked for any suggestions for improvement they may have had. This schedule was conducted for each of the sub-tasks covered in the session. See Table 5.2 for the interview questions.

To end the session, respondents were asked to rate how difficult they found each of the sub-tasks. All responses were tape recorded and later transcribed.

5.5.4. Verbal Protocol Procedure.

The verbal protocol procedure used, was an aided subsequent verbal

Table 5.2. Interview Questions.

1. Entering data.
   - A. How did you go about entering data into the programme?
   - B. Did you have any problems entering the data? If so, what were they?
   - C. Are there any suggestions that would make entering the data easier to do?

2. Saving data.
   - A. How did you go about saving the data?
   - B. Did you have any problems saving the data? If so, what were they?
   - C. Are there any suggestions that would make saving the data easier to do?

3. Recovering data.
   - A. How did you go about recovering data back into the programme?
   - B. Did you have any problems recovering the data? If so, what were they?
   - C. Are there any suggestions that would make recovering the data easier to do?

4. Deleting data.
   - A. How did you go about deleting data?
   - B. Did you have any problems deleting data? If so, what were they?
   - C. Are there any suggestions that would make deleting data easier to do?

5. Printing data.
   - A. How did you go about printing data?
   - B. Did you have any problems printing data? If so, what were they?
   - C. Are there any suggestions that would make printing the data easier to do?

6. Did you have any other problems? If so, what were they?
   Are there any other suggestions that would make using the programme easier? If so, what are they?

7. One final thing I would like you to do is to rate these tasks from one to six according to how difficult the task was (one being easy, six being hard).

Thank you for your involvement in the study. Are there any questions you have about the study before we finish?

protocol similar to that reported by Hoc and Leplat (1983). This method was chosen for two reasons. First, Hoc and Leplat showed that this approach produced data patterns that were similar to the concurrent verbal protocol analysis. Secondly, they also showed that the method allowed for more than one individual at a time to undertake the evaluation task. This also meant that to control demand effects across the evaluation methods, all subjects were video taped while undertaking the experimental task.

When using this method subjects were asked to view the video tape of themselves undertaking the work they had previously completed on the computer. Subjects were asked to "Think aloud as if you were undertaking the task."

All verbal protocol sessions were held in a private room, with only the researcher and the subject present. Prompting was kept to a minimum. All protocols were tape recorded and later transcribed.

5.6. Analyses.

5.6.1. Equivalence of Groups: Subjects' Confidence Ratings.

To examine the adequacy of the randomisation process, the analysis of variance (ANOVA) procedure was used to examine equivalency of groups. Two separate series of ANOVAs were conducted. One series examined the equivalence of the perceptions of the individuals across the three software packages evaluated. The other series examined the perceptions of the individuals across the evaluation group assigned.

5.6.2. Analysis of the Logged Data.

The resulting logs were examined in several ways. Examination procedures can be broadly delineated into the two categories of actuarial and contextual analysis. The actuarial analysis examined the time taken to perform separate sub-tasks over consecutive occasions. Reduction in correlated mean and variance t-tests (Glasnapp and Poggio, 1985) was used to infer learning and ease of use.

Noise may appear in the logged data method in the form of typing errors, speed of typing, and distractions from the task being undertaken at that time. Although transformations may be undertaken to help reduce this noise, ambiguity may arise in identifying causal factors. Consequently, the transformations of these data may not be advantageous, and therefore was not undertaken. Other forms of actuarial analysis were also considered as appropriate.

Further to the strictly actuarial examination of the logged data, the logs were also examined in a contextual way. This consisted of an inspection of the movement through the tasks by the subjects, involving identifying the key strokes used by the subject to undertake each task. It was expected that this procedure would provide an account of how the subject approached each task. By inference, problem areas and mistakes made could be located. Frequency counts of the occurrence of problems were also made.

5.6.3. Analysis of the Questionnaire Data.

The means and standard deviations resulting from rating the questionnaire attributes were calculated. Cut-off scores were decided to operationalise a good, poor, or indifferent attitude. For the purpose of this research, a poor score was operationalised as a mean below 2.5, an indifferent score being reflected by a mean score between 2.51 and 5.5 and a good score being indicated by a mean score above 5.51. These cut-off scores were arbitrary being chosen simply to divide the scale into three equal segments. Standard deviations were also examined, as were the responses to the "Don't Understand" and "Not Applicable" response categories.

The internal reliability coefficients of the sub-scales for the questionnaire were derived using coefficient alpha. Here, Nunnally (1967) suggests that reliability is context sensitive and,

> "in the early stages of predictor tests or hypothesised measures of a construct, one saves time and energy by working with instruments that have only modest reliability, for which purposes reliabilitiesof 0.60 or 0.50 will suffice." (Nunnally, 1967, p.226).

Consequently, an internal reliabilitycoefficient of 0.60 was used as an acceptable internal reliabilityfor development and interpretation in this research.

Further to the ratings of attributes the questionnaire elicited open ended statements. Content analysiswas conducted using these statements with unitization at the referential level of analysis (Krippendorff, 1980). Problems highlighted were recorded and the overall frequency of problem identification examined. Krippendorff reported five possible levels of unitization (physical, syntactical, referential, propositional and thematic) with each having different uses and reliability.The referential level was chosen for this study because it seemed the best trade-off regarding reliabilityand usefulness.

5.6.4. Analysing the Interview.

The interview elicited statements about three aspects of each sub-task undertaken. Using a content analysis, statements were examined to the referential level of unitization (Krippendorff, 1980). Frequency of problem identification and problems highlighted were recorded and relationships between problems encountered and suggestions for improvement were noted and examined. Mean ratings were derived from the difficulty scores, which provided an index of how difficult each sub-task was perceived by the subjects.

5.6.5. Verbal Protocol Analysis.

The verbal protocol analysis provided an in-depth narrative of how each subject progressed through the sub-tasks. This analysis examined the problems encountered by subjects, and the thoughts and emotions they experienced. These statements were again analysed using a content analysis procedure, with unitization at the referential level of analysis (Krippendorff, 1980). Problems highlighted and the frequency of problem identification were recorded. Table 5.3 outlines the general approach to the data analysis for each evaluation method.

5.7. The Between Evaluation Methods Comparisons.

The first comparison of the evaluation methods used a case study approach. Here, a narrative of the information elicited by all the methods was discussed. This approach was used to gain a holistic feel for the type of information obtained from each of the software evaluation methods.

An empirical examination was conducted using the usability problems identified by each evaluation method as the dependent variable, with a problem being operationally defined as any statement which directly referred to, or any instance that impeded progress towards, task completion. Task completion was defined by the experimental task instruction sheet.

Frequency graphs of the number of problems identified by each evaluation method across each software package evaluated were produced. Inferential statistics included the chi-square test and hierarchical log linear analysis. All analysis was conducted using the SPSSX statistical package (1986). Hierarchical log linear analysis is a

Table 5.3. General Approach to the Data Analysis for Each Evaluation Method.

Logged data.

1. Times over consecutive occasions of performing the specified sub-tasks.
2. Any other relevant form of actuarial data.
3. Movement through the task in the form of times between each key stroke and the ASCII character associated with each key stroke.

Questionnaire.

1. Mean ratings of each attribute with associated standard deviation. The proportion of individuals answering "Not Applicable" and "Don’t Understand" the question will also be examined.
2. Comments in the form of open ended statements about the problems encountered and suggestions for improvement.
3. Coefficient alpha for each sub scale for each type of software evaluated.

Interview.

1. Reports about how individuals undertook each sub-task.
2. Reports about the problems encountered while the subjects undertook each sub-task.
3. Suggestions for improvement of the package on each sub-task.
4. Ratings of how difficult each sub-task was.

Verbal protocol analysis.

1. Narrative of how each individual progressed through each sub-task. This includes problems encountered and suggestions for improvement.

multivariate categorical modelling technique. The procedure examines which factors are required to produce a set of observed data. The objective of the technique is to examine the variables required to produce a set of data that is not dissimilar to the observed data. The model used, was a saturated model with backward elimination. This model begins with the full data set and eliminates those variables that are not required to generate a model that reproduce the observed data set (Tabachnick and Fidell, 1989).

The incidence of reporting a problem within an experimental group was examined. This was done by identifying six high frequency problem areas, specified by the logged data records. The percentage of individuals displaying the problems in the logged data group was then calculated. This figure was used as the standard to test the other evaluation methods. Subjective approaches rely on hindsight and recall and are subject to perceptual errors and decay. Thus, it was hypothesised that the subjective evaluation approaches of the questionnaire, interview, and verbal protocol analysis would exhibit a significantly lower rate of problem reporting than the logged data approach.

A comparison of the problems identified when two methods are combined was conducted. It was hypothesised that by combining two evaluation methods, a significant improvement in the number of problems identified by a single evaluation method would be observed. To examine this prediction, a series of matrices was constructed. Here the percentage of problems identified by each combination of two evaluation methods could be compared to each single method. This was done by first identifying the percentage of the total number of problems identified by each evaluation method used alone. The percentage of the total number of problems identified by using two evaluation methods, with duplications eliminated, was then calculated.

The log linear modelling technique was used to examine the properties of the evaluation methods when used together. Also of interest was the notion of incremental improvement. If two methods are used, was there a significant improvement, with regards to the number of problems highlighted, over using the best single method? Incremental improvement was examined using the chi-square statistic.

5.7.1. Specific Hypotheses.

In summary, it was hypothesised that:

1. the efficacy of the evaluation methods would be independent of the software being evaluated.

2. there would be no difference between the evaluation methods with regards to the number of problems identified.

3. the subjective approaches of the questionnaire, interview, and verbal protocol analysis would exhibit a significantly lower rate of problem reporting than the more objectively recorded logged data.

4. by combining two evaluation methods a significant improvement in the number of problems identified by a single evaluation method would be observed.

5.8. Practical Considerations: The Evaluator's Perspective.

The practical considerations associated with using the evaluation methods were considered by carrying out an introspective examination of the practical problems encountered while conducting the evaluations. The limitations of this approach must be acknowledged, and it was conducted to be informative, rather than providing a definitive account.

The examination consisted of initially identifying the steps associated with conducting and reporting an evaluation of a software package. These steps were delineated into the four phases of collecting the data, reducing the data prior to analysis, analysing data, and interpreting the data.

Within each phase, the aspects arising during the evaluation were considered, using a modified version of a paired comparison approach used by Blum and Naylor (1968). The process consisted of considering each pair of evaluation methods in turn, and indicating which was more easy to use, with regards to its practicality on the attribute being considered. This resulted in a paired comparison matrix allowing for the proportion of times each evaluation method was preferred over others to be calculated.

The average proportion for each phase of the evaluation process could then be calculated and transformed into a standard score (z score). The resulting z scores were the scale values for the evaluation method. Blum and Naylor (1968) reported that the advantage of this procedure was that these scale values may be taken as representing not only the rank order of the evaluations, but also the degree to which two evaluation methods differ with regards to their practicality on the attributes.

Furthermore, by an appropriate mathematical manipulation (adding the required constant), these scores were transformed into a standardised continuum of merit.

5.9. Method.

5.9.1. Subjects.

A total of 148 individuals of mixed age and both genders took part in the study. In all, 54 individuals evaluated the spreadsheet, 48 the word processor, and 46 the database. Of these, 40 individuals used the logged data evaluation method, 39 the evaluation questionnaire, 37 the interview, and 32 the verbal protocol analysis method. Table 5.4 shows the cell sample sizes and the breakdown within the software domain evaluated, and the evaluation method used. All cells are independent, with no

Table 5.4. Cell Sample Sizes Used in the Study.

	Evaluation method				
Software	Logged data	Questionnaire	Interview	Verbal protocol	TOTAL
Spreadsheet	15	13	14	12	54
Word Processor	11	14	13	10	48
Database	14	12	10	10	46
TOTAL	40	39	37	32	148

individual appearing in more than one cell.

Subjects were recruited from a first year psychology course. The course is a general course in psychology and is used as both a service course to advanced psychology papers and as a filler for degree students of mixed academic background.

5.9.2. Materials.

There was a degree of standardisation among the three software types evaluated, resulting in a core set of materials being used. These were supplemented, where necessary, by material specific to each software evaluation method.

The core material consisted of:

1. One informed consent form (see Appendix 2).
2. One overall usability rating form (see Appendix 15).
3. Video recording equipment, including cameras.
4. IBM compatible computers with colour graphics.
5. A colour video recorder.

6. A colour television.
7. An audio tape recorder.

The specific material consisted of the following items:

1. One introductory information sheet that introduced the subject to the type of software they were going to use, the instructions and the task (see Appendices 3 to 9 for the introductory passages and tasks used in the study.)

2. One outline diagram of the menu options for the package subjects were to use (see Appendices 10 to 12 for the diagrams used in the study.)

5.9.3. Procedure.

For all three components of the study the procedure was the same. Prospective subjects were first contacted by telephone (see Appendix 1 for the telephone contact procedure used). Here an enquiry was made about whether the individual would take part in the study. Upon agreement, it was then ascertained if they had ever used a computer. Next, a mutually suitable time to conduct the study was arranged.

Upon arrival at the experimental centre, subjects were allocated to one of the four experimental conditions by a random process. Subjects were informed about the purpose of the study and that they could not be told which experimental group they were in until after the experimental task had been completed, so as to prevent demand characteristics. At this point, each subject had the opportunity to ask questions. After questions had been answered, the subjects were asked to complete an informed consent form (see Appendix 2.)

Subjects were then given the introductory passage that covered the background concepts of the software they were about to use, along with a brief demonstration of the piece of software involved. During this demonstration the concept of a "cursor" was explained. Also, the ideas of the "cell" and "field" were elaborated on in the spreadsheet and the database respectively.

Subjects were shown how to move the cursor, and how to activate the menu system of the software they were to use. At this stage subjects were allowed to ask questions. Next, it was explicitly stated that the aim of the study was to test the software, not the individual, and if they had problems, they should attract the researcher's attention and questions would be answered. Furthermore, due to ethical considerations, they could stop at any time, if they felt they did not wish to continue.

Prior to commencing the research, subjects were asked to express the degree of confidence they had in using the major types of business computer software. Specifically, subjects were required to rate how confident they would feel using a spreadsheet, a word processor, a database, and using computers in general. Next, subjects individually worked through the experimental task. During the entire session all subjects were videotaped, and all instances of help were recorded.

After completion of the study, and prior to being informed which evaluation group they were in, subjects were asked to rate, on a one to ten scale, how usable they felt the package was. Subjects were then informed about the experimental group to which they had been allocated.

Subjects in the logged data group were free to leave, but subjects in the questionnaire group were asked to complete the questionnaire. Subjects in the interview and the verbal protocol groups were asked to either stay and complete the study, or make an appointment within the next two days to complete the study. Each subject was individually debriefed following the collection of all data.

Chapter Six: The Evaluations and Discussion of Each Single Evaluation Method.

6.1. Equivalence of the Groups.

It was initially hypothesised that no difference in perceived confidence, when using computers, would be found between the experimental groups. To examine this hypothesis, the means and standard deviations of the subjects' confidence ratings when using computer programmes were calculated, as were the means and standard deviations of the subjects' overall usability rating of the software. Table 6.1 presents the cell means, standard deviations, and sample sizes for these ratings. It can be noted that the mean confidence ratings of the individuals in the word processor group are lowest.

To test the equivalence of each group, ANOVAs were used to examine the variability of the pre-experimental confidence ratings, and the post-experimental overall usability score of the software evaluated. All ANOVA results and probability levels are presented in Table 6.2.

The results showed that effects were present for perceived confidence when using word processors and databases. Also, there was an effect for the usability of the software. This suggests that at least one of the groups of subjects differed with regard to their perceived ability to use a word processor and a database, prior to commencing the study. Also, at least one of the packages evaluated was perceived as being more usable than the others.

The second series of ANOVAs examined the variabilitywithin each of the evaluation method groups. In all cases the results were non-significant.

Table 6.1. Summary of Group Means, Standard Deviations, and Sample Sizes of the Subjects' Confidence Ratings and Overall Usability, for the Three Software Packages Evaluated. Subjects Rated How Confident They Were at Using Specific Software Types Prior to Commencing the Study, Whereas the Usability Rating was Made Immediately After Completion of the Experimental Task and Related Specifically to the Software They had Used.

Key. SS = Spreadsheet
WP = Word processor
DB = Database
OC = Overall confidence
User = Usability rating

Spreadsheet evaluation group

Confidence and Usability ratings

	SS	WP	DB	OC	User
Mean	4.12	3.57	2.45	3.94	6.98
(Std.)	1.95	1.71	1.46	1.81	1.93
n	49	49	49	49	54

Word processing evaluation group

Confidence and Usability ratings

	SS	WP	DB	OC	User
Mean	3.36	2.52	2.00	3.39	7.48
(Std.)	1.71	1.62	1.22	1.62	1.77
n	44	44	44	44	44

Database evaluation group

Confidence and Usability ratings

	SS	WP	DB	OC	User
Mean	3.84	2.91	2.84	3.73	6.11
(Std.)	1.72	1.74	1.82	1.70	1.98
n	45	45	45	45	47

* Note: Sample sizes may not equal the experimental cell sizes due to technical problems e.g., programme crash, turning system off to finish the session rather than saving the data, disk contamination, etc.

Table 6.2. Summary of ANOVAs Used to Test the Subjects' Confidence Ratings and Usability Scores, by Application Domain, and by Evaluation Group Assigned.

Key df = Degrees of Freedom.
BG = Between Groups.
WG = Within Groups.

By software application domain

	df BG	df WG	F-Ratio	F-Prob.
Spreadsheet	2	135	2.08	.129
Word Processor	2	135	4.59	.012*
Database	2	135	3.44	.035*
Overall Confidence	2	135	1.22	.298
Usability Rating	2	135	6.13	.003*

* = p <.05.

By evaluation group

	df BG	df WG	F-Ratio	F-Prob.
Spreadsheet	3	134	0.06	.980
Word Processor	3	134	0.16	.921
Database	3	134	0.03	.995
Overall Confidence	3	134	0.04	.991
Usability Rating	3	134	1.15	.332

These statistically significant findings were unexpected. A randomisation process was used which should have resulted in between groups equivalence.

No statisticallysignificant differences were obtained in confidence estimates between those individuals allocated to the logged data evaluation method, the questionnaire evaluation method, the interview evaluation method and the verbal protocol analysis evaluation method. This suggests that comparisons between the results from the evaluation methods can be made.

As might be expected differences also existed with regards to the estimates of usabilityof the three programmes evaluated. The group means were 7.48 for the word processor, 6.98 for the spreadsheet and 6.11 for the database group.

Examination of the perceptual, between groups, differences suggested a gender difference may have been the cause. Table 6.3 presents the cell means, standard deviations, and sample sizes in terms of a gender based breakdown. At this time it was also decided to examine a series of performance statistics, to test if these differences resulted in hitherto unforseen problems. These performance statistics consisted of: total time to perform the experimental task, number of keys used to perform the task, number of times help was requested, and the usabilityrating the subject gave the package they were required to evaluate. From Table 6.3 it can be seen that, with regard to confidence ratings in the spreadsheet and database tasks, there appears to be little difference between male and female ratings.

This is not the case for the word processing group, however, where quite large differences in mean ratings were found. In particular, the females' mean ratings are lower than the males' mean ratings. It can also be noted that in all of the performance measures, and the usabilityratings, there appears little difference between male and female means and standard deviations.

Table 6.3. Means, Standard Deviations and Sample Sizes of the Three Sub-Samples When a Gender Based Breakdown Is Undertaken.

Key: SS = Spreadsheet. WP = Word Processor.
DB = Database. OC = Overall Confidence
Time = Time to complete task.
Keys = Number of keys used to complete task.
Help = Number of times the subject requested help.
User = Usability rating.

	Confidence ratings				Performance scores			
Spreadsheet								
Female								
	SS	WP	DB	OC	Time	Keys	Help	User
Mean	3.84	3.44	2.16	3.64	23.59	863.92	0.58	6.75
(Std.)	1.91	1.78	1.14	1.78	6.53	170.24	0.76	2.10
n	25	25	25	25	22	24	24	25
Male								
	SS	WP	DB	OC	Time	Keys	Help	User
Mean	4.42	3.71	2.57	4.25	23.64	884.57	0.25	7.23
(Std.)	1.99	1.68	1.70	1.82	6.89	205.83	0.64	1.72
n	24	24	24	24	21	21	20	24
Word Processor								
Female								
	SS	WP	DB	OC	Time	Keys	Help	User
Mean	2.94	2.25	1.66	2.91	31.79	1977.76	1.13	6.36
(Std.)	1.61	1.48	0.94	1.51	8.67	372.81	1.01	2.13
n	32	32	32	32	28	29	30	32
Male								
	SS	WP	DB	OC	Time	Keys	Help	User
Mean	4.50	3.25	2.92	4.67	26.57	1783.25	1.29	5.64
(Std.)	1.51	1.82	1.44	1.16	11.48	695.85	1.11	1.55
n	12	12	12	12	8	8	7	14
Database								
Female								
	SS	WP	DB	OC	Time	Keys	Help	User
Mean	3.80	2.60	2.80	3.67	22.18	801.35	0.96	7.35
(Std.)	1.75	1.57	1.86	1.67	7.44	204.54	1.23	1.93
n	30	30	30	30	23	26	24	29
Male								
	SS	WP	DB	OC	Time	Keys	Help	User
Mean	3.93	3.53	2.93	3.87	26.36	909.08	1.00	7.73
(Std.)	1.71	1.96	1.79	1.81	7.61	255.37	1.61	1.44
n	15	15	15	15	10	12	11	15

NB: Sample sizes differ because all performance data were not available due to technical difficulties.

Table 6.4 presents the results of a series of one-way ANOVAs. It should be noted that with regards to the confidence data in the spreadsheet and database groups, no gender differences were found. However, in the

Table 6.4. Summary of One Way ANOVAs Used to Test the Differences Between the Female and Male Self Perceived Confidence Scores and Performance Scores for the Three Sub-Samples.

Spreadsheet evaluation group

	df BG	df WG	F-Ratio	Prob.
Spreadsheet	1	47	1.07	.3067
Word Processor	1	47	0.30	.5878
Database	1	47	2.05	.1591
Overall confidence	1	47	1.41	.2416
Time to do task	1	41	0.00	.9795
Number of keys used	1	43	0.14	.7144
Requested help	1	42	2.36	.1321
Usability rating	1	47	0.84	.3648

Word Processor evaluation group

	df BG	df WG	F-Ratio	Prob.
Spreadsheet	1	42	8.53	.0056 *
Word Processor	1	42	3.52	.0677
Database	1	42	11.61	.0015 *
Overall confidence	1	42	13.30	.0007 *
Time to do task	1	34	1.96	.1708
Number of keys used	1	35	1.14	.2929
Requested help	1	35	0.13	.7258
Usability rating	1	44	1.09	.3011

Database evaluation group

	df BG	df WG	F-Ratio	Prob.
Spreadsheet	1	43	0.06	.8094
Word Processor	1	43	2.10	.0905
Database	1	43	0.05	.8199
Overall confidence	1	43	0.14	.7141
Time to do task	1	31	2.18	.1501
Number of keys used	1	36	1.95	.1716
Requested help	1	33	0.01	.9334
Usability rating	1	42	0.47	.4970

* p<.05

NB: Degrees of freedom alter due to technical difficulties.

word processor task, three statistically significant differences were observed (p <.05: see Table 6.4).

When examining both the performance data and the usability ratings in Table 6.4, it is important to note that in all cases no gender differences were found. From these results, it appears that the randomisation process was insufficient to assure total group equivalence.

Of more importance, is the finding that no performance differences were found. That is, where gender perceptual differences were observed, gender based performance differences were not observed. This suggests that the validity of the study have not been undermined by this unexpected result. The gender variable is therefore an extraneous variable, as far as the comparisons of the software evaluations in this research are concerned. More precisely, no gender differences in the logged data, questionnaire, interview, and verbal protocol analyses sub-groups are present. Also, the ratio of male to females will not affect the overall usability rating.

However, difference in self perceived confidence is interesting, as it lends itself to the notion that such estimates of confidence were unfounded. This has implications for both the educational and computing fields. Specifically, as confidence seems related to experience (Howard and Smith, 1986) it could be argued that all children should be exposed at some stage to the use of computers. Furthermore, with regards to the computing industry, situational approaches to personnel selection for the industry should be used because the present study showed that performance estimates do not exhibit a gender bias, whereas perceptions did.

6.2. Specific Problems Highlighted by Each Evaluation Method.

Tables 6.5, 6.6 and 6.7 present the usability problems found in each of the software packages evaluated, and which evaluation method(s) found each problem area. By examining these tables, it can be seen that sometimes much overlap between the methods and the problems found is evident, with some problems found by all evaluation methods. Such overlap could reflect convergence of results, suggesting a form of concurrent validity. Conversely, if more than one of these evaluation methods is used, duplicated, redundant

Table 6.5. Problems Identified with the User Interface of the Spreadsheet Package and the Evaluation Method Identifying the Problem.

Key: LD = Logged data. I = Interview.
Q = Questionnaire. VP = Verbal protocol.

PROBLEM	METHOD			
	LD	I	Q	VP
ENTER				
Cursor during data entry.	●	●	●	●
Use of numeric key pad.		●	●	●
Input appearing at the bottom of the screen.		●		●
Target cell identification.		●		
Cursor positioning.		●		
Cell editing facility.		●		
Not sure of cursor movement.				●
Problem with input/ command mode.	●		●	●
Problem when output scrolled.				●
Input string was longer than one cell.			●	●
Wished to enter whole line at once.				●
Cursor control.			●	
SAVE				
Problem completing the save sequence.	●	●	●	
Retyping file name each time to save.	●	●		●
Feedback from save.	●			●
Activating main menu.	●	●		●
Navigation in menu system.		●	●	●
Saved under more than one name.				
General problem.		●		
PRINT				
Problem completing command sequence.	●	●	●	●
Problem using the menu system.		●		●
Error in programme code appeared.				●
LOAD				
Mistyped file name/ could not find file.	●			
Apprehension when loading.				●
INSERT				
Place cursor in wrong place.	●	●		●
Inserted row instead of column.				●
DELETE				
Problem positioning cursor for delete.		●	●	●
Problem using block commands.	●			●
Using menu system.	●	●		
Couldn't specify limit of delete.		●		●
Deleted whole row rather than intended cell only.				●
Deleted one cell at a time.	●		●	●
Difficult to deal with rows and cells concepts.				●
OTHER				
Error messages inappropriate.			●	
Poor menu lay out.			●	
Poor prompts.			●	
Key options not eye catching.			●	
No way of reducing input/output according to user experience.	●		●	
Total Usability Problems Identified = 38	13	17	15	24

Table 6.6. Problems Identified with the User Interface of the Word Processing Package and the Evaluation Method Identifying the Problem.

Key: LD = Logged data. Q = Questionnaire. I = Interview. VP = Verbal protocol.

PROBLEM	METHOD			
ENTER	LD	Q	I	VP
Typing skill problems.	●	●	●	●
Knowledge of keyboard functions.		●		●
Relocation of altered text.		●		●
Cursor control problems.				●
Text colour.			●	●
Text formatting.				●
Positioning of cursor after printing.				●
SAVE				
Saving when overwrite required.	●		●	
Lack of task feedback.	●	●	●	●
Navigation problems.	●		●	
Problem completing save sequence.				●
PRINT				
Problem saving before printing.	●	●	●	●
Problem with save error message.	●	●	●	●
Navigation problems.	●		●	
Lack of task feedback.		●		●
Problem completing print sequence.				●
DELETE				
Problem using block commands.	●	●	●	●
Problem with block command names.				●
Problem with recovery of deleted text.		●	●	●
Problem with connecting altered text.		●	●	●
INSERT				
Problem with help instructions.			●	●
Poor command names.				●
Positioning of cursor when inserting text.			●	
OPEN/CLOSE				
Problem with more than one file open.	●		●	●
Menu cursor jumping protected options.				●
Lack of task feedback.			●	●
OTHER				
Search from cursor rather than top.			●	
Menu headings.		●		●
Command names.			●	
Poor error correction system.		●		
Poor help screen instructions.		●		●
Navigation problems.	●			●
No reduction in entry for experience.		●		
No alternate data entry systems.		●		
Would be hard to relearn.				
Total Usability Problems Identified = 35	10	15	17	25

Table 6.7. Problems Identified with the User Interface of the Database Package and the Evaluation Method Identifying the Problem.

Key:	LD = Logged data.	I = Interview.
	Q = Questionnaire.	VP = Verbal protocol.

PROBLEM	METHOD			
ENTER	LD	Q	I	VP
Typing skill problems.			●	●
Exiting add option.	●		●	●
Not saving before exit.	●	●		●
Multiple saves.	●		●	
Adding blank record.	●			
Cursor control.	●	●		●
SEARCH				
Confusing input structure.		●	●	●
No multiple search structure.			●	
Used scroll technique.	●	●	●	●
Order of entry.			●	
Poor entry system.				●
Cursor control.	●		●	●
LIST/PRINT				
Lack of feedback.	●	●	●	●
Navigation problems.			●	●
Incorrect option selection.	●		●	●
System failure.		●		●
UPDATE				
Cursor position.				●
Not saving after update.	●		●	
Feedback problems.	●		●	
Problem exiting sub-system.	●			
Entering altered information.			●	
DELETE				
Accidental deletion of record.	●	●	●	●
Non-completion of sequence.	●			
OTHER				
Menu names.		●		●
Menu navigation.	●	●	●	●
No help option.		●		●
Positioning of menu items.		●		●
Poor error messages.		●		
Poor colour.		●		●
Poor menu lay out.		●		
Unintentional option activation.	●			
Total Usability Problems Identified = 31	16	14	16	19

information would be obtained.

6.3. The Logged Data Method.

Theoretically, a reduction in mean and variance over separate occasions of performing the task would suggest an improvement in performance, and should therefore be related to the usability of software. To examine these attributes of the software, the mean times taken to undertake each sub-task were isolated. These times were then subjected to correlated t-tests, and reduction in variance t-tests. Tables 6.8 to 6.10 represent a summary of these tests. It should be noted that changes in mean and variance are common, and that they are not all in the direction that would suggest an increase in performance. In some cases the change denotes an increase in the mean, or variance, between two occasions of performing a sub-task, suggesting a decrease in performance.

The concept of efficiency also was to have been examined. Here efficiency was to be operationalised as the minimum number of keystrokes needed to complete the task divided by the actual number of keystrokes used to complete a task. However, problems were encountered with this proposed analysis. First, within the spreadsheet package a structured menu system was used. This resulted in a high efficiency score which was not useful as a discriminating tool. Such scores were also confusing because errors seemed to be mainly typing errors. This essentially confounded the efficiency score. The efficiency scores could have been used, provided that typing errors were partialled out. However, this would have involved further transformations of the data, and subjective classification, and therefore may have reduced the validity of such scores.

Within the word processing package other problems developed when using the efficiency scores. In the word processor there were multiple ways of

Table 6.8. Summary t-test Values for a Correlated Mean t-test and a Reduction of Variance t-test for the Three Instances of the Spreadsheet Sub-Tasks.

Key:

occasion:	Mean t value	df.	Sig.
	Variance t value	df.	Sig.

* Two tailed test with alpha set at p<.05

Task: Print.

	t value	df.	Sig.
1.2	0.81	13	
	0.44	12	
1.3	3.91	11	*
	7.57	10	*
2.3	2.79	11	*
	3.66	10	*

Task: Load.

	t value	df.	Sig.
1.2	1.64	12	
	0.73	11	
1.3	2.06	11	
	1.51	10	
2.3	1.23	11	
2.3	1.31	10	

Task: Clear.

	t value	df.	Sig.
1.2	4.44	12	*
	3.89	11	*
1.3	1.56	11	
	-0.04	10	
2.3	-0.44	11	
	-1.92	10	

Task: Delete.

	t value	df.	Sig.
1.2	0.94	12	
	11.28	11	*
1.3	0.84	6	
	8.04	5	*
2.3	-0.83	6	
	-1.43	5	

Task: Insert.

	t value	df.	Sig.
1.2	2.05	9	
	0.66	8	
1.3	1.77	4	
	0.17	3	
2.3	0.75	4	
	0.81	3	

Task: Save.

	t value	df.	Sig.
1.2	3.63	13	*
	1.48	12	
1.3	3.35	12	*
	4.09	11	*
2.3	0.50	12	
	2.36	9	*

- Note the degrees of freedom alter as some individuals did not complete all specified sub-tasks.

Table 6.9. Summary t-test Values for a Correlated Mean t-test and a Reduction of Variance t-test for the Three Instances of the Word Processor Sub-Tasks.

Key:

Occasion:	Mean t value Variance t value	df. df.	Sig. Sig.

* Two tailed test with alpha set at p<.05

Task: Print.

	t value	df.	Sig.
1.2	2.83	10	
	-0.82	9	
1.3	5.32	10	*
	4.25	9	*
2.3	1.08	10	
	10.36	9	*

Task: Save.

	t value	df.	Sig.
1.2	5.32	10	*
	3.01	9	*
1.3	5.44	10	*
	3.89	9	*
2.3	0.20	10	
	0.52	9	

Task: Open.

	t value	df.	Sig.
1.2	1.65	8	
	3.58	7	*
1.3	1.65	7	
	2.66	6	*
2.3	-0.65	7	
	-1.78	6	

Task: Close.

	t value	df.	Sig.
1.2	-2.98	9	
	-0.98	8	
1.3	-4.84	7	*
	-1.02	8	
2.3	-0.51	7	
	1.86	6	

● Note the degrees of freedom alter as some individuals did not complete all specified sub-tasks.

Table 6.10. Summary t-test Values for a Correlated Mean t-test and a Reduction of Variance t-test for the Three Instances of the Database Sub-Tasks.

Key:

occasion:	Mean t value	df.	Sig.
	Variance t value	df.	Sig.

* Two tailed test with alpha set at p<.05

Task: Enter.

	t value	df.	Sig.
1.2	4.23	13	*
	4.69	12	*
1.3	2.83	13	*
	1.77	12	
2.4	-4.12	13	*
	-6.40	12	*

Task: Print.

	t value	df.	Sig.
1.2	0.71	13	
	0.01	12	
1.3	2.01	9	
	2.49	8	*
2.3	0.28	9	
	2.06	8	

Task: Update.

	t value	df.	Sig.
1.2	0.58	12	
	2.40	11	*
1.3	1.56	9	
	-0.98	8	
2.3	1.34	9	
	3.60	8	*

Task: Delete.

	t value	df.	Sig.
1.2	0.93	12	
	1.67	11	
1.3	0.91	9	
	3.01	8	*
2.3	-0.34	9	
	0.97	8	

• Note the degrees of freedom alter as some individuals did not complete all specified sub-tasks.

achieving a desired objective, resulting in confusing and ambiguous scores, with sometimes the more efficient method taking longer than the less efficient one. The approach seemed to work better when used within the database package where efficiency scores seemed more valid. However, as comparisons between the software packages were not valid the method was dropped from the analysis. Measures of efficiency taken in this way may not be very useful and may be interface dependent.

The logged data were also examined in a contextual fashion. This involved a playback approach, where the evaluator examined the interaction by running the logged data back through the computer, and thus saw both the key strokes, and the resulting screen output for these interactions. When this was done, 13 interaction problems were identified with the spreadsheet interface, 10 with the word processor interface and 16 with the database interface. These problems are presented in Tables 6.5 to 6.7 (pp. 116 -118).

The use of automated data acquisition routines tends to add credence to Penniman and Dominick's (1980) statement that this approach is perhaps the easiest way of obtaining an objective record of user behaviour, and must be seen as the epitome of objective data acquisition. As noted by Drury (1987), the approach proved to be unobtrusive and, technical difficulties excluded, is a relatively inexpensive, accurate and reliable form of data collection. It should be noted that the method used, encompassed the objective interactions of the user, and no attempt was made to record subjective user impressions of the software usability characteristics. However, in accordance with Penniman and Dominick's (1980) report, such measures could be recorded.

However, if technical difficulties do arise when using the software oriented logged data approach, the total data file, and therefore all data, are usually lost. This problem was encountered during the course of the present study where the data acquisition routine was designed to save the data on the request to exit the experimental software. However, on more than one occasion the user exited the programme by switching off the computer, resulting in a total loss of data. Care should therefore be taken when developing such software oriented logged data acquisition approaches. Such problems may not be so prominent with the on-line tap (Neal and Simons, 1984a, 1984b) and the video collection methods as outlined by Kirakowski and Corbett (1990).

Table 6.11 presents a short transcript of a log obtained when using the logged data method. This particular log is from the spreadsheet package. It should be noted that the number of keystrokes activated for the experimental task used in this study was about 1100. When considering that the experimental task on average took little more than 20 minutes, the complexity of the analysis process, and the sheer size of the data set generated, when using this form of evaluation is clear.

After obtaining the raw data, the evaluator's task becomes one of data reduction, analysis, and interpretation. The major form of data reduction used, was the isolation of target key sequences, that indicated specific sub-tasks of interest. These generally consisted of the times to do basic input-output operations, over each of three separate occasions. The identification of such key sequences was problematic and time consuming (see Table 6.11). Data preparation routines may help, however. This possibility has been raised by Yoder et al. (1985) and Laws and Barber (1989), and should be considered before deciding whether to use the logged data approach in this form.

Having identified and isolated the target interaction strings, times were calculated and comparisons made by subjecting these data to correlated t-tests and reduction of variances t-tests. When this was done, it was found that data used in this form were generally ambiguous and sometimes misleading. In particular, it was often difficult to discern trends because there were sometimes significant reductions in means accompanied by significant increases in variances and vice-versa. This problem, and the potential validity problems associated with performance data (Eason, 1984; Kondakci, 1985; MacLean et al., 1985; Yamagishi and Azuma, 1987) suggest that logged data used actuarially should not be used on their own.

Table 6.11. Short Transcript and Interpretations of an Obtained Log When Using the Logged Data Collection Process.

Time	Key		Interpretation
9 19 47 54	47	/	- activate main menu
9 19 48 86	115	s	- choose spreadsheet option
9 19 49 96	115	s	- choose save option
9 19 53 31	80	P	- enter name of the file to be saved. In this case the user decided to use all capital letters after initially entering two lower lower case. This resulted in increase in total time and a decrease in efficiency.
9 19 54 18	105	i	
9 19 54 57	101	e	
9 19 57 64	8	Back space	
9 19 57 86	8	Back space	
9 19 58 96	73	I	
9 19 59 35	69	E	
9 19 59 73	83	S	
9 20 0 23	76	L	
9 20 0 61	69	E	
9 20 0 83	83	S	
9 20 3 19	13	Enter	- enter the file name
9 20 8 41	121	y	- answer yes to overwrite the previous existing file of this name.
			Time = 20.87 seconds

The logged data were also examined in a contextual way. This involved a playback of the subject's interaction with the computer, where the evaluator examined both the key strokes, and the resulting screen output. The evaluator followed the interaction using the experimental instruction sheet. As highlighted by Yoder et al. (1985) much effort can be expended in this sort of post data collection processing, prior to the actual data analysis.

When using the contextual form of analysis, problem areas were more easily identified than with the actuarial method. Problems were often discovered by going over and over particular sequences of interaction, which highlighted the subject's difficulties. This approach is quite subjective, however, because the model of the user's behaviour used by the evaluator is initiated and guided by the sequence of the experimental task, and his or her knowledge of the system being evaluated. It is assumed that the user is also using this same experimental task to guide their behaviour, which may or may not be the case. For example, on one occasion the user entirely missed out one step of the experimental task (presumably by misreading the point they were up to). This resulted in a different task sequence being used by the user to that dictated by the task instruction from that point on. This example is just one of many that could have been given; other problems may have existed that also seriously compromised data interpretation. For instance, a user might have been experimenting with the software and this may have been interpreted as an error. Such instances may have been misinterpreted by the evaluator, resulting in incorrect evaluation decisions being made. One way potentially to solve these problems is to have the user present during the data interpretation phase of the evaluation.

Thus, the interpretation of logged data is subject to ambiguity, and used alone, will suffer reliabilityand validity problems. It would seem that Yoder et al.'s (1985) statement that,

> "analyzing logged data is akin to archaeology as behaviour must be inferred from low level data artifacts" (Yoder et al., 1985, p.907).

has been supported in this study.

It is essential that studies are conducted into the validity of the inferences made by judges examining logged data in a non-actuarial way. This could be undertaken by having an evaluator analyse the data and then have the user also work through the task explaining their actions. To do this form of study an adequate memory aid for the user would be needed. Perhaps a video recording of the interaction is the most practical way of achieving this. Such studies would need careful design because of the possibilityof interactions with such factors as task complexity, user experience, and the time between the study and the evaluation.

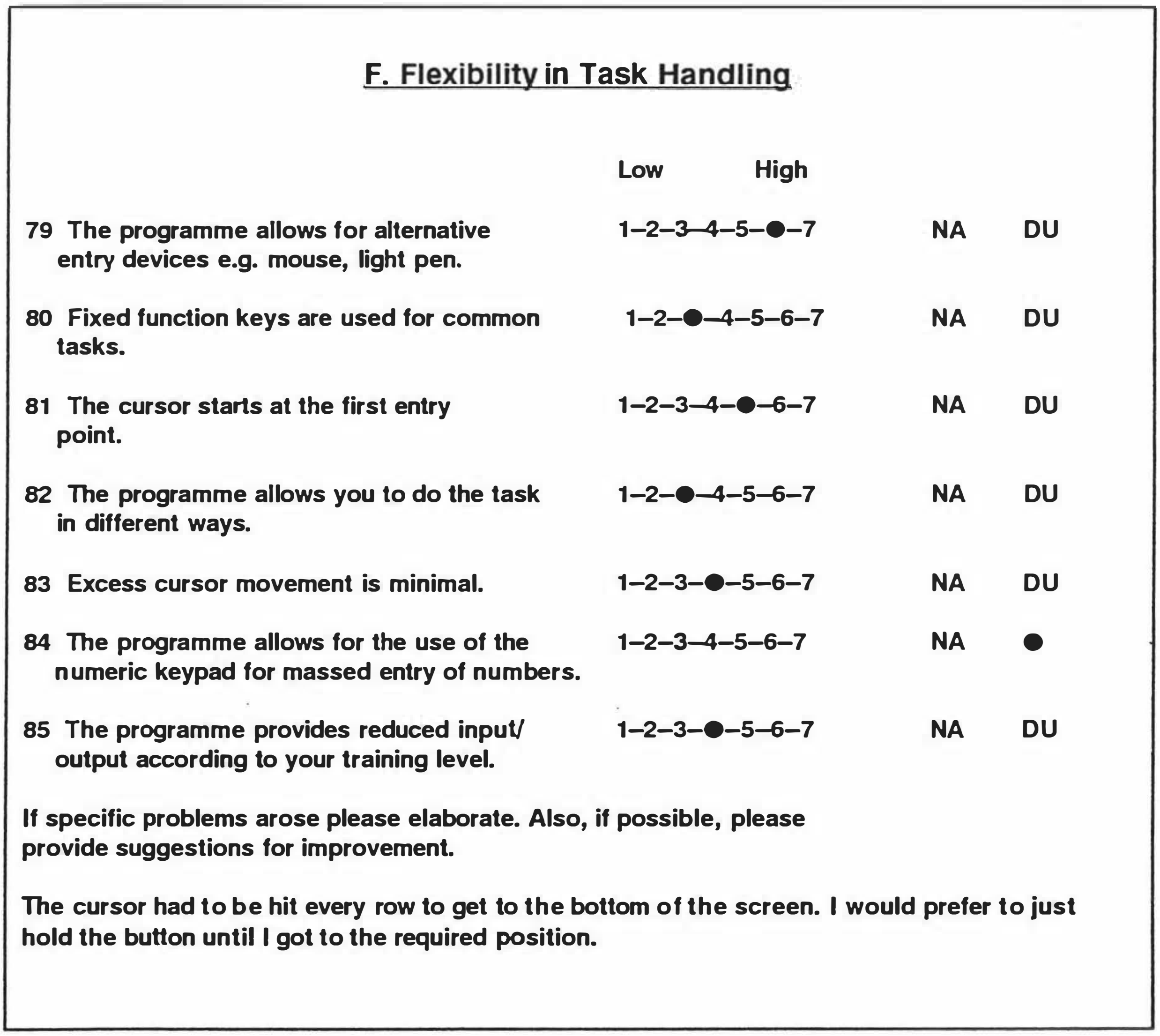
One way of dealing with the ambiguity of problems with the software is by using large sample sizes. However, this would result in large data sets which would be time consuming to analyse, causing the evaluation procedure to be unwieldy and impractical compared to other evaluation procedures. Even with comparatively small samples the use of the logged data method results in the generation of large amounts of data. This is a practical disadvantage, especially if the method has to be used repeatedly.

In summary, when using the logged data alone, this research supports many of the assertions made by past researchers, and therefore has played a confirmatory role. The method is unobtrusive (Penniman and Dominick, 1980) and inexpensive (Drury, 1987). The present research supports the assertions made by Yoder et al. (1985) with regards to their suggestions for using this approach in that evaluators should develop the statistics over time, record intermediate statistics over time, and expect to devote much effort prior to processing data. The research also supports the strengths, as highlighted by Lea (1988), in that the method collects accurate, reliable data and is unobtrusive, and the suggestions made by Laws and Barber (1989) in that pre-specified grammars of actions are required to help with the extraction of meaningful data, before any analysable data can be obtained. Concerns have also been expressed about the validity of some of the operationalisations as highlighted by Eason (1984) and Yamagishi and Azuma (1987).

6.4. The Questionnaire Method.

Table 6.12 reproduces part of a transcript of a completed questionnaire. It can be seen that the ratings do not draw the attention of the evaluator to a particular problem area, but to a more global problem. In contrast, the open ended comments tend to be more directive and specific.

Table 6.12. Transcript of a part of a Completed Evaluation Questionnaire.



F. Flexibility in Task Handling

	Low High		
79 The programme allows for alternative entry devices e.g. mouse, light pen.	1–2–3–4–5–●–7	NA	DU
80 Fixed function keys are used for common tasks.	1–2–●–4–5–6–7	NA	DU
81 The cursor starts at the first entry point.	1–2–3–4–●–6–7	NA	DU
82 The programme allows you to do the task in different ways.	1–2–●–4–5–6–7	NA	DU
83 Excess cursor movement is minimal.	1–2–3–●–5–6–7	NA	DU
84 The programme allows for the use of the numeric keypad for massed entry of numbers.	1–2–3–4–5–6–7	NA	●
85 The programme provides reduced input/ output according to your training level.	1–2–3–●–5–6–7	NA	DU

If specific problems arose please elaborate. Also, if possible, please provide suggestions for improvement.

The cursor had to be hit every row to get to the bottom of the screen. I would prefer to just hold the button until I got to the required position.

The questionnaire also elicited a large amount of data. The means and standard deviations of usability attributes were first calculated and appear in Appendix 16. The attributes that had a poor rating (a mean of 2.5 or below) were isolated and appear in Table 6.13. It should be noted that they do not directly refer the evaluator to identifiable problem areas, but rather to a general problem. This made the isolation of problem areas difficult.

Table 6.13. Attributes Rated as Low Using the Questionnaire.

Key: SS = Spreadsheet.
WP = Word processor.
DB = Database.

Attribute:	SS	WP	DB
The presentation of what the programme can do is clearly arranged.	●	●	
The programme provides a list of abbreviations.		●	
Moving between menus is easy.		●	●
Error messages are appropriate.		●	●
Relearning after intermittent use would be easy.			●
Terminology closely relates to the task area.		●	●
The next screen in a sequence is predictable.		●	
Terminology is consistent.		●	
The programme allows for alternative entry devices eg. mouse, light pen.		●	
The programme provides reduced input/ output according to your training.	●	●	
Context sensitive help is provided.		●	●
Error correction is at the point at where it occurred.		●	
Auditory signals are used appropriately.		●	
Error messages contain correction hints.			●
The programme provides messages with different levels of detail dependent on your experience.	●	●	●
Percentage of questions rated poor.	5.13%	10.26%	4.27%

When examining the attributes that were rated as "high," problems emerged. In particular, it was found that, sometimes, some attributes were rated "high" yet were not present in the interface. For example, the use of alternate entry devices was rated as "high" in the database when no such devices were set up. The correct rating should have been "Not Applicable." This raises the problem of the validity of this particular rating.

In general, the number of attributes rated "low" were small compared to the total number of questions asked (see Table 6.13). This suggests that either the interface did not cause adverse reactions, or perhaps some response acquiescence on the part of the subjects was present.

The problems identified by the questionnaire, using both the criterion of a mean below 2.5 and a content analysis of the open ended statements, appears in Tables 6.5, 6.6 and 6.7 (pp. 116 - 118). The open ended comments tended to be more specific, and of more use when using this sort of evaluation process. Further problems concerning the potential validity of the elicited ratings arose at this point. Sometimes the ratings appeared confusing or contradicted the open ended statements. For example, the "Informative feed-back is appropriate" was rated as high in the database group, yet subjects reported that they accidentally deleted multiple records, as there was no feedback on task completion.

Unfortunately, this was not an isolated incident, with confusing and contradictory mean ratings appearing on several occasions. Again, the data gave rise for concern over the validity of the questionnaire method.

To address the topic of internal reliability of the questionnaire, Cronbach's coefficient alpha for each sub-scale of the questionnaire was computed. Table 6.14 presents these results. It should be noted that factors 3, 4, 5 and 7 meet the internal reliability criterion of .60, as suggested by Nunnally (1967). Overall, the calculations suggest that within the factors the questions are tapping the same construct. Factors 1,2 and 8 meet this criterion on two of the three calculations. This suggests that although the measure proved reasonably internally consistent, factors 1,2,8, and especially 6, require further development.

The problem of how well the subjects understood the questionnaire was also addressed by calculating the proportions of individuals who reported that they did not understand a particular question. These results appear as Figures 6.1 to 6.4 It should be noted that sometimes a large proportion of individuals did not understand the questions asked. This poses problems because in Fiske's (1982) terms there was a communication problem with a number of the questions. This has implications for the validity of the results. To be of use for user based evaluations the questions must be comprehensible to individuals who are not computer professionals.

Analysis of the response category "Not Applicable" was also undertaken. Here it was found that some variance existed concerning whether the question was regarded as applicable to the interface the subject had used or not. This again raises problems regarding the validity of these results.

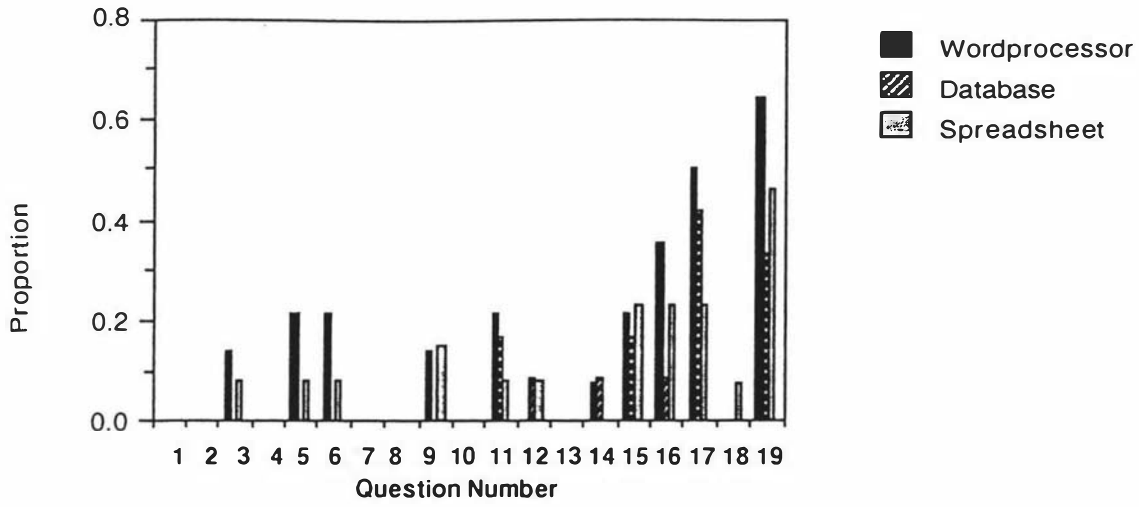
Table 6.14. Cronbach's Alpha Reliability Coefficients for the Eight Factors of the Questionnaire Used in the Study.

Factor

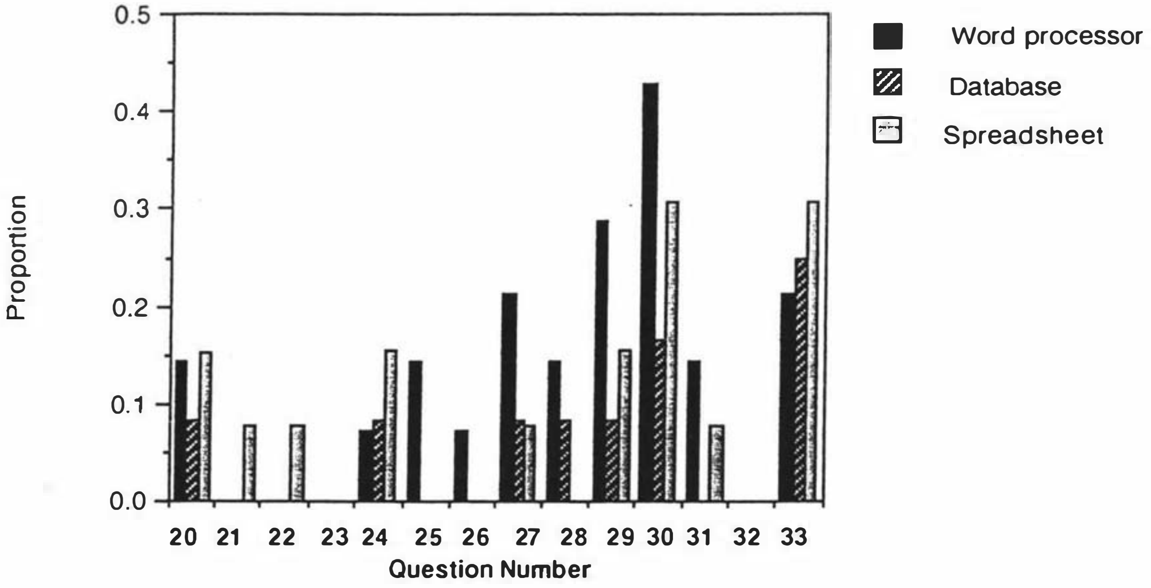
Programme	1	2	3	4	5	6	7	8
Spreadsheet	.719	.206	.832	.617	.686	.594	.864	.358
Word Processor	.059	.669	.813	.774	.715	.425	.681	.762
Database	.919	.837	.837	.720	.715	.588	.880	.784

Note: Calculations were performed without the response category of "Don't understand."

As noted by Karat (1988), the questionnaire proved to be a relatively inexpensive, quick and unobtrusive means of gaining evaluation data. However, after calculating the means and standard deviations of the attribute ratings, the limitations of the approach became apparent. Having identified a potential problem area it was then hard, if not impossible, to match this problem area to a specific component of the software. This led to problems

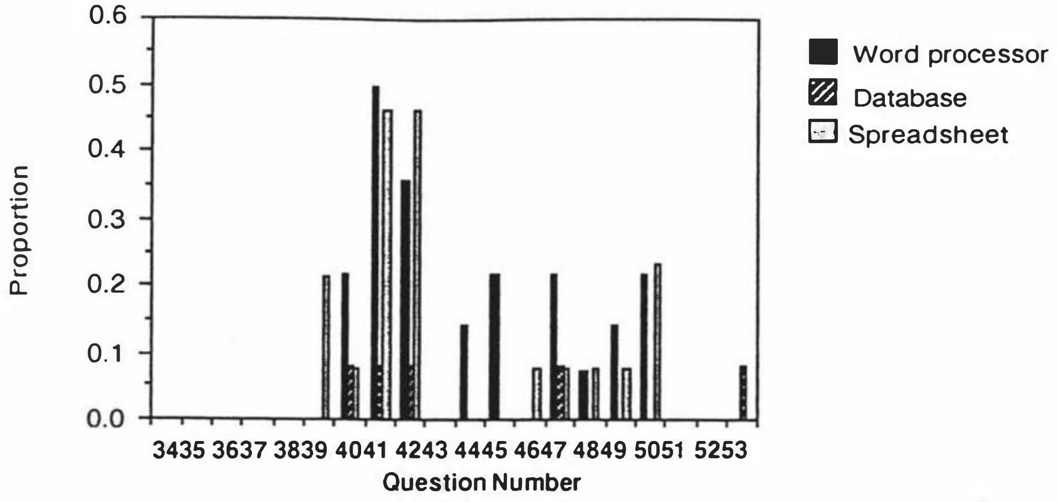


Factor 1 - Programme Self Description

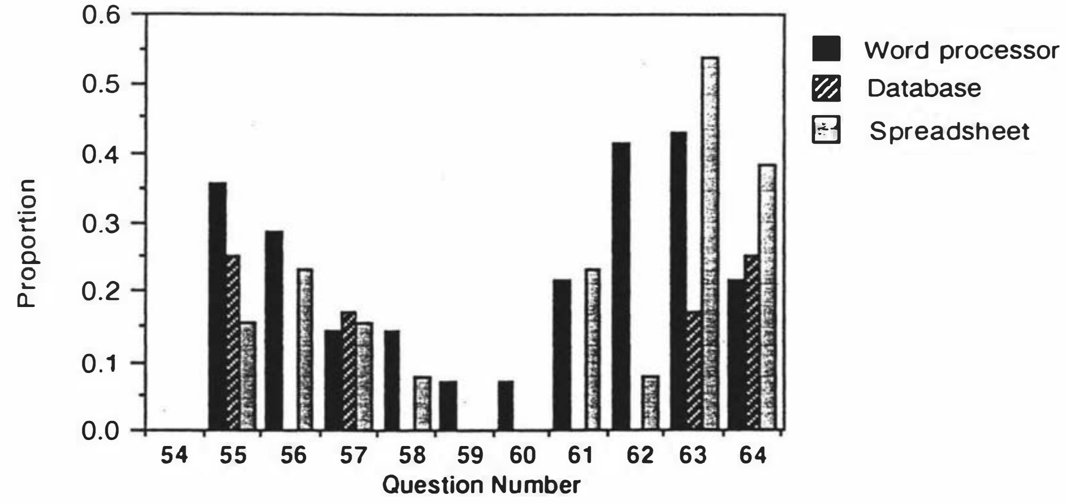


Factor 2 - User Control of the Programme

Figure 6.1. Proportions of Individuals Who Responded 'Don't Understand the Question,' for the Questions in Factors 1 and 2 Over the Three Software Packages Evaluated.

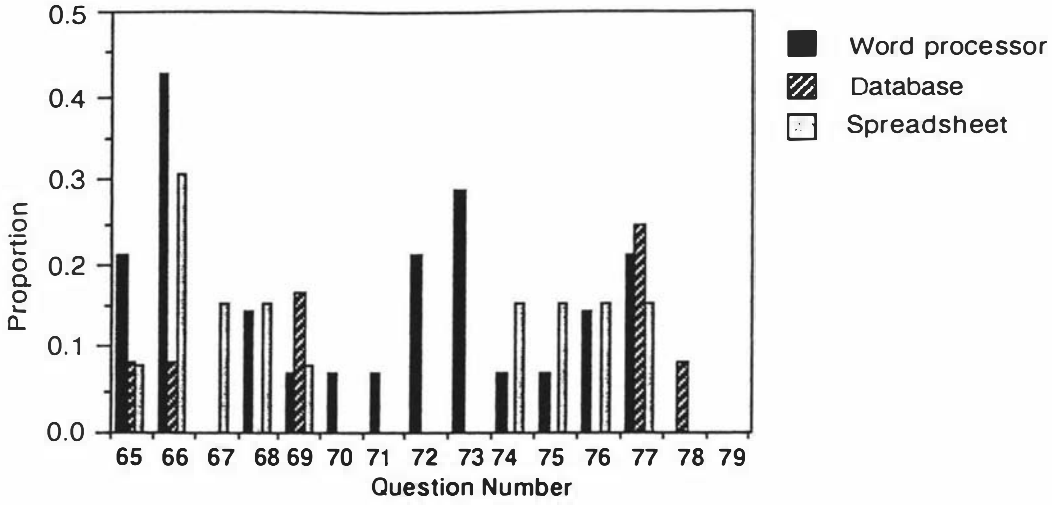


Factor 3 - Ease of Learning the Programme

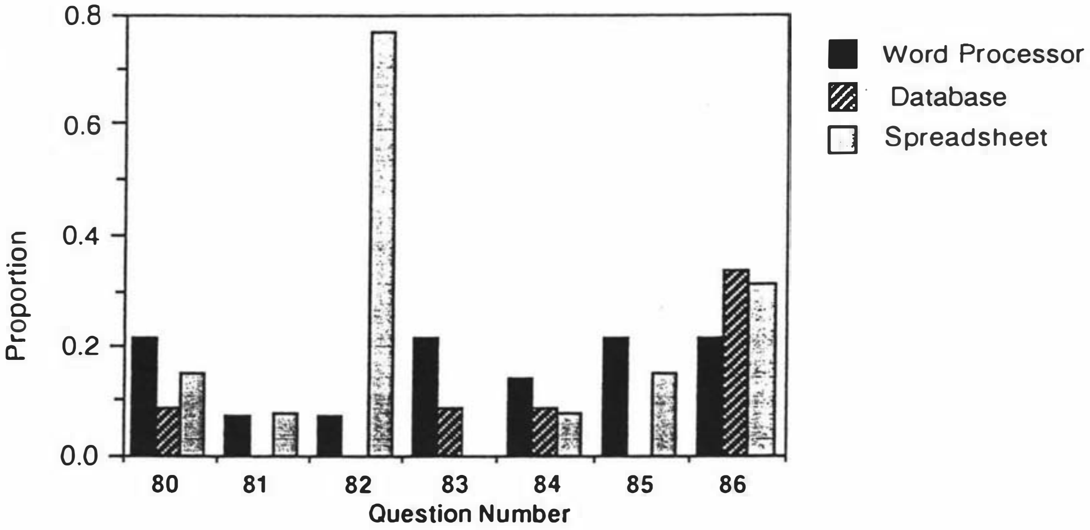


Factor 4 - Completeness of the Programme

Figure 6.2. Proportions of Individuals Who Responded 'Don't Understand the Question,' for the Questions in Factors 3 and 4 Over the Three Software Packages Evaluated.

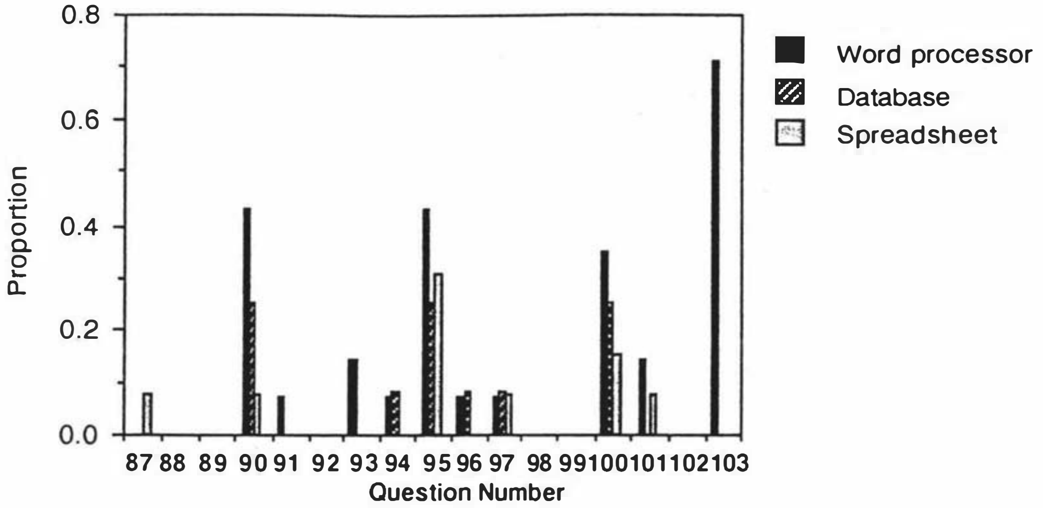


Factor 5 - Correspondence with User Expectations

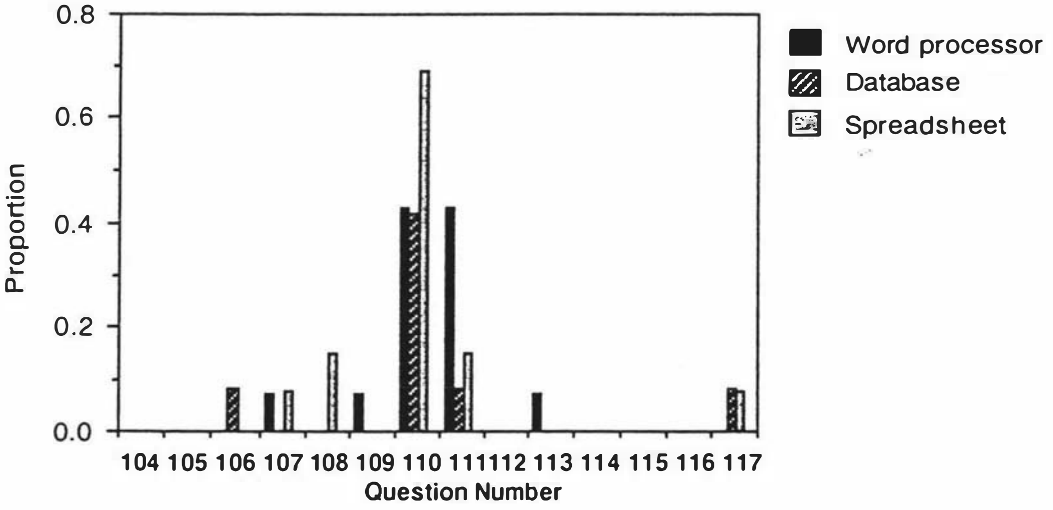


Factor 6 - Flexibility in Task Handling

Figure 6.3. Proportions of Individuals Who Responded 'Don't Understand the Question,' for the Questions in Factors 5 and 6 Over the Three Software Packages Evaluated.



Factor 7 - Fault Tolerance



Factor 8 - Formatting

Figure 6.4. Proportions of Individuals Who Responded 'Don't Understand the Question,' for the Questions in Factors 7 and 8 Over the Three Software Packages Evaluated.

with the reliability of the questionnaire ratings. This suggests that the attribute rating approach may not be as useful as expected. The problems with the attribute rating approach are disturbing, as the majority of software usability questionnaire evaluation forms use this approach (Shneiderman, 1987; Yamagishi and Azuma, 1987; Johnson et al., 1989).

The internal consistency of the sub-scales was calculated and it was found that four of the seven sub-scales were reasonably stable. However, several sub-scales still require further refinement to bring them in line with the desired criterion of 0.60 internal consistency coefficient, stipulated by Nunnally (1967).

It was also found that a large portion of individuals did not understand some of the questions posed, a factor also highlighted by Yang (1989). Using Fiske's (1982) prescription for accurate design it appears that communication had broken down. This break down in communication problem was found in all three situations where the questionnaire was used. If subjects do not understand a question and are forced to respond, the response will not have any validity. It is surprising that usability questionnaires employed in the literature often do not have a "Don't Understand" response category included. This is particularly important when the prospective user group, and therefore hopefully the evaluation group, are non-computer professionals, and may be unfamiliar with computer terms that may be used in the questionnaire.

A similar problem was also encountered with the "Not Applicable" response category. Disagreement about whether or not an attribute was "Not Applicable' or "Applicable" was found. It is assumed that when developing a general questionnaire it will be used on occasions when the software does not use all the attributes, thus making the "Not Applicable" category an important response. However, when there is disagreement as to whether or not an attribute is applicable, there is the suggestion that some response acquiesence on the part of the user may be present. In fact, response acquiescence may have been partly responsible for the low internal reliability coefficients obtained in some sub-scales of the questionnaire. Reasons for this acquiescence may have been due to such factors as poor commitment by the evaluation group, too many rating attributes, poor understanding, etc.

After examining the data it was felt that the quantification of ratings made using this method are inappropriate as a means of assessing the usability of the interface. This finding is congruent with the findings of Johnson et al. (1989), but contrary to the opinions of Shneiderman (1987) and Yamagishi and Azuma (1987). Specifically, there appeared no logical way to combine the information parametrically. The mean scores could not be added to derive an overall index of usability of a package. This parametric approach assumes that all attributes are of equal importance, which is not tenable. Alternatively, although a mathematical formula may be derived, its validity would be very difficult to establish. Furthermore, comparisons across programmes cannot be made because interactions will be present. Therefore, the use of mathematical formulae appears to have dubious validity.

Responses to open ended questions were also obtained, and subjected to a content analysis. Here it was found that the questionnaire elicited responses about problems encountered, a result similar to Root and Draper (1983). However, the overall rate of responding, and the number of problems identified, was low.

The face validity of a questionnaire is also important. Although there was no doubt that the questionnaire looked like a usability questionnaire, problems arose when users tried to answer it. They found that they were required to rate a set of attributes that may not have directly reflected the problems they found with the software. The only opportunity individuals had of expressing the problems they encountered, was in the open ended questions.

It should be stressed that the method used to analyse the questionnaire data may underrepresent the utility of the attribute ratings. The utility of these ratings has not been tested using multivariate statistics. Conversely, the majority of software evaluation sample sizes are small, imposing limits on such analysis for this type of evaluation approach.

The attribute rating approach does not appear to be a very good way of evaluating the user interface, but attribute statements may be useful as a tool in the early stages of the development of the interface. However, further research would be required to evaluate this idea. For pragmatic reasons it is probable that the questionnaire will be used for evaluation, because it is relatively quick, inexpensive and unobtrusive. Further research needs to be conducted into making this method more effective.

In conclusion, when considering the questionnaire results, this research has generally confirmed the findings of previous researchers. Support was found for the findings of Lea (1988), in that the questionnaire results can be used to elicit cognitions and self reports about the interface. The advantages and disadvantages, as highlighted by Meister (1986) (pp 55), and the assertions of Karat (1988), regarding the restricting nature of questions and ex-post facto nature of the approach, also appear to be supported. It is only when focusing on the format of the questionnaire that there is some dispute. The present research disagrees with the findings of Root and Draper (1983) about the suitability and validity, of checklist style questionnaires. This disagreement is disturbing because it seems to be the dominant questionnaire method (Dzida et al., 1978; Root and Draper, 1983; Simes and Sirsky, 1985; Shneiderman, 1987; Yamagishi and Azuma, 1987; and Ravden and Johnson, 1989). Furthermore, the present results do not agree with the suggestions made by Shneiderman (1987) and Yamagishi and Azuma (1987) about the suitability of parametrically combining questionnaire ratings, but the results are in line with Johnson et al. (1989) and Yang (1989).

6.5. The Interview as a Software Evaluation Method.

Table 6.15 presents a short portion of an interview after the audio tape had been transcribed. It should be noted that the edited content tends to be short and relevant. Furthermore, there is little problem understanding what this information relates to.

It was found that, in general, the subjects could articulate how they undertook the task. However, the description tended to be general and brief. The second series of questions addressed problems experienced by the user while undertaking the experimental task. A content analysis of these statements was conducted to identify the problem areas. The results of this analysis appear as Tables 6.5, 6.6, and 6.7 (pp. 116 -118).

Table 6.15. Short Portion of an Interview After Transcription.

Question:	How did you go about saving the spreadsheet?
Answer :	I pressed the / key, then the s for spreadsheet to save. I got a bit lost as there were two different modes, one for entering and one for the menu, telling the difference was trial and error.
Question:	Did you have any other problems while saving the spreadsheet?
Answer:	I pressed enter and I thought the whole thing would save, I didn't realise you had to enter the name.
Question:	Do you have any suggestions as to how we could make this part of the software easier to use?
Answer:	Have a fixed key to bring up the menu.

The final series of questions addressed during the interview involved suggestions for improvement. Although suggestions were given, there appeared to be a direct relationship between the answer given to the "problems experienced" question and the "suggestions for improvement" question. Table 6.16 outlines this relationship. It should be noted that these proportions are based on exact matches only.

Table 6.17 presents the mean ratings and rankings of how difficult each sub-task was perceived. In each case the entering of data was perceived as the easiest task. No other trend is evident across the three packages.

Table 6.16. Match Between the Reply in the Interview "Problems" Section and the Interview "Suggestions for Improvement" Section.

Spreadsheet		Word processor		Database	
Enter	.429	Enter	.462	Enter	.667
Save	.714	Save	.615	Search	.625
Print	.500	Print	.385	Print	.400
Delete	.357	Delete	.385	Delete	.400
Load	.286	Load	.538	List	.500
				Update	.571

Note: These proportions are based on exact matches only.

It was anticipated that the initial question addressing the mode in which the subject undertook the task would provide some insight into the mental model and strategy used in the task. It was found that although the majority of subjects could cite the approach they used when undertaking a sub-task, the value of this information was dubious. This was because no underlying heuristic processes were extracted. However, this could be an artifact of the interview method chosen, and perhaps a more in-depth probing approach could reveal these processes.

The second line of questioning was directed at the problems experienced by users. This line initially held promise because the questions were task oriented and open ended. It was found that problem areas were reported freely by users at a relatively high rate. It was difficult, however, to assess the impact of memory decay, problem recognition, overload and prominence. For example, it is likely that recall of problems was affected by the time between the study and the interview.

Table 6.17. Mean Ratings, and Relative Rankings of Difficulty for Each of the Separate Sub-Tasks Undertaken.

	Package evaluated						
	Spreadsheet		Word processor			Database	
TASK	MEAN	RANK	MEAN	RANK		MEAN	RANK
Enter	2.22	1	2.29	1		2.68	1
Save	2.67	2	3.07	3	Search	3.00	2
Print	5.00	6	3.43	4		3.44	4
Load	3.55	3	2.85	2	List	3.13	3
Delete	3.66	4	5.11	6		3.68	5.5
Insert	3.88	5	4.68	5	Update	3.68	5.5

The third line of questioning was on possible improvements that could be made to the software. It was found that there was a moderate relationship between the problems experienced and the suggestions made. This is understandable, and has implications for the experimental group used in a study such as this. If individuals are too competent in the use of interactive systems, they may not experience difficulties.

The validity of suggestions for improvements are problematic. It seems that they are largely based on the difficulties experienced by the user, and the knowledge the user has of other systems. This tends to limit the efficacy of such suggestions. It can also be argued that the designer(s) tend to know what alterations are feasible, and may therefore have a wider scope for making improvements. Thus, the most important function of such evaluations is to alert the designer to potential usability problems, which they will later refine and retest (although the theoretical and actual may not converge).

Bouchard (1976) has suggested that tape recording the interview may be intrusive and have a strong didactic effect. However, in this study this appeared not to be a problem. Interviewee/ interviewer rapport may be a factor here. This result confirmed the findings of Weiss (1975), however, who suggests that the use of tape recording equipment have several methodological advantages, and that the negative effects are small.

Again, when the interview results are considered on their own, the current research supports most of the previous research. The funnelled interview (Bouchard, 1976), and the quasi-formal questioning style (Meister, 1986), seemed to elicit relevant information about usability issues. The results also lend support to Bainbridge's (1979) statement that, when time and equipment are limited, the interview is probably the best method for soliciting ergonomic information. On the other hand, the differentiating effect of the tape recording process noted by Bouchard (1976) was not evident here.

6.6. The Subsequent Aided Verbal Protocol Analysis Method.

Analysis of the information elicited during the subsequent aided verbal protocol analysis consisted of a content analysis of the problems identified. These results appear in Tables 6.5 to 6.7 (p. 116 - 118). It can be seen that the subsequent aided verbal protocol analysis process highlighted the largest number of problems in all three software packages evaluated.

An initial concern with this process was the intrusiveness of the video taping. However, of those individuals who were asked if they were aware of the video camera, nearly all responded that they were initially aware, but once they began on the task they did not notice it. Also, when playing back the video tapes, subjects did not spend a large amount of time turning and looking at the camera, but were generally task oriented.

The analysable data were gathered during the second experimental session. To gather the data the video was played back, in real time to subjects, and they were asked to "think aloud" while viewing themselves undertaking the task. Prompting was kept to a minimum. This verbal protocol was audio taped. The result of this work supports Karat (1988) who reports that in practice a small number of subjects will provide a rich protocol. It was found that, in general, the subjects did not give a great deal of insight into their cognitive processes as is usually expected when using this approach (Ericsson and Simon, 1980; Bainbridge, 1990). Rather, they seemed to give a running narrative of what they were doing, the problems they were experiencing, and after this, about what went wrong, and how they could overcome the problem. This proved to be beneficial because the information was unambiguous. Tracking problem areas was relatively straight forward, and some insight was gained into the possible causal nature of the problem. Sometimes ways to improve the package were also made explicit.

Table 6.18 presents a short transcript from one of the verbal protocol analysis evaluations. This particular portion is from the word processing task, and is rich in information. By examining the text it can be seen that there is much relevant information here. However, the user makes reference to different situations to those being viewed, and tends to be vague about what is being referred to. This must be expected as the protocol is a free train of thought, and may not move in a sequential fashion. The resulting confusing nature of the data, however, makes the data reduction and analysis time-consuming anc demanding.

In general, the protocols were short, and verbalisation only became prominenl during times of problem identification. It should further be noted that sometimes subjects would contradict themselves saying "I had no problems with this part" at the beginning of a sub-task, and then upon viewing themselves having a difficulty, changing to "Oh, that's right I couldn't move the cursor here . . . " Thus it can be inferred that these reports were imbedded in the realm of reality. Furthermore, the potential problems of recall when using unaided evaluation methods was highlighted, as subjects changed their statements after viewing themselves undertaking the experimental task.

Table 6.18. Short Transcript of a Protocol Obtained While Using the Word Processor.

> "I had problems with the directions here, I needed to hit print twice. It was difficult finding options from the screen, the directions in the help screen were not clear. Here I pushed a line over with the cursor and did not know how to fix it . . . Here I was trying to print and the screen said I had to save but I didn't understand, the second time around I did though, and saved before printing, but this time I just waited for a while and wondered if it had saved, I think the colours are no good here too."

It was found when playing back the video recordings that the technical difficulties associated with flicker rate (Laws and Barber, 1989) were present. This did not seem to impede the protocol process, however. It appeared that the screen definition problem (Laws and Barber, 1989) was not a major problem when using the video record as a protocol prompt.

The subsequent aided verbal protocol analysis method also proved to be extremely time consuming, a point noted by Bainbridge (1979, 1990), Sweeney and Dillon (1987) and Yamagishi and Azuma (1987). In order to obtain data that were in an analysable form, the process consisted of videoing individuals using the system, conducting a post session verbal protocol, and transcribing and reducing this information to an analysable form. All in all, a long tedious process that may not be error free. On the positive side the use of this method transforms the user's role in the evaluation from a passive subject to a more active participant, which leads to extra information being made available to help in the evaluation process and increased commitment to it.

This form of protocol analysis directly illustrates to users what they did, and what problems they had. This has several possible effects. First, the video tape acts as a "perceptual anchor." Faced with the stark facts, a subject cannot gloss over a segment and comment that "I had no problems here," an option open to them in the interview and questionnaire methods. Secondly, as the subjects are now not undertaking the task, all their cognitive activities may be directed towards the problem they had, by doing this the problem is not only identified, but also it may be highlighted why the problem occurred. This second feature still requires validation, but is consistent with the views of Lund (1985).

The subsequent aided verbal protocol analysis method proved efficient at highlighting the usability problems associated with the user interface, supporting previous research findings. In the past, the debate with regards to its validity has focused on the type of information elicited (Nisbett and Wilson, 1977; Ericsson and Simon, 1980). In particular, whether this method has the ability accurately to tap the higher order cognitive processes of the subject. There has never been disagreement as to the validity of personal information, historical facts, focus of attention and current emotions, evaluations and plans (Nisbett and Wilson, 1977).

The information obtained from the verbal protocol method is highly relevant to the software developer and if it is used appropriately, the subsequent aided verbal protocol analysis method is as valid as other forms of evaluations. The present results support the findings of other researchers (Sweeney and Dillon, 1987; Karat, 1988; Robson and Crellin, 1989; Bainbridge, 1990; Kirakowski and Corbett, 1990).

Chapter Seven: A Comparison of the Evaluation Methods.

7.1. A Qualitative Examination of the Information Elicited.

An overview of the information elicited by each of the evaluation methods was conducted by comparing the information obtained from a sub-task, for each of the software packages evaluated. The print sub-task was chosen for this comparison, because it was common to all of the three software packages.

7.1.1. Spreadsheet: The Print Sub-Section.

When examining how the users obtained a print-out, the majority of subjects interviewed could articulate the process they used. Some individuals mentioned that they had experienced difficulty finding the print option when using the menu system. One individual also mentioned that there was a mis-match between the process and their expectations.

It was also found that all four evaluation methods highlighted problems with completing the print sequence. Further to this, a problem associated with the menu system was emphasised by the interview and protocol analysis methods. One problem with the source code was also found by an individual when protocol analysis was used. This, however, was a low frequency problem, so it may have been chance that it was the verbal protocol analysis method that highlighted it.

The logged data, used actuarially, exhibited a reduction in mean and variance in two of the three combinations of the print option. These reductions suggest that learning did take place. Data used in this form did not, however, highlight the nature of problems experienced by the users, suggesting that this information may be less useful than when examined in a contextual fashion.

In the interview the print task was ranked as the most difficult of the sub-tasks, which suggests that the task was difficult, and almost seems at variance with the logged data when used actuarially, which showed significant reductions in the mean times to perform the task and the variance in subjects performance.

Suggestions for improvement obtained during the interview included putting the print command by itself as a main menu option, better prompts, and access to a list of file names that may be printed. There was a moderate relationship between the problems experienced and the suggestions for improvement. The scores on the questionnaire suggested problems in areas such as menus, error messages and lack of help. All this information is non-specific and not as helpful as provided by the other methods. Ambiguity also existed because attributes were sometimes rated as "high," yet also cited as problem areas in the open ended statement section of the questionnaire.

7.1.2. Word Processor: The Print Sub-Section.

Again, subjects interviewed could generally articulate how they undertook the print sub-task. It was also found that all four evaluation methods suggested problems with the error message associated with saving a document before printing, and with the subsequent process of saving the document. The interview and logged data methods also indicated a navigation problem when conducting the print, which may also have compounded the previous problem. The questionnaire and protocol analysis emphasised the lack of feedback on completion of the print task. The protocol analysis also highlighted a problem completing the print sequence that did not show up in previous analyses.

The actuarial logged data were not easy to interpret. There was one reduction in the mean, and two reductions in variance, out of the three occasions, with no specific trend emerging. The ratings for the questionnaire also highlighted problems with the menus, terminology, error correction and messages. However, again, these were non-specific and did not directly refer the evaluator to the print sub-section.

The print task was ranked the fourth most difficult by the interview group with only the insert and delete tasks being perceived as more difficult. Suggestions for improvement consisted of informing the user at an earlier stage, that they must save the document before printing, or deleting the requirement to save documents before printing.

Again the bulk of this information seems to converge. Problems were highlighted by all methods, and ambiguity in interpretation of the actuarial logged data and questionnaire ratings persisted.

7.1.3. Database: The Print Sub-Section.

All four evaluation methods draw attention to a lack of feedback on task completion. The logged data, interview, and protocol analysis also stressed a problem with option selection. Furthermore, the interview and protocol analysis highlighted problems moving between menus. The protocol analysis and questionnaire also emphasised a source code problem, and the interview and protocol analysis highlighted a navigational problem separate to the option selection problem.

The actuarial based logged data again proved difficult to interpret with no specific trend emerging. There was a significant decrease in time taken to print the information between the first and third occasions, with no significant decrease in variance. The questionnaire highlighted several problem areas that may be associated with the print sub-task, but these were non-specific. In the interview, suggestions for improvement included feedback, better instructions, and a buffer step inserted to highlight the print task. The sub-task was ranked the fourth most difficult, with only the delete and update being ranked more difficult, suggesting that the printing task was not easy. A moderate relationship between suggestions for improvement and problems encountered was obtained.

7.1.4. An Overview of the Qualitative Information.

Overlap between the evaluation methods was shown by the number of problems that were suggested by more than one evaluation method. Furthermore, if more than one method is used some redundant information will be elicited. Ratings from the questionnaire tended to be non-specific as they are attribute oriented, rather than task oriented. This latter orientation reduces the usefulness of this technique. It should also be noted that the relationship between the ratings and open ended comments on the questionnaire was not strong, and indeed, were sometimes contradictory, suggesting validity problems. There were also difficulties using the logged data actuarially. It was found that the information was difficult to interpret and was sometimes contradicted by the qualitative data, raising the issue of the relationship between performance indicators and qualitative indicators of usability.

The qualitative comparison also suggested that, in general, information gained from the interview and subsequent aided verbal protocol analysis evaluation methods converged, a result similar to that reported by Yamagishi and Azuma (1987). This relationship was less so for the logged data and questionnaire methods. This finding is interesting, as it suggests that these latter methods may be deficient; that is, they may not adequately cover the usability construct when addressing problem areas. However, as with all criterion development projects, these results cannot be taken at face value. It must be remembered that the questionnaire produced a large number of ratings. It is possible that these attributes are tapping another facet of the usability construct, or alternatively there could be a problem of criterion contamination. After all, a large portion of users did not understand some questions, and there was a large amount of variance associated with the "Not Applicable" response, and some nonexistent attributes were rated as "good." All these factors strongly suggest that some of the ratings are unreliable.

The same problems are also present when examining the performance based logged data. Here it was found that, sometimes, no discernable trends emerged. Furthermore when trends did emerge, they were sometimes contradictory, suggesting that such information exhibit criterion contamination.

7.2. Usability Problems Highlighted by Evaluation Methods Used Alone.

It was hypothesised that the efficiency of the evaluation methods would be independent of the software being evaluated. Table 7.1 presents the number of problems highlighted by each evaluation method in each of the three packages evaluated (derived from Tables 6.5 - 6.7, pp. 116 - 118). It can be seen that in all cases the protocol analysis identified the most problem areas, a result similar to that obtained by Sweeney and Dillon (1987). These results are depicted in graphical form in Figure 7.1. It can be seen in Figure 7.1 that the protocol analysis was the most effective at highlighting problem areas, followed by the interview, then the questionnaire, with the logged data being the least effective.

A hierarchical log linear analysis was conducted to examine the robustness of this trend. A fully saturated model was chosen, while using a backward elimination process, to find the best model. The results of this analysis are presented in Table 7.2. It should be noted that the main effect for evaluation method was significant and this was independent of software type. Furthermore, the pattern generated by the evaluation method alone is not significantly different from the observed data (χ^2= 10.522, df=8, p= .230). These results suggest that evaluation efficiency is not related to the type of software but rather to the type of evaluation method.

Table 7.1. Summary of the Number of Problems Identified by Each Evaluation Method Across Each Software Package Evaluated (derived from Tables 6.5 to 6.7).

Key. LD = Logged data.
I = Interview.
Q = Questionnaire.
VP = Verbal protocol analysis.

	Evaluation Method				
Software	LD	I	Q	VP	TOTAL
Spreadsheet	13	17	15	24	38
Word Processor	10	17	15	25	35
Database	16	16	14	19	31
Total	39	50	44	68	

* Row totals will not equal column totals as any one problem may be highlighted by more than one method.

It was also hypothesised that there would be no difference between the evaluation methods with regards to the number of problems identified. To examine this hypothesis, a chi square analysis was conducted on the column totals, for each evaluation method used (see Table 7.1). It was found that there was at least one significantly different evaluation method with regards to its ability to identify problem areas (χ^2= 9.287, df=3, p=.026). A pair-wise comparison revealed that the subsequent aided verbal protocol analysis was significantly better at identifying problem areas than the logged data and questionnaire methods (see Table 7.3). No other significant differences were observed.

In summary, the hypothesis that there would be no differences between the methods with regards to the number of problems identified was rejected. After establishing that evaluation method alone was the best model predictor, a chi square test confirmed that there were significant differences between the evaluation methods with regards to the number of problems highlighted.

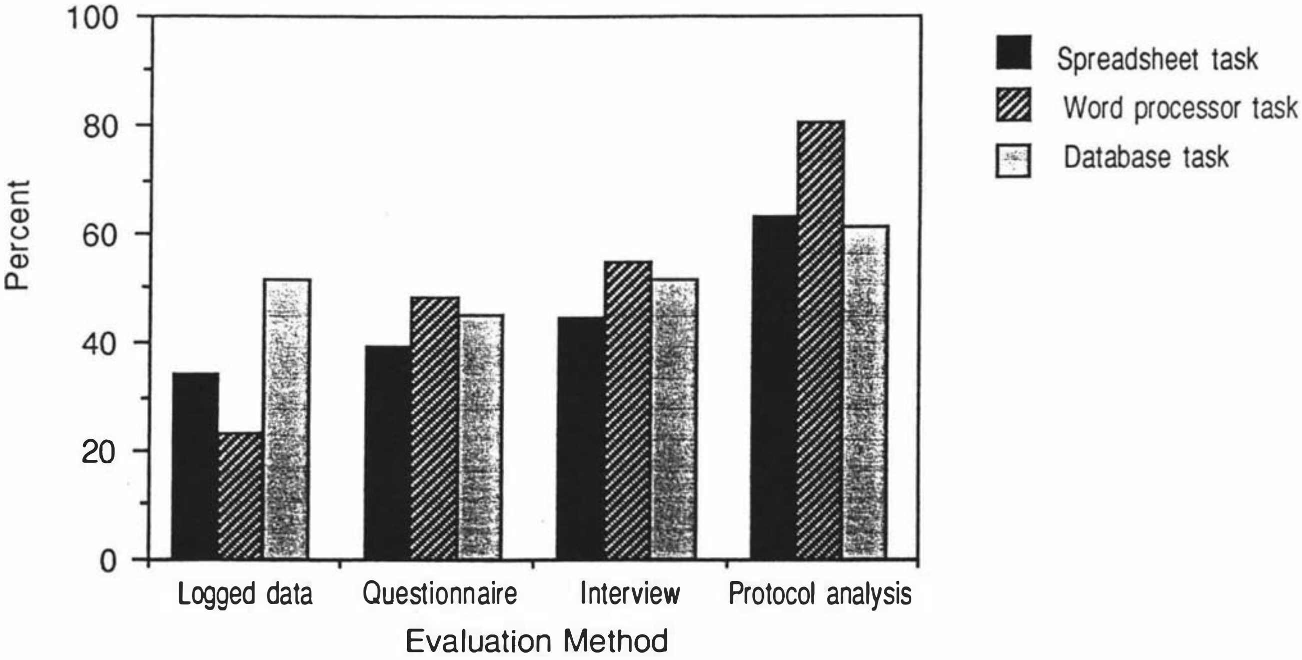
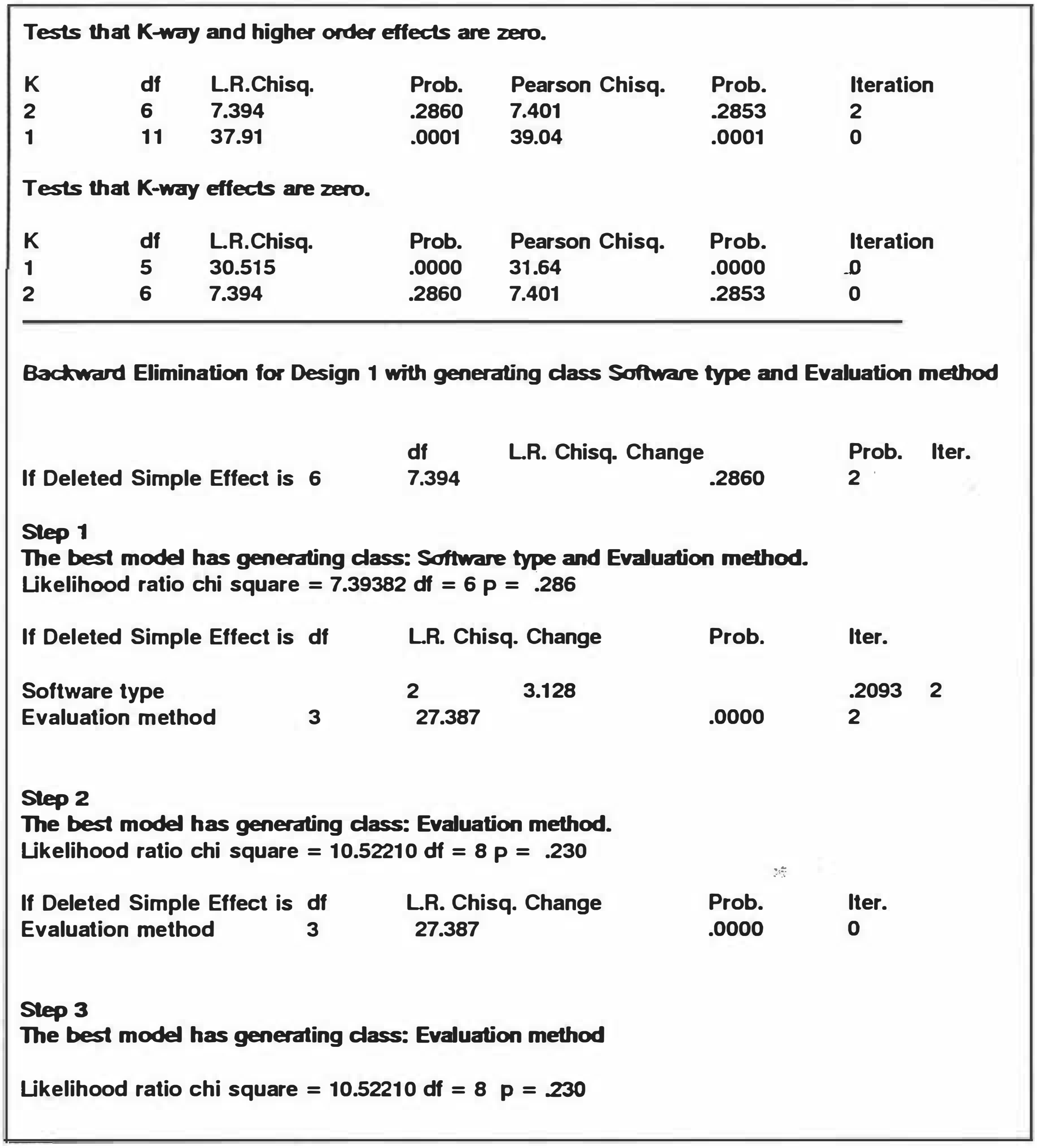


Figure 7.1. Percentage of the Total Usability Problems Identified by Each Evaluation Method Over the Three Evaluation Tasks.

Table 7.2. Hierarchical Log Linear Analysis of the Frequency of Problems Identified by Each Evaluation Method Within Each of the Three Application Domains Evaluated.



```
Tests that K-way and higher order effects are zero.

K        df    L.R.Chisq.        Prob.    Pearson Chisq.    Prob.        Iteration
2        6     7.394             .2860    7.401             .2853        2
1        11    37.91             .0001    39.04             .0001        0

Tests that K-way effects are zero.

K        df    L.R.Chisq.        Prob.    Pearson Chisq.    Prob.        Iteration
1        5     30.515            .0000    31.64             .0000        .0
2        6     7.394             .2860    7.401             .2853        0

Backward Elimination for Design 1 with generating class Software type and Evaluation method


                                  df       L.R. Chisq. Change              Prob.   Iter.
If Deleted Simple Effect is  6    7.394                         .2860      2

Step 1
The best model has generating class: Software type and Evaluation method.
Likelihood ratio chi square = 7.39382 df = 6 p =  .286

If Deleted Simple Effect is  df    L.R. Chisq. Change        Prob.         Iter.

Software type                      2         3.128                         .2093   2
Evaluation method            3      27.387                   .0000         2


Step 2
The best model has generating class: Evaluation method.
Likelihood ratio chi square = 10.52210 df = 8 p =  .230

If Deleted Simple Effect is  df    L.R. Chisq. Change        Prob.         Iter.
Evaluation method            3      27.387                   .0000         0


Step 3
The best model has generating class: Evaluation method

Likelihood ratio chi square = 10.52210 df = 8  p = .230
```

Table 7.3. Comparison Between the Number of Problem Areas Identified by Each Evaluation Method Using the Chi Square Statistic.

Key. LD = Logged data.
Q = Questionnaire.
I = Interview.
VP = Verbal protocol analysis.

Software	LD	Q	I	VP

LD	-	χ^2= .43
		df = 1
		p = .51
			χ^2= 1.36
			df = 1
			p = .24
				χ^2= 7.86
				df = 1
				p = .005*

Q		-	χ^2= .26
			df = 1
			p = .61
				χ^2= 4.68
				df = 1
				p = .03*

I			-	χ^2= 2.75
				df = 1
				p = .098

VP				-

* $p < .05$.

An examination within the evaluation methods indicated that the subsequent aided verbal protocol analysis method was significantly better at highlighting problem areas than the logged data and questionnaire method.

7.3. The Incidence of Problem Identification within each Evaluation Group

The total number of problems identified by an evaluation method is one dimension of efficiency. The incidence of reporting a problem within an experimental group is another dimension. To examine the incidence of reporting a problem a total of six high frequency problems were chosen (two from each software package evaluated). This analysis was done by analysing the logged data and identifying the two most frequently encountered problem areas within each software package.

The percentage of individuals displaying the problems in the logged data group was then calculated. This figure subsequently was used as a standard to test the other evaluation methods. Table 7.4 presents these results (also presented in visual form in Figure 7.2). An examination of Table 7.4 suggests that, in general, the results from the subsequent aided verbal protocol analysis group seem similar to the results obtained from the logged data group. However, in the questionnaire and interview groups there appears to be a lower incidence of individuals reporting the problems.

Chi square tests were conducted between the frequency of reporting each of the problems using the logged data method and each of the other three methods (see Table 7.5). In all cases there were no differences with regards to the incidence of problem reporting between protocol analysis groups and logged data groups. In the interview groups, differences were observed on four of the six tests, and with the questionnaire groups, differences were observed on five of the six tests. In each of these cases the incidence of reporting problem areas was lower than the logged data rate.

It was initially hypothesised that the "subjective" evaluation approaches of the questionnaire, interview and verbal protocol analysis would exhibit a significantly lower incidence of problem reporting within the three groups than the more "objectively" recorded logged data. Unexpectedly, the protocol analysis method did not exhibit a different incidence of problem reporting on any of the six problem areas examined. In the interview group, however, a significantly different, and lower, rate of reporting was found for four of the six problems, and in the case of the questionnaire a significantly different, and lower, rate was found in five of the six areas.

Table 7.4. Percentage of Subjects Who Reported Each of the Six Most Frequently Occurring Problems.

Key. LD = Logged data.
Q = Questionnaire.
I = Interview.
VP = Verbal protocol analysis.

Problem	Method			
	LD	Q	I	VP
Spreadsheet				
- Activating main menu.	46.67	7.67	21.43	50.00
- Incorrect row insertion.	66.67	7.69	28.57	58.33
Word processor				
- Multiple open files.	45.45	0.00	23.08	40.00
- Print problems.	63.64	28.57	53.84	70.00
Database				
- Delete.	42.86	33.33	50.00	50.00
- Exit add sub system.	28.57	0.00	10.00	20.00

Table 7.5. Summary of Chi Square Tests of the Relationship Between the Percentage of Individuals Encountering a Problem in the Logged Data Group and the Questionnaire, Interview and Protocol Analysis Groups Respectively.

Questionnaire			Interview			Protocol analysis		
χ^2	df	p	χ^2	df	p	χ^2	df	p
29.630	1	.000*	9.328	1	.002*	.093	1	.761
46.431	1	.000*	16.011	1	.000*	.648	1	.421
46.000	1	.000*	7.118	1	.008*	.421	1	.518
13.172	1	.000*	1.633	1	.201	.066	1	.798
1.052	1	.305	0.527	1	.468	.527	1	.468
29.000	1	.000*	8.526	1	.004*	1.653	1	.199

* Significant beyond $p<.05$.

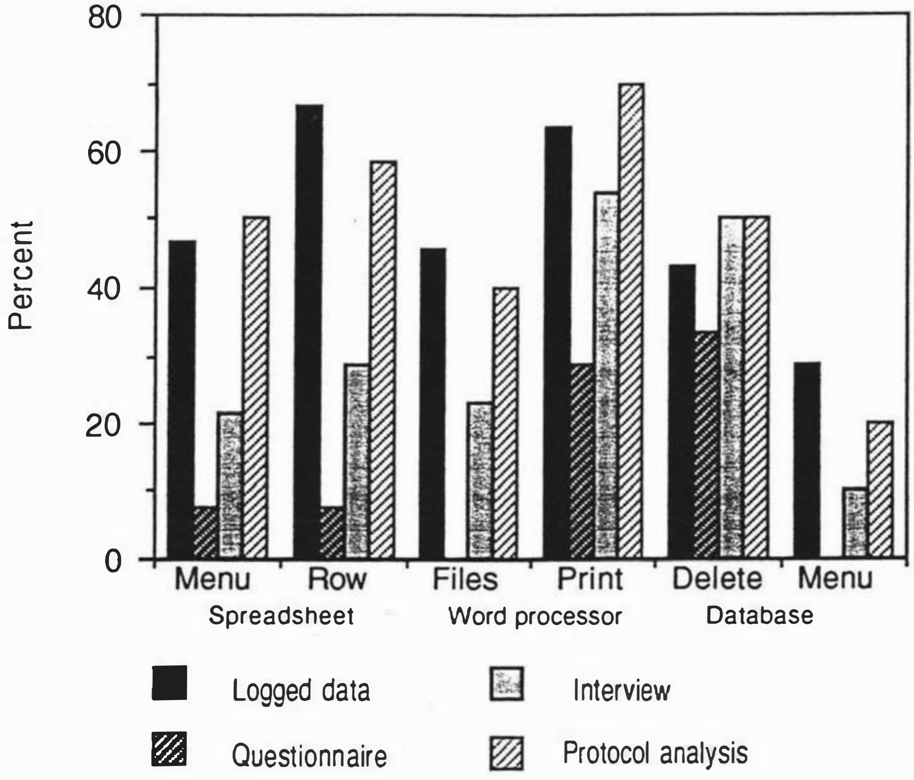


Figure 7.2. Percentage of Individuals Reporting Each of the Six Most Frequently Occurring Problems Examined.

The rate of reporting a problem within a group can be used as an indicator of the validity of the information elicited using these methods. The assumption made, is that groups are equivalent, with respect to the rate at which individuals within each group encounter problems. If so, unobtrusively recorded logged data can be used as an "objective" record of the incidence at which individuals within a group encountered a specific problem. The difference between the actual rate of individuals encountering a problem area, and the rate at which it is reported, is an indicator of the validity of the measure. Subsequent aided verbal protocol analysis produced a response pattern that is similar to the logged data, but this was not the case for the interview and questionnaire.

7.4. Problem Identification Using Two Evaluation Methods.

It was hypothesised that by combining two evaluation methods a significant improvement, on the number of problems identified by a single evaluation method, would be observed. To examine this prediction a series of matrices was constructed. Here the percentage of problems identified by each combination of two evaluation methods could be compared to each single method. This was done by first calculating the percentage of problems identified by each evaluation method used alone. Next, the percentage of problems identified by combining two evaluation methods in a composite fashion, eliminating duplication, was calculated (see Table 7.6). It can be seen that some combinations do not highlight as many problems as the protocol analysis used alone. Furthermore, the protocol analysis combined with other methods appears again the dominant method for identifying problem areas (see Figure 7.3).

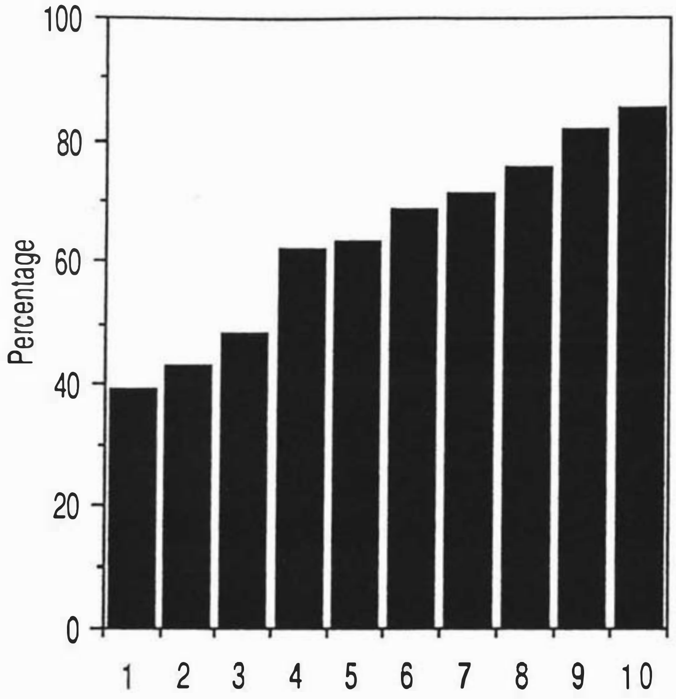
These results were further investigated using hierarchical log linear analysis. This was done by conducting the analysis on the percentage of the total problems identified by each method alone, and the percentage of problems identified when using a combination of two methods. Again a main effect for

Table 7.6. Percentage of the Total Number of Problems Identified Using Two Evaluation Methods, Compared to Each Single Method.

Key. LD = Logged data
Q = Questionnaire
I = Interview
VP = Verbal protocol analysis

Software	LD	Q	I	VP
Spreadsheet				
Single method	34.21	39.47	44.74	63.16
LD	-	57.89	60.53	73.00
Q		-	68.42	81.59
I			-	76.32
VP				-
	LD	Q	I	VP
Word Processor				
Single method	28.57	42.86	48.57	71.43
LD	-	51.62	58.06	90.32
Q		-	67.74	80.65
I			-	100.0
VP				-
	LD	Q	I	VP
Database				
Single method	51.61	51.61	45.16	61.29
LD	-	77.42	70.97	80.65
Q		-	77.42	64.52
I			-	80.65
VP				-

type of evaluation was observed, and this was independent of the software evaluated (see Table 7.7).



Evaluation Methods

1 = Logged data
2 = Questionnaire
3 = Interview
4 = Logged data and questionnaire
5 = Logged data and interview
6 = Protocol analysis
7 = Questionnaire and interview
8 = Questionnaire and protocol analysis
9 = Logged data and protocol analysis
10 = Interview and protocol analysis

Figure 7.3. Average Percentage of the Total Number of Problems Identifed Using Two Evaluation Methods, Compared to Using a Single Evaluation Method.

Table 7.7. Hierarchical Log Linear Analysis of the Frequency of Problems Identified by Each Evaluation Method and Each Combination of Two Methods Within Each of the Three Types of Software.

```
Tests that K-way and higher order effects are zero.

K      df     L.R. Chisq.   Prob.        Pearson Chisq.        Prob.     Iteration
2      18     22.536        .2091        22.604                .2063     2
1      29     138.398       .0000        133.750               .0000     0

Tests that K-way effects are zero.

K      df     L.R. Chisq.   Prob.        Pearson Chisq.        Prob.     Iteration
1      11     115.863       .0000        111.147               .0000     0
2      18     22.536        .2091        22.604                .2063     0
_____________________________________________________________________________

Backward elimination for Design 1 with generating class Software type and Evaluation method

                             df          L.R. Chisq. Change    Prob.     Iter.
If Deleted Simple Effect is  18          22.536                .2091     2


Step 1
The best model has generating class: Software type and Evaluation method.
Likelihood ratio chi square = 22.53576 df = 18  p = .209

If Deleted Simple Effect is  df          L.R. Chisq. Change    Prob.     Iter.

Software type                    2       3.437                 .1793     2
Evaluation method                9       112.426               .0000     2


Step 2
The best model has generating class: Evaluation method
Likelihood ratio chi square = 25.97275 df = 20  p = .167

If Deleted Simple Effect is  df          L.R. Chisq. Change    Prob.     Iter.

Evaluation method            9           112.426               .0000     0


Step 3
The best model has generating class: Evaluation method.
Likelihood ratio chi square = 25.97275 df = 20 p =  .167
```

Also of interest is the notion of incremental improvement. If two methods are used, is there a significant improvement, with regards to the number of problems highlighted, over using the best single method? To examine this question, a chi square test was conducted between the mean percentage of problems identified by the verbal protocol analysis, and the mean percentage of problems identified by each of the possible combinations of any two evaluation methods. The fact that no significant effects were found (χ^2 = 6.411, df= 6, p = .379) suggests that although improvement may be affected by using more than one evaluation method, this improvement is marginal at best.

The finding that no significant improvement was found when using two evaluation methods is important because a large number of methods being advocated (for example, Playback type approaches: Neal and Simons, 1984a, 1984b; Hietala, 1985; Morris et al., 1988), are essentially composite methods and involve more than one of the four methods, used in this study. The results of the present study suggest that this may not be a good strategy and may lead to redundancy of information.

7.5. Practical Psychology.

Smith (1982) makes the distinction between pure, applied and practical psychology. Smith argued that pure psychology is directed at establishing and testing psychological theory. In contrast, applied psychology focuses its energy on applying pure psychological findings to the applied setting. Practical psychology, on the other hand, is directed at empowering practitioners to use the tools, techniques, and findings of both experimental and applied psychology. As Landy (1989) has said, criterion data must be reliable, valid, and practical. Regardless of how reliable and valid an evaluation method is, it will not be used if there is not some practical means by which the evaluation information can be obtained, analysed and interpreted. It is therefore appropriate to examine the practicality of each evaluation method used.

7.5.1. Practical Feasibility of the Methods.

To investigate the practicality of the software evaluation methods a small post hoc examination was undertaken by the evaluator, in an introspective manner. While acknowledging the limitations of such a subjective account, the impressions of difficulty gained by the evaluator are of practical use to future researchers and evaluators, contemplating the use of these techniques. For example, within the evaluation setting, such practical consideration will impact upon the choice of the evaluation method.

First, the issues concerning the software methods during the study were first isolated. To do this the practical aspects associated with the data collection, reduction, analysis, and interpretation phases of the evaluation were first delineated. This involved considering each phase of the study and articulating the steps, problems, and considerations encountered during the evaluation. Next, the practical considerations associated with carrying out each identified aspect were compared across evaluation methods, using a paired comparison approach. This involved comparing each method, with regard to its level of difficulty. For example, the logged data and the verbal protocol analysis method were compared on how intrusive each method was during the data collection phase of the study, then the logged data and the interview, etc., until all had been compared. These comparisons were repeated for each aspect within each phase of the evaluation.

This paired comparison approach resulted in a ranking on each evaluation method for each aspect considered. These rankings were then transformed into an interval scale, thus allowing for a single standardised score for each phase of the evaluation process (Blum and Naylor, 1968). Table 7.8 presents the interval level scores for each evaluation method for the aspects considered.

Table 7.8. Summary of Practical Aspects. All Attributes Have Been Compared Using a Paired Comparison Method. A Score of One Indicates the Most Easy to Use Method Through to Zero Representing the Most Difficult Method.

	Evaluation method			
	Logged Data	Questionnaire	Interview	Protocol analysis
Low Score = Hard to use. High Score = Easy to use.				
1. Data collection.				
Hardware.	0.33	1.00	0.67	0.00
Effort.	1.00	0.67	0.33	0.00
Intrusiveness.	1.00	0.67	0.33	0.00
Technical expertise.	0.33	0.00	1.00	0.67
Subjectivity.	1.00	0.00	0.33	0.67
Bias.	1.00	0.00	0.33	0.67
2. Data reduction.				
Subjectivity.	0.67	1.00	0.00	0.33
Transformations.	0.00	1.00	0.67	0.33
Time.	0.00	1.00	0.67	0.33
Technical Expertise.	0.00	1.00	0.67	0.33
3. Data analysis.				
Researcher bias.	0.67	1.00	0.33	0.00
Expertise.	0.00	0.33	1.00	0.67
Time.	0.67	1.00	0.33	0.00
Richness of information.	0.67	0.00	0.33	1.00
Problems.	0.00	0.33	0.67	1.00
Frequency.	1.00	0.00	0.33	0.67
Subjectivity.	1.00	0.00	0.67	0.33
Convergence.	0.00	0.33	0.67	1.00
Redundancy.	0.00	0.33	0.67	1.00
4. Data interpretation.				
Subjectivity.	0.33	0.00	1.00	0.67
Type of information.	0.33	0.00	1.00	0.67
Technical expertise.	0.00	0.33	1.00	0.67
Sample size.	0.33	0.00	0.67	1.00
Researcher bias.	1.00	0.00	0.33	0.67

The interval level scores can then be transformed into a graph to highlight the comparative strengths of each evaluation method over each of the evaluation stages. A low score denotes difficulty, and a high score denotes ease of use (see Figure 7.4). From Figure 7.4 it can be noted that during different

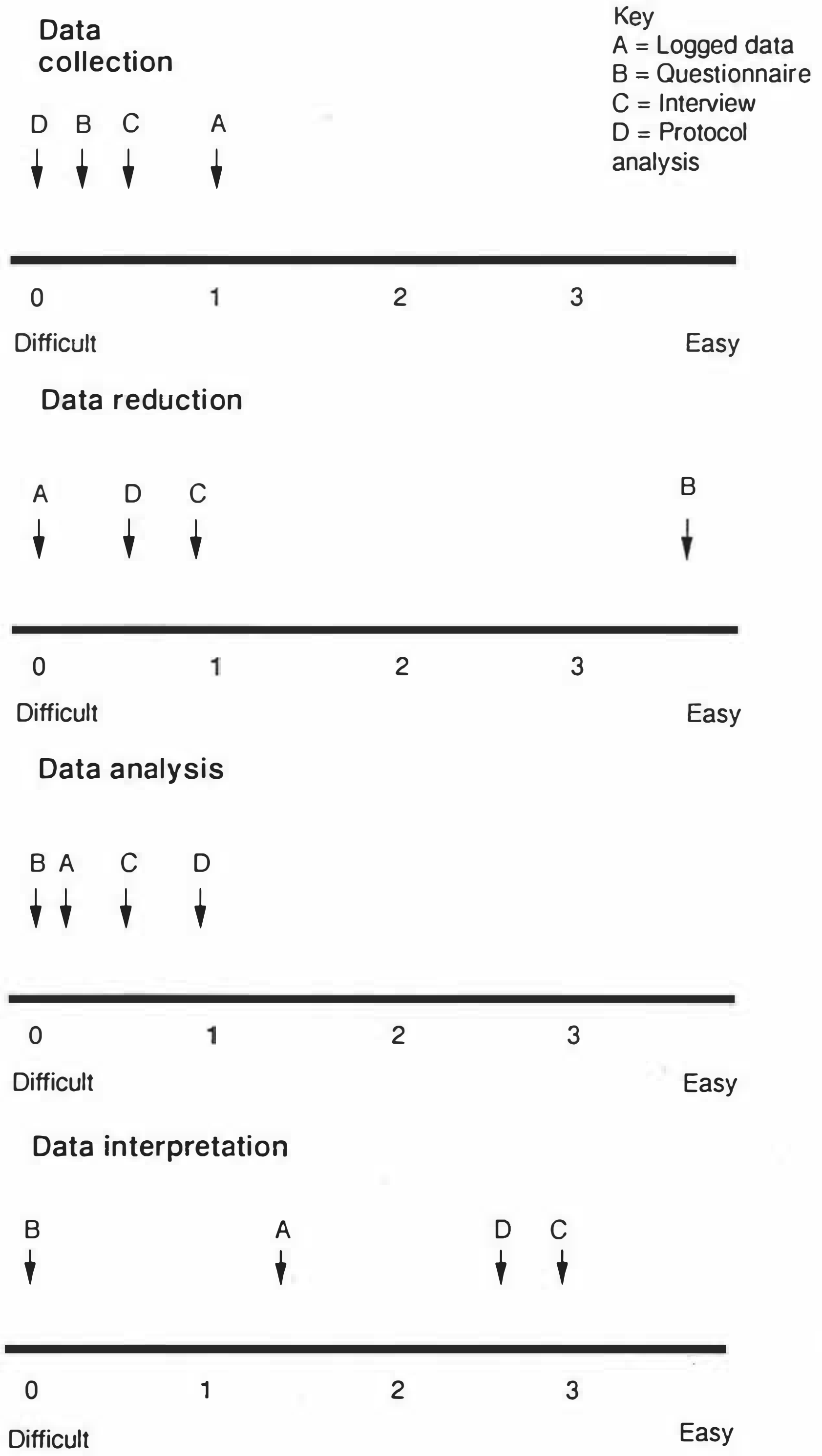


Figure 7.4. Comparison of Evaluation Methods on a Standardised Continuum of Merit.

phases of the evaluation, the relative merit of each method may change.

During the data collection phase the logged data are the easiest to use, during data reduction the questionnaire, during data analysis the interview, and during the interpretation, the protocol analysis. No one method stands out as being the best single method as far as practical aspects are concerned.

It can also be seen from Figure 7.4 that during data collection and analysis the methods seem to cluster relatively closely. During the data reduction phase, the logged data, interview and protocol analysis cluster, but the questionnaire stands out strikingly as the easiest method to implement. The results of the foregoing analysis suggest that there is generally little difference between the methods during these phases of the evaluation and preference may dictate the approach used.

It is during the data interpretation phase that a large difference is exhibited. Here it should be noted that the interview and protocol analysis are relatively close together on the high side of the continuum compared to the logged data and the questionnaire methods. If data interpretability is the focal point in the evaluation process it would appear that the protocol analysis and interview have the most merit.

The information gained from this paired comparison exercise is interesting. Due to the simplicity of data collection and analysis, new or inexperienced evaluators may initially be tempted to use the logged data or questionnaire approach. Each method is easy to administer, and appears to gather important information. However, these methods tend to be less useful in the interpretation phase of the evaluation. It would seem that the price for simplicity is less useful information. In contrast, evaluators may be reluctant to use the interview or protocol analysis methods because they appear time consuming and labour intensive. However, these methods are the methods that yield data that are more interpretable.

Chapter Eight: General Discussion.

8.1. A Theory of the User Based Software Evaluation Process.

The efficiency of any human-computer interaction evaluation method is dependent upon a number of factors. When identifying usability problem areas, the user must initially experience interaction problems. The relationship between undertaking a task and experiencing difficulty is dependant upon both external and internal factors. Internal factors such as the user's intelligence, aptitude, experience with computer based technology, and similar software, will all impact upon whether or not the individual will experience difficulty. For example, users may initially experience difficulty with some aspect of using a programme. However, if they have been exposed to similar problems in the past, they may have an adequate coping strategy, which may result in the problem being overcome. External factors such as the difficulty of the task to be undertaken, length of the evaluation session and the evaluation setting itself will also impact upon whether or not the individual will experience difficulty.

Furthermore, if the cause of a problem is perceived as internal (e.g., "I made the mistake, there is nothing wrong with the software"), the individual may not report the instance as a problem. Conversely, if the cause of the problem is attributed to software usability, this will be more likely to be perceived, and then reported as a usability problem.

Reporting an experience as a problem may also be dependent upon several other factors. Difficulties that are overcome with little effort are less likely to be reported than problems that require more effort. The impact the problem has upon the task completion will also play a part in deciding whether or not it is reported. Those that cause major difficulties, for example, exiting the programme, or irreversibly deleting information, are more likely to be reported than problems that simply require several further key strokes to complete a task.

The number of problems encountered will also impact upon the probability of reporting. If there are only one or two problems they are likely to be reported, but where there are many problems some may not be reported due to retention difficulties.

The interaction between the user and evaluator is another important factor. This interaction particularly has an impact on evaluation methods that rely on interpersonal communication as a medium for gathering evaluation data.

Reporting a problem may also be related to the probability of the experience going undetected. That is, the probability of reporting a problem when using a questionnaire may be different from that of watching a video in the presence of the evaluator. Furthermore, the probability of reporting problems may also be related to the amount of personal commitment and the knowledge of the user. People who will be required to use the product in the future may be more likely to report problems, than those who will never have to use the system again.

The probability of the evaluator wrongly interpreting data is related to several factors. The amount of subjectivity in the data reduction, and the analysis and interpretation phases of the evaluation are some of these. The probability of incorrect interpretation can be moderated by the input of the user. Specifically, those evaluation methods that encompass large amounts of data reduction, and are independent of the user, are more likely to lead to incorrect interpretations by the evaluator, compared to those that require a small amount of data reduction. Access to the user will also aid the interpretation of the data. An evaluator could consider an area to be of little importance but if the user is available, this view could change.

8.1.2. A Post-Hoc Examination of the Evaluation Process Using the Data in the Present Study.

In the present experiment, users encountered problems. Their cause will have been attributed to either the usability of the programme or some other factor, such as user ability, experience, etc. If the problem is attributed to the usability of the programme each user will then have mentally noted the problem area. It must be remembered that no user knew which evaluation method they would be using, until after completing the experimental task. Therefore, all of them would have been similarly motivated to remember problems and comments.

After completing the experimental task, each evaluation group theoretically would have a similar distribution of experiences to report. At this stage the usability evaluation is important. In the logged data group all key strokes have been recorded and, excluding technical difficulties, the fidelity of the evaluation information at this stage is sound. However, at the same point with the questionnaire group, the user was asked to rate the package on a series of attributes. When doing these ratings the relationship between the attributes and the problems encountered is questionable. Although the questionnaire may look like a usability questionnaire, the respondent may be slightly put off when asked to rate some attributes to which they had paid little attention. The open ended sections will, however, give the users an opportunity to direct the evaluator to their specific concerns and experiences. The problems of memory retention, comprehensiveness and the comprehensibility of the questionnaire all impact on the validity of the responses.

In the interview, the user was asked to address each specific sub-section and recount the approach they used to do the task. This in effect may have focused the respondent on the task, and involved some mental rehearsal of how they undertook it. Subjects were then asked to highlight the problems they experienced while undertaking each task. Retention and factors such as the user's causal attribution of the problem will impact at this time. It was only when using the subsequent aided verbal protocol procedure that the problem of retention was resolved. With this method the user recounts their experiences in real time with the evaluator. A user cannot say they did not experience a problem, and then view themselves on a video monitor experiencing obvious difficulties.

The next stage of the evaluation involved data reduction and analysis. Here the fidelity of the logged data is a problem. When using the data actuarially, target key sequences must be identified and relevant information recorded. This process can be accomplished relatively effortlessly, but it is time consuming, and sometimes target key sequences are ambiguous. Next, the data must be analysed using some statistical approach, with both bivariate and multivariate analysis procedures possible.

Analysis of variance has been proposed as an applicable technique for such analysis (Shneiderman, 1987). It must be stressed, however, that homogeneity of variance is an explicit assumption when using this method. For the present studies this assumption was not tenable, making it somewhat inappropriate to use this technique. It is also an omnibus approach so its usefulness is questionable. This leaves the correlated t-test and reduction in variance tests as the most appropriate bivariate statistical procedures.

A host of multivariate procedures are potentially available to the evaluator. In particular, regression analysis, cluster analysis and multivariate analysis of variance, all of which can be used both to test hypotheses and model data. It is doubtful, however, whether many commercial developers of software would have access to personnel who could use these methods. Thus, in practice, their use might be limited.

Other data reduction techniques when using logged data may consist of a content analysis, conducted formally or informally, followed by frequency counts. Analysis can take many forms ranging from non-parametric analysis to multivariate analysis, including cluster analysis, and log linear analysis. This process, although time consuming, is not as difficult as when analysing logged data actuarially.

The questionnaire requires relatively straight forward data reduction techniques. The derivation of means and standard deviations is relatively easy, and answers to open ended questions can be subjected to a low level content analysis. The interview is more difficult in the data reduction and analysis phases. First, audio tapes must be transcribed, which allows error variance in the form of misperceptions to enter the evaluation process. Next, content analysis must be performed on the data. Although this procedure seems straightforward the exercise is time consuming and tiring. By using a semi-structured interview the evaluator is aided by limiting comments to one task area at a time, and this adds at least some structure and context to the data reduction phase of the interview.

The verbal protocol analysis system is similar to the interview, but has the problem of being relatively unstructured and prone to repetition. To reduce data, information must first be transcribed from the audio tape, and then a content analysis must be performed. To conduct the content analysis the evaluator may use the evaluation task outline as an aid when interpreting statements made by the user. However, this portion of the evaluation is still confusing and tiring. Furthermore, the process is sometimes confounded by the user making statements that cannot be specifically analysed from the audio tape. For example, "Oh, yes! I had problems with this part of the print out." This, in turn, results in either data being lost, or inferences being made by the evaluator. Subjects may also repeatedly encounter the same problem leading to a repetition of comments.

Further to the problems encountered during the data reduction and analysis process, the evaluator must make some sense of the data. Problems with the logged data become apparent again. Essentially, the evaluator has difficulties identifying and classifying potential problem areas. Why did the user choose this path to undertake this task? There are a host of possible reasons, and without the presence of the user to elaborate, the evaluator is working "blind," and becomes subject to perceptual biases and their own theories of causality, none of which are verifiable.

The questionnaire also presents problems associated with data interpretability. What does a mean of 4.5 mean? How does this effect the score of 2.5 on the next attribute? How should cut-off points be set? All of these issues and many others, need to be addressed. The only data that do not suffer from this problem in the questionnaire are the open ended answers. But it is not possible to infer causality from the open ended statements, as responses tend to be directed towards the problem experienced and are therefore the products of the interaction.

The information obtained from the interview is more interpretable than that from the questionnaire. Difficulties cited tended to be unambiguous. The cause of the problem is unknown, however. The subsequent aided verbal protocol analysis presents information that is much easier to interpret with statements tending to be specific and unambiguous. By viewing the video tape, causes of the problems may be found, especially if the user is present.

8.1.3. Specific Hypotheses about the Efficiency of a User Based Evaluation Strategy.

Having examined the theory of efficiency and related the data to this theory it is now possible to articulate a set of explicit postulates addressing the determinants of the effectiveness of any particular proposed user based, human computer evaluation strategy (see Table 8.1). These postulates may be applied to any user interface evaluation methodology to predict its adequacy, however, it is acknowledged that these postulates need further testing and refining.

8.2. Suggestions to Improve the Evaluation Strategies Used in this Study.

Logged data is difficult to use in the field setting, due to the generation of large data sets resulting in low practical utility. One possible way of overcoming this difficulty would be to use a revolving buffer, in conjunction with a 'gripe' key command. With this approach designers may implement a logged data routine in the software that holds the key sequences and associated times of the last key string (say 100 keys).

This would act as a revolving buffer such that if the gripe key (a specified function key such as the F2 key) is not activated, the buffer key is replaced one for one; that is, as a new key is activated the key activated 100 keys previously is removed from the buffer. If, however, the user is having difficulty or feels that the system is confusing, or poorly designed, the gripe key is activated. When this key is activated the buffer area is now expanded to also encompass the next 100 keys activated by the user. When this buffer is full, now 200 keys long, it is automatically numbered and saved.

In conjunction with the buffer and "gripe key" an on line note pad could be useful. This would allow the user to enter notes about the programme as they occurred to them. Such a note pad would provide qualitative information about the saved key string. Through the use of this method the amount of redundant data should be drastically reduced.

Table 8.1. Explicit Hypotheses About the Efficiency of a User-Based Evaluation Method for Identifying Problem Areas Within a Human-Computer User Interface.

1. The probability of experiencing and perceiving problems during an interaction is related to extraneous factors such as the subjects' previous experience with similar systems, experience with computer based technology, and intelligence.

2. The probability of reporting a problem is directly related to the impact of the problem on the task undertaken.

3. The probability of reporting a problem is directly related to the number of problems encountered.

4. The probability of classifying an interaction as a problem is related to the effort required to overcome the problem.

5. The probability of reporting a problem is inversely related to the time before reporting the problem.

6. The probability of a user based evaluation method highlighting a problem area is related to the amount of perceptual error that may go undetected within the evaluation method.

7. The probability of an evaluator identifying a problem is related to the amount of subjectivity of classification.

8. Input by the subject may moderate the probability of the evaluator identifying a specific problem.

9. The usefulness of the information is related to the relationship between the elicited data and the identification and location of the causal reason.

The most likely productive area for the development of the questionnaire approach is in the form of a filtered, open answered, task oriented questionnaire. Assuming the evaluation tasks are ecologically valid, the task approach is preferable to the traditional attribute dimensional approach, for several reasons. A comment during a print out that "I couldn't find the command" is inherently more useful to the developer than a score of "2" under the "screens are cluttered" statement. The task related comment is specific to the situation, and therefore identifies not only the location of the problem, but also the nature of the problem. The dimensional score, however, only alludes to a problem area.

The task oriented approach may be more useful in obtaining relevant information when sample sizes are small. Such a task based approach is briefly outlined in Table 8.2. This approach could also be altered to cover attitudinal aspects of the functionality and acceptability of the system, and could prove useful for software development purposes.

Several methods can be proposed to overcome the recall problems associated with the interview approach. One method is to video the subjects, and then use this as a memory prompt. Doing this moves the interview closer to the subsequent aided verbal protocol analysis methodology. Here, as with the subsequent aided verbal protocol analysis, the problems of the associated times to undertake the evaluation and the complexity of analysis may in turn make the use of such an approach less attractive.

Another, as yet untested, interview procedure is the group format. Goldman (1962) has reported that the group interview has several advantages over the individual interview. In particular, in the domains of stimulating new ideas, the opportunity to observe group processes, an understanding of the temporal dynamics of attitudes and opinions, and spontaneity and candour.

It is interesting to speculate if such a format would have a facilitation effect. That is, one member of the group reporting a concern, or problem, may have the effect of reminding others of the problem or other similar experiences. If this was the case, the format would have several

Table 8.2. A Suggested Format for a Questionnaire for Assessing the Human-Computer Interface.

You have just finished using the _____ software package. Below are some questions about the package and the experiences you had. Could you please answer these questions as we need to know how you, the user, found the package. This information will then be used to try to improve the package and make it a better tool for your needs.

1. The first task you did was ______. Could you please explain any problems or things you felt could be improved to make doing this task easier or more pleasant.

.

.

1a. If you mentioned any problems in the above section could you please tell me exactly what happened, what you thought should have happened, and how you fixed the problem.

.

.

1b. If you have any suggestions on how to improve the package to make the task easier or more pleasant could you please put your thoughts below.

.

.

methodological advantages. One could obtain large quantities of user based data relatively easily. Through probing, the interviewer could also obtain information about the prevalence of a problem, and possible causal explanations. Furthermore, the 'brainstorming processes' of the group could be used to elicit a host of possible ways of improving the interface.

The disadvantage of the group interview approach, however, may be the problem of one, or a small sub-set, of the group monopolising the ideas and suggestions. Also, if a group obtains individuals with particularly good or particularly poor computing skills, some individuals may be less likely to volunteer information on what could be seen as their personal inadequacies. This problem could be partly controlled by good interviewing practices, and the judicial arrangement of the interview groups based on biographical data.

The major methodological difficulty associated with using the subsequent aided verbal protocol analysis process for evaluating the user interface, is the large amount of time required to reduce and analyse the data. It would seem that this is the one major disadvantage with this form of evaluation. One way of possibly alleviating this problem would be to use a taxonomy of errors in the classification process. A potential system for this has been outlined by Bagnara and Rizzo (1989). They have developed what they refer to as a "generic error modelling system" (GEMS) which they claim is better than previous error classification systems and can be used in the software development environment. The system essentially divides errors into the three categories of slips, rule based mistakes, and knowledge based mistakes. The process then takes into account such factors as error detection and recovery. A taxonomy of errors may reduce work by having the evaluator conduct the coding process at the time that the protocol analysis is generated, and thereby hopefully reducing the data reduction time. Reliability may be an issue, however, but no more than in any other evaluation reduction system used to reduce protocol data.

An attitudinal problem concerning the use of the subsequent aided verbal protocol analysis may prove to be a major obstacle. The author, when asked to advise on a user interface evaluation strategy to a group of computer science professionals encountered resistance just by the use of such phrases as "verbal protocol analysis" and "content analysis." It appeared that the use of these phrases was aversive. It was stated that they did not understand the processes of "content analysis" or "verbal protocol analysis." They therefore preferred the "objective" approaches of logged data collection. It is absurd to suggest that computer science professionals could not undertake a protocol analysis. Rather, it may appear that when striving to develop and utilise technical terms, psychology inadvertently has limited the application of its findings.

A similar phenomenon has been reported by Downs (1977) who had to rename the procedures of applied psychology in "practical" terms before the methods would be used by practising professionals. It is suggested that the terms "content analysis" and "verbal protocol analysis" are sufficiently aversive stimuli to impede their use by the practitioner. It may therefore be more appropriate to rename these terms along the lines of "guided interview."

8.3. Suggestions for Future Research.

The validity of the results of the present study rest upon many explicit and implicit assumptions. These range from the motivation of the subjects, to the robustness of the operationalisations.

The current investigation only considered QWERTY initiated interfaces. The generalisability of these findings to direct manipulation interaction and other more advanced interaction methodologies has not been addressed, and would make an exciting area for future research. It is possible that similar results would be found, however, as the same interaction process occurs irrespective of the interface.

Throughout the thesis prospective ways of improving the effectiveness of each evaluation method have been given. Specific areas of future research could include an in-depth examination of the relationship between various performance based measures and user impressions when using the logged data method. The computing industry, and engineering professionals, have been comfortable with such performance based measures because they appear "objective." This research has cast doubt upon the validity of assuming such data are objective in the classic sense. Furthermore, this study, in conjunction with others, has shown that actuarial logged data should be used cautiously as an indicator of usability.

An in-depth examination of the reliability of evaluator inferences based upon contextually based logged data also needs to be conducted. The relationship between evaluator inferences and those intended by users has not been accurately established.

The current study has also cast doubt upon the use of ratings in human-computer interaction questionnaires. Task oriented approaches may be more relevant to the software developer. The questionnaire will probably be used more in the future, due to its relatively low cost and its speed. Consequently, more effective methods for obtaining data in the questionnaire need to be researched.

The semi-structured group approach may be a useful area for future research when using the interview for evaluation. Comparisons between the information elicited using the single and group formats need to be conducted. Furthermore, this information needs to be compared to that obtained during the subsequent aided verbal protocol method.

For the subsequent aided verbal protocol to be used effectively, the analysis time must be drastically reduced. One option here is the development of a quick coding format. Using a taxonomy of errors in conjunction with a task based scenario may be a good way of approaching this problem.

For the development of user based usability standards, the evaluation should use more than one criterion, and the results of the final evaluation could then provide valuable information to the potential purchasers. Statements such as "independent user evaluations show that 92 percent of secretaries preferred this package to the present package they were using," or "Independent evaluations show that the average time for an inexperienced user to learn how to use the print sub-system was three minutes" are inherently useful to potential users and purchasers.

Such usability statements could be added by software engineers to the performance tests of the efficiency of software source code. This would introduce usability factors into the software development process, and allow them the chance of having equal status to software performance source code efficiency in the evaluation process. What is needed is an index for software usability. This could be achieved by defining the learning time of a task as one unit and thereafter describing the usability requirement in relation to this benchmark. This would also have the effect of deriving an industry standard, which could later be used to further develop empirically based standards.

8.4. Levels of Criteria: A Reframe of the Evaluative Outcomes.

Hamblin (1974) has expressed a concern with regard to collecting information about the changes caused by training, and suggests that training programme evaluation criteria can be delineated into five levels of training effects: reaction criteria (subjective impressions), learning criteria (measures of what has been learned on the course), behaviourial criteria (transfer to the work setting), organisational criteria (implementation of organisational goals and objectives) and the ultimate criteria which encompass the ultimate values of the course.

In general, each level of each criterion has different properties and uses, but it can be said that the higher the level of the criterion used, the more power the evaluation has. That is, learning criteria may show that learning has taken place during the course. However, the course is of little use if this learning is not transferred into the work place. Also, this change in behaviour must have beneficial outcomes in the form of organisational objectives.

It would appear that the levels of criteria evaluation paradigm may be one other way of viewing the outcome data gathered in the usability evaluation process. The subjective impressions of the users about the programme may be considered a reactions type criterion. Measures about what the users have learnt about the software may be used as an indicator of learning criteria and may infer the usability of the software. Performance data can perhaps be used as a form of behaviourial data. The organisational criteria may be recorded in the form of changes in the performance of the users performing work tasks in the organisational setting, and the ultimate criteria ideally would be seen in the form of a system that would be transparent to the user and facilitate desired organisational objectives.

If this analogy is accepted, it is useful to examine the obtained evaluation results in a comparative fashion to shed some light on the potency of the evaluation methods. That is, it may be the level of the criteria used that is an indicator of the power of the evaluation.

In the present study the logged data recorded the interaction at the behaviourial level. However, the analysis rested upon the reactions of the evaluator and therefore resulted in different outcome measures. It can be argued that the logged data used actuarially elicited learning criteria. The times to undertake the sub-tasks, over separate occasions, were in fact an indicator of learning how to use the programme. The analysis using the context of the programme in a playback mode, however, is perhaps a more valid indicator of the behaviour of the user. This relationship is moderated as the interpretation of these data rests on the reactions of the evaluator. Used in this way, the reactions of the evaluator cannot be regarded as behaviourial criteria. To move the evaluation data closer to behaviourial data, the user would need to be present during the data interpretation phase of the evaluation.

It can be argued that the questionnaire elicited purely reactions criteria. The subjective ratings cannot be used to infer learning or the behaviour of the user in any way. The open ended statements may hold more validity as an indicator of the behaviour of the user.

The interview initially looks to be firmly entrenched in the realm of reactions criteria. However, the fidelity of this approach is better than that of the questionnaire. This is due to many reasons. The first question addressed the user's approach to each sub-task. When articulating this the data that are being gathered are of a type of learning criteria. If the user can articulate how they undertook the task, they have learnt the command sequence, or have a cognitive model of the operation of the command sequence.

The second two questions, however, seem to be based largely in the field of reactions criteria. The reporting of the actual problems they encountered should be based upon behaviourial data, however. The reason for this is that if these problems cannot be verified then they are reduced to reactions type data. These data may be of a generally better quality than when the evaluator alone has to interpret the logs without the aid of the user.

The subsequent aided verbal protocol analysis also appears to be reactions criteria. The subject viewing themselves using the computer, and reciting the process, may result in outcome data closer to behaviourial criteria. Reactions are picked up but behaviour is also explicitly shown, due to the real time reenactment of the process.

By taking the levels of criteria approach, one can see how the data elicited can be viewed in a different light. If this paradigm is accepted then evaluators should use the more advanced criteria where possible. The relationship between the levels of criteria, and their validity to infer usability, will be moderated by several factors. In particular, this relationship is dependent upon software use. If an individual is purchasing a word processing package to write personal letters then the reactions may be more important than behaviourial criteria. Conversely, if the same software is purchased to be used as a word processing package in a business setting the behaviourial criterion of speed and the organisational criterion of sustained output, may be more important than reactions criteria. For the business person, it may be initial learning criteria that are important. Thus the software evaluator should explicitly define the objectives of the evaluation prior to the design. It is only by doing this that the evaluation will elicit relevant data.

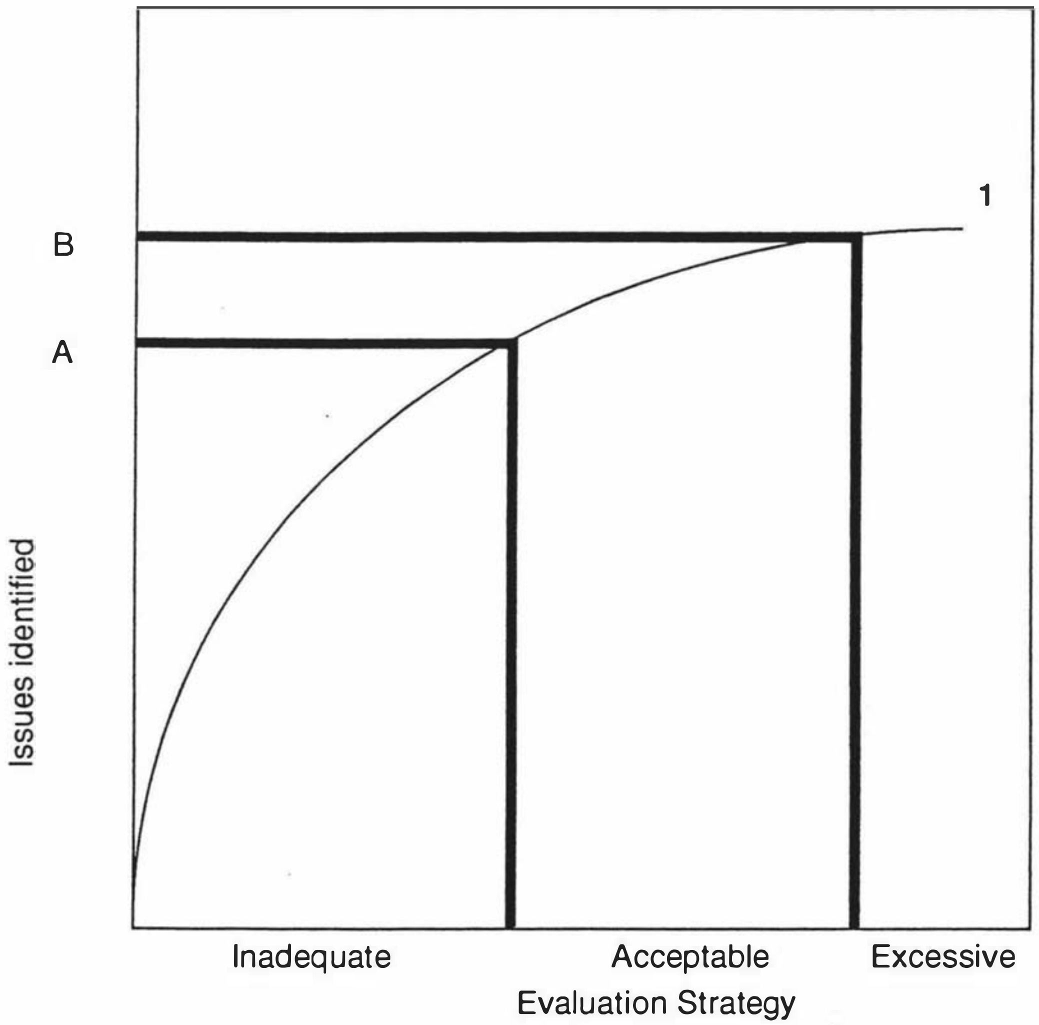
8.5. The Law of Diminishing Returns as Applied to the Software Usability Evaluation Process.

In the race to fill a market niche Christie and Gardiner (1990) has suggested that it may be better to release a product with a few problems, rather than miss the market opportunity, and become a second runner in a competitive environment. In this case, the political factors of the evaluation cycle may be the limiting factor. It may be that time and budget requirements mean that a quick and inexpensive evaluation method may have to be used to highlight the major problems. In this case the questionnaire alone may be the best evaluation strategy to use.

In other circumstances, a more expansive evaluation using protocol analysis could be used alone, or in conjunction with other evaluation strategies. The choice of evaluation strategies will be moderated by many external factors, of which time and budget considerations are only two. Such factors as the experience and knowledge of the evaluator and design team, the stage of the product's life cycle and user characteristics will also have an effect.

As noted by Mantei and Teorey (1988) and Christie and Gardiner (1990) the process of usability evaluation is set within a practical context. Consequently, it is unwise and unrealistic, to assume that in all cases a full scale usability evaluation will be conducted on all aspects of the software, and for all iterations in its development. Rather, the ideas of cost benefit analysis and the law of diminishing returns may apply. As Mantei and Teorey have noted, in some circumstances, it may not be feasible to conduct human factors evaluations because of the cost.

With regard to the present research, therefore, it would be unwise to suggest that an in depth subsequent aided verbal protocol analysis of the software should always be conducted. Rather the law of diminishing returns must be addressed (see Cole and Baumol, 1973). This law shows that the input to output ratio for products is not linear in nature. This can be hypothetically illustrated as shown in Figure 8.1. Line 1 shows a hypothetical line of the amount of usability issues identified by using usability evaluation strategies. Of importance to the discussion are lines A and B. Line A represents a hypothetical point of market acceptable usability. Above this point the product will be seen as having acceptable usability characteristics by the market, below this point the product will be seen as having unacceptable usability characteristics. It is also suggested that there may be a point at which further usability attributes



A = Point of minimal market acceptability

B = Point of maximum market acceptability

1 - Line of diminishing returns of problems identified

Figure 8.1. Law of Diminishing Returns Applied to the Evaluation Method Used and Issues Identified.

do not significantly impact on the value of the software. This hypothetical point is represented by the line B. That is, beyond this point extra usability attributes will not affect the impressions or performance of the software.

Of concern is the importance of the usability issues identified by each evaluation method. Where, on the line of diminishing returns of usability issues identified, will an evaluation strategy fall? Hypothetically, it may be that all the methods examined in this research will elicit information that may take a product beyond the point of market acceptable usability. In such a case, evaluators may use other criteria to chose an evaluation method (other than the number of problems identified). Ease of administration, cost and time factors may weigh heavily.

For comparative purposes, each problem has been considered equally important. In the present research this assumption is permissable. However, in the practical context this may not be the case. In the commercial context the impact of each problem is important. Here Karat (1988) notes that,

> "To a large extent the time and effort expended on an evaluation should reflect the impact of the decision which will result from it . . . Decisions with relatively little impact (such as the format of a given menu panel in a system) should not have more resource applied to them than they are worth." (Karat, 1988, p. 892).

this context it can be hypothesised that major usability problems will come apparent early in the usability evaluation and may be ependent of evaluation methods. This is supported by the number of blems highlighted by all four evaluation methods. For example, the idental deletion of records in the database programme, menu navigation problems within the database, and lack of task feedback within the word processor, to name but a few.

The extra problems identified may be minor and of little consequence, as far as market acceptability is concerned. This in turn adds a new dimension in the research of usability evaluation and should be examined further. The critical incident approach (Flanagan, 1954) may have some use in this context. With the critical incident technique, problems that are critical to successful functioning, can be identified and then rated as to the severity. Simple indices of efficiency of each evaluation method can then be derived.

8.6. The Software Evaluation Process and the Psychologist.

Several themes have become apparent throughout this dissertation. Specifically, the software evaluation process is essentially similar in structure to other forms of evaluation process, whether they be social or industrial. It is suggested that this evaluation research is therefore relevant to the software evaluation process. The problems, and theoretical considerations highlighted by Wortman's (1975) model are entirely relevant to the software industry. So too are the notions of the training feedback cycle, as discussed by Hamblin (1974). The iterative notions of educational action research (Cohen and Manion, 1980) and the concepts of formative and summative feedback as outlined by Scriven (1972). This does not serve to undermine the work of Williges et al. (1987), but rather reinforces the notion of applying the approaches and methods of psychology to the software industry.

t is surprising that psychologists have not become more involved in the oftware evaluation process. A large portion of the problems facing the oftware usability evaluation process were addressed in the social and ndustrial psychology literature some time ago. Why have software usability evaluators been left to reinvent a well researched process? A number of researchers have advocated the inclusion of psychologists in the design team. For example, Eberts (1987), Richardson (1987), Shneiderman (1988), and Carroll (1989) to name but a few. This provides support also for Meister (1986) who suggests that the methods and techniques are similar to those used in industrial psychology.

Furthermore, the same problems of criterion development apply to the usability construct, as they do to all other psychological constructs. The ideas of criteria relevancy, deficiency, and contamination (Blum and Naylor, 1968) are exceedingly relevant to the development of usability evaluation criteria. The notion of iterative refinement (Gould and Lewis, 1985; Gould, 1987,1988) rests upon the assumption of valid evaluation information; to the extent that the usability evaluations are relevant, deficient or contaminated, so too will the resulting alterations be relevant, deficient or contaminated. When developing evaluation criteria the paradigm of the reliable, valid and practical criterion as outlined by Landy (1989) seems the best avenue to follow.

References

Allen, R.B., & Scerbo, M.W. (1983). Details of command language keystrokes. ACM Transactions: Office Automation Systems, 1, 159-178.

Anderson, E.A. (1989). A heuristic for software evaluation and selection. Software-Practice and Experience, 19, 707-717.

Anderson, W.F., Friedman, B.J., & Murphy, M.J. (Eds.) (1977). Program evaluation and the management of organisations. In Managing Human Services. Washington D.C.: International City Management Association.

Bagnara, S., & Rizzo, A. (1989). A methodology for the analysis of error processes in human-computer interaction. In M.J. Smith & G. Salvendy (Eds.), Work with Computers: Organisational, Management, Stress and Health Aspects. Amsterdam: Elsevier.

Bainbridge, L. (1979). Verbal reports as evidence of the process operator's knowledge. International Journal of Man-Machine Studies, 4, 411-436.

Bainbridge, L. (1990). Verbal protocol analysis. In J.R. Wilson & E.N. Corlett (Eds.), Evaluation of Human Work: A Practical Ergonomics Methodology. London: Taylor & Francis.

Barnard, P.J., Hammond, N.V., Morton, J., & Long, J.B. (1981). Consistency & compatibility in human-computer dialogue. International Journal Of Man-machine Studies, 15, 87-134.

Benbasat, I., Dexter, A.S., & Masulis, P.S. (1981). An experimental study of the human-computer interface. Communications of the ACM, 24, 752-762.

Benjamin, R.I. (1982). Information technology in the 1990s: A long range planning scenario. MIS Quarterly, June, 11-35.

Bjorn-Anderson, N. (1988). Are 'human factors' human? The Computer Journal, 31, 386-390.

Blum, M., & Naylor, J. (1968). Industrial Psychology. New York: Harper & Row.

Boring, E.G. (1953). A history of introspection. Psychological Bulletin, 50, 169-189.

Borland International. (1987). Scotts Valley, Canada.

Bouchard, T.J. (1976). Field research methods: Interviewing, questionnaires, participant observation, systematic observation, unobtrusive measures. In M.D. Dunnette (Ed.), Handbook of Industrial and Organisational Psychology. Chicago: Rand McNally.

Bradburn, N.M., & Sudman, S. (1979). Improving Interview Method and Questionnaire Design. San Francisco: Jossey-Bass.

Branscomb, L.M. (1982). Bringing computing to people: The broadening challenge. Computer, 15, 68-75.

Brooks, F.P. (1987). No silver bullet: Essence and accidents of software engineering. IEEE Computer, 20, 10-19.

Brown, C.M. (1986). Human-Computer Interface Design Guidelines. New Jersey: Ablex.

Card, S.K., Moran, T.P., & Newell, A. (1983). The Psychology of Human Computer Interaction. New Jersey: Hillsdale.

Carey, J.M. (Ed.). (1988). Human Factors in Information Systems. New Jersey: Ablex.

Carroll, J.M. (1989). Evaluation, description and invention: Paradigms for human-computer interaction. In Y. Marshall (Ed.), Advances in Computers. Boston: Academic Press.

Carroll, J.M. & Rosson, M.B. (1985). Usability specifications as a tool in iterative development. In H.R. Hartson (Ed.), Advances in Human-Computer Interaction (Vol I). New Jersey: Ablex Publishing.

Chapanis, A. (1976). Engineering Psychology. In M. Dunnette (Ed.), Handbook of Industrial and Organisational Psychology. Chicago: Rand McNally.

Chapanis, A. (1982). Computers and the common man. In R. Kasschau, R. Lachman, & K. Laughery (Eds.), Information Technology and Psychology: Prospects for the Future. U.S.A.: Praeger.

hilton's Computing Series: Micro Software for Business. Infosource,Inc. (1984). Pennsylvania: Chilton Book Co.

Christie, B. & Gardiner, M. (1990). Evaluation of the human-computer interface. In J. R. Wilson & E. N. Corlett (Eds.), Evaluation of Human Work: A Practical Ergonomics Methodology. London: Taylor and Francis.

Christie, B., & McEwan, J. (1985). Introduction, In B. Christie (Ed.), Human Factors of Information Technology in the Office. Suffolk: Wiley.

Cohen, L., & Manion, L. (1980). Research Methods in Education. London: Groom Helm.

Cohill, A.M., Gilfoil, D.M., Pilitsis, J.V., & Carey, T. (1988). Measuring the utility of application software. In H.R. Hartson & D. Hix (Eds.), Advances in Human-Computer Interaction, (vol.2). New Jersey: Ablex.

Cole, C.L., & Baumol, W.J. (1973). Microeconomics: A Contemporary Approach. New York: Harcourt Brace Jovanovich.

del Galdo, E.M., Williges, R.C., Williges, B.H., & Wixon, D.R. (1987). A critical incident evaluation tool for software documentation. In L.S. Mark, J.S. Warm, & R.L. Huston (Eds.), Ergonomics and Human Factors: Recent Research. New York: Springer-Verlag.

DiNucci, D. (1985). Environmental Impact. PC World, December, 224-231.

Downs, S. (1977). Trainability Testing: A practical approach to selection. Training Information Paper, (No.11). London: M.S.O.

Draper, S.W., & Norman, D.A. (1985). Software engineering for user interfaces. IEEE Transactions on Software Engineering, SE-11, 252-258.

Dreyfuss, H. (1955). Designing for People. New York: Simon and Schuster.

Drury, C.G. (1987). Computerised data collection in ergonomics. In J. Wilson, E.N. Corlett, & I. Manenica (Eds.), New Methods in Applied Ergonomics. London : Taylor Francis.
N

Dzida, W. (1984). Towards an ergonomic design of software tools. In H. Schmidtke (Ed.), Ergonomic Data for Equipment Design. New York: Plenum Press.

Dzida, W., Herda, S., & Itzfeldt, W.D. (1978). User perceived quality of interactive systems. IEEE Transactions on Software Engineering, Se-4(4), 270-276.

Eason, K.D. (1988). Information Technology and Organisational Change. London: Taylor & Francis.

Eason, K.D. (1984). Towards the experimental study of usability. Behaviour and Information Technology, 3, 133-143.

Eberts, R. (1987). Human Computer Interaction. In P. Hancock (Ed.), Human Factors Psychology. Amsterdam: Elsevier.

Edmonds, E. (1987). Good software design: What does it mean? In H.J. Bullinger & B. Shackel (Eds.), Human-Computer Interaction - INTERACT'87. North-Holland: Elsevier.

Edmunds, R.A. (1985). The Prentice-Hall Standard Glossary of Computing Terminology. Englewood Cliffs, N.J.: Prentice-Hall.

Edwards, W., Guttentag, M., & Snapper, K. (1975). A decision-theoretic approach to evaluation research. In E. Struening & M. Guttentag (Eds.), Handbook of Evaluation Research. London: Sage.

Ericsson, K.A., & Simon, H.A. (1980). Verbal reports as data. Psychological Review, 87, 215-251.

Ericsson, K.A., & Simon, H.A. (1984). Protocol Analysis: Verbal Reports as Data. Massachusetts: MIT Press.

Farooq, M.U., & Dominick, W.D. (1988). A survey of formal tools and models for developing user interfaces. International Journal of Man-Machine Studies, 29, 479-496.

'iske, D.W. (1982). Asking Questions: A Practical Guide to Questionnaire Design. California: Jossey-Bass Inc.

anagan, J.C. (1954). The critical incident technique. Psychological Bulletin, 51, 327-358.

ley, J.D., Wallace, V.L., & Chan, P. (1984). The human factors of computer graphics interaction. IEEE GG&A, November 84, 13-48.

se, M. (1987). Human-computer interaction in the office. In C.L. Cooper & I.T. Robertson (Eds.), International Review of Industrial and Organisational Psychology 1987. Chichester: Wiley.

Frese, M., Ulich, E., & Dzida, W. (Eds.). (1987). Psychological Issues of Human-Computer Interaction in the Work Place. Amsterdam: Elsevier.

Gale, A. (1985). Assessing product usability: A psychophysiological approach. In B. Christie (Ed.), Human Factors of Information Technology in the Office. Chichester: Wiley.

Gardiner, M.M. (1986). Psychological issues in adaptive interface design. Proceedings of an IEEE Colloquium on Adaptive Interface Design. IEE Digest No. 1986/110, pp. 6/1-6/3.

Gardiner, M.M., & Christie, B. (1987). Applying Cognitive Psychology to User-Interface Design. Chichester: Wiley.

Gilb, T. (1985). The "impact analysis table" applied to human factors design. In B. Shackel (Ed.), Human-Computer Interaction - INTERACT'84. North Holland; Elsevier.

Glasnapp, D.R., & Poggio, J.P. (1985). Essentials of Statistical Analysis in the Behaviourial Sciences. Ohio: Charles E. Merril Publishing.

Goldenson, R.M. (Ed.). (1984). Longman Dictionary of Psychology and Psychiatry. New York: Longman Press.

ìoldman, A.E. (1962). The group depth interview. Journal of Marketing, 26, 61-68.

oodwin, N.C. (1987). Functionality and usability. Communications of the ACM, 30, 229-233.

Gould, J.D. (1987). How to design usable systems. In H.J. Bullinger & B. Shackel (Eds.), Human Computer Interaction - INTERACT'87. North-Holland: Elsevier.

Gould, J.D. (1988). How to design usable systems. In M. Helander (Ed.), Handbook of Human-Computer Interaction. North-Holland: Elsevier.

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

Grimes, J., Ehrlich, K., & Vaske, J. (1986). User interface design: Are human factors principles used? SIGCHI Bulletin, 17, 22-36.

Grudin, J., & Maclean, A. (1985). Adapting the psychophysical method to measure performance and preference trade-offs in human-computer interaction. In B. Shackel (Ed.), Human-Computer Interaction - INTERACT '84. North Holland: Elsevier.

Hamblin, A.C. (1974). Evaluation and Control of Training. New York: McGraw-Hill.

Hannigan, S. & Herring, V. (1986). The role of human factors inputs to design cycles: Deliverable A1.2b, HUFIT CODE: HUFIT/9-IAO-6/86.

Hanson, S.J., Kraut, R.E. & Faber, J.M. (1984). Interface design and multivariate analysis of UNIX command use. ACM Transactions on Office Information Systems, 2, 42-57.

Hartson, H.R., & Hix, D. (1989). Human-computer interface development: Concepts and systems for its management. ACM Computing Surveys, 21, 5-25.

Hietala, P. (1985). Combining logging, playback, and verbal protocols: A method for analyzing and evaluating interactive systems. In J. Rasmussen & P. Zunde (Eds.), Empirical Foundations of Information and Software Science III. New York: Plenum press.

Hoc, J.M. & Leplat, J. (1983). Evaluation of different modalities of verbalisation in a sorting task. International Journal of Man-Machine Studies, 18, 283-306.

Hodgson, G.M. & Ruth, S.R. (1985). The use of menus in the design of on-line systems: A retrospective view. SIGCHI Bulletin, 17, 16-22.

Howard, S., & Murray, D.M. (1987). A taxonomy of evaluation techniques for HCI. Human-Computer Interaction - INTERACT' 87, 453-459.

Howard, G.S., & Smith, R. (1986). Computer anxiety in management: Myth or reality? Communications of the ACM, 29, 611-615.

Isaac, S., & Michael, W.B. (1981). Handbook in Research and Evaluation. California: Edits Publishing.

ohnson, B., & Anderson, B. (1981). The Human-Computer Interface in Commercial Systems. Linkoping Studies in Science and Technology, Dissertations, No. 58, Linkoping University: Sweden.

Johnson, G.I., Clegg, C.W., & Ravden, S.J. (1989). Towards a practical method of user interface evaluation. Applied Ergonomics, 20, 255-260.

Karat, J. (1988). Software evaluation methodologies. In M. Helander (Ed.), Handbook of Human-Computer Interaction. North-Holland: Elsevier.

Kieras, D.E., & Polson, P.G. (1985). An Approach to the formal analysis of user complexity. International Journal of Man Machine Studies, 22, 365-394.

Kirakowski, J., & Corbett, M. (1990). Effective Methodology for the Study of HCI. North-Holland: Elsevier.

Kondakci, S. (1985). An Interactive Approach for Scheduling of Job Shops with Dual Constraints. Unpublished Ph.D. Thesis, Department of Industrial Engineering, Suny, Buffalo, New York.

Kopp, E.F. (1988). A plan for evaluating usability of software products. In J.M. Carey (Ed.), Human Factors in Management Information Systems. New Jersey: Ablex Publishers.

Kretz, F. (1985). Evaluation of interactive audiovisual applications: Some results and perspectives. In B. Shackel (Ed.), Human-Computer Interaction - INTERACT. North Holland: Elsevier.

Krippendorff, K. (1980). Content Analysis. An Introduction to its Methodology. Beverly Hills: Sage.

Krueger, W.G. (1989). Ergonomic evaluation of user-interaction by means of eye-movement data. In M.J. Smith & G. Salvendy (Eds.), Work with Computers: Organisational, Management, Stress and Health Aspects. Amsterdam: Elsevier.

Landauer, T.K. (1988). Research methods in human-computer interaction. In M. Helander (Ed.), Handbook of Human-Computer Interaction. North-Holland: Elsevier.

Landy, F.J. (1989). Psychology of Work Behaviour. Illinois: The Dorsey Press.

Laws, J.V. & Barber, P.J. (1989). Video analysis in cognitive ergonomics: A methodological perspective. Ergonomics, 32, 1303-1318.

Lea M. (1988). Evaluating user interface designs. In T. Rubin (Ed.), User Interface Design. Chichester: Ellis Horwood.

Luchins, A. (1957). Primacy-recency in impression formation. In C.I. Hovland (Ed.), The Order of Presentation in Persuasion. New Haven: Yale University press.

Lund, M.A. (1985). Evaluating the user interface: The candid-camera approach. In L. Borman & B. Curtis (Eds.), Human Factors in Computing Systems. Proceedings of the CHI'85 Conference. New York: ACM.

McClelland, I. (1990). Product assessment and user trials. In J.R. Wilson & E.N. Corlett (Eds.), Evaluation of Human Work: A Practical Ergonomics Methodology. New York: Taylor and Francis.

McDonald, J.E., Stone, J.D., Liebelt, L.S., & Karat, J. (1982). Evaluating a method for structuring the user-system interface. Proceedings of the Human Factors Society, 26th Annual Meeting.

MacLean, A., Barnard, P.J., & Wilson, M.D. (1985). Evaluating the human interface of a data entry system: User choice and performance measures yield different trade-off functions. People and Computers: Designing the Interface. Proceedings of the Annual Conference of the British Computer Society Human Computer Interaction Special Interest Group, University of East Anglia. Cambridge: Cambridge University Press.

Maguire, M. (1982). An evaluation of published recommendations on the design of man-computer dialogues. International Journal of Man-Machine Studies, 16, 237-261.

Maguire, M., & Sweeney, M. (1989). System monitoring: Garbage generator or basis for comprehensive evaluation system? In A. Sutcliffe & L. Macaulay (Eds.), People and Computers V: Proceedings of the Fifth Conference of the British Computer Society Human-Computer Interaction Specialist Group. New York: Cambridge Press.

Mais, C. & Giboin, A. (1989). Helping users achieve satisficing goals. In M.J. Smith & G. Salvendy (Eds.), Work with Computers: Organisational, Management, Stress and Health Aspects. Amsterdam: Elsevier.

Mantei, M.M., & Teorey, T.J. (1988). Cost/benefit analysis for incorporating human factors in the software life cycle. Communications of the ACM, 31, 428-439.

Marshall, C., Mcmanus, B., & Prail, A. (1990). Usability of product X-lessons from a real product. Behaviour and Information Technology, 9, 243-253.

Meads, J.A. (1985) Friendly or frivolous? Datamation, 31, 96-100.

Meister, D. (1986). Human Factors Testing and Evaluation. Amsterdam: Elsevier.

Moran, T.P. (1981a). An Applied Psychology of the User. ACM Computing Surveys, 13, 1-11.

Moran, T.P. (1981b). The command language grammar: A representation scheme for the user interface of interactive systems. International Journal of Man Machine Studies, 15, 3-50.

Morris, D., Theaker, C.J., Phillips, R., & Love, W. (1988). Human-computer interface recording. The Computer Journal, 31, 437-444.

Mosteller, F. (1981). Innovation and evaluation. Science, 211, 881-886.

Neal, A.S., & Simons, R.M. (1984a). Playback: A method for evaluating the usability of software and its documentation. In A. Janda (Ed.), Human Factors in Computing Systems. Amsterdam: Elsevier.

Neal, A.S. & Simons, R.M. (1984b). Playback: a method for evaluating the usability of software and its documentation. IBM Systems Journal, 23, 82-96.

Nickerson, R.S. (1986). Using Computers: The Human Factors of Information Systems. Cambridge: The MIT Press.

Nisbett, R.E. & Wilson, T.D. (1977). Telling more than we can know: Verbal reports on mental processes. Psychological Review, 84, 231-259.

Nunnally, J. (1967). Psychometric Theory. New York: McGraw Hill.

Oborne, D.J. (1985). Computers at Work: A Behaviourial Approach. Chichester: Wiley.

Olson, M.H., & Ives, B. (1981). User involvement in system design: An empirical test of alternative approaches. Information and Management, 4, 183-195.

Otte, F.H. (1984). Consistent user interfaces. In Y. Vassiliou (Ed.), Human Factors and Interactive Systems, Proceedings of the NYU Symposium on User Interfaces (May 1982). Norwood, N.J.: Ablex Publishing.

Parker, E. (1976). Social implications of computer/telecom systems. Telecommunications Policy, 1, 3-20.

Panko, R.R. (1988). End User Computing: Management, Applications, and Technology. USA: Wiley.

Penniman, W.D., & Dominick, W.D. (1980). Monitoring and evaluation of on-line information system usage. Information Processing & Management, 16, 17-35.

Praetorius, N. & Duncan, K.D. (1988). Verbal reports: A problem in research design. In L.P. Goodstein, H.B. Anderson, & S.E. Olsen (Eds.), Tasks, Errors, & Mental Models. London: Taylor & Francis.

Rackman, N., Honey, P. & Colbert, M. (Eds.). (1971). Developing Interactive Skills. London: Wellens Publishing.

Ravden, S.J., & Johnson, G.I. (1989). Evaluation Usability of Human-Computer Interfaces: A Practical Method. New York: Wiley.

Reisner, P. (1984). Formal grammar as a tool for analyzing ease of use: Some fundamental concepts. In J.C. Thomas & M.L. Schneider (Eds.), Human Factors in Computing Systems. New Jersey: Ablex.

Richardson, S. (1987). Operationalising usability and acceptability: A methodological review. In J. Wilson., E. N. Corlett., & I. Manenica (Eds.), New methods in Applied Ergonomics. London: Taylor & Francis.

Riecken, H. W. (1977). Principal components of the evaluation process. Professional Psychology. 392-410.

Roberts, T.L., & Moran, T.P. (1983). The evaluation of text editors: Methodology and empirical results. Communications of the ACM, 26, 265-283.

Robson, J.I., & Crellin, J.M. (1989). The role of user's perceived control in interface design, employing verbal protocol analysis. Applied Ergonomics, 20, 246-251.

Root, R.W., & Draper, S. (1983). Questionnaires as a software evaluation tool. In A. Janda (Ed.), Human Factors in Computing Systems. CHI'83 conference proceedings, Boston. P. 83-87.

Rosson, M, B. (1984). Effects of experience on learning, using, and evaluating a text editor. Human factors, 26, 463-475.

Rubin, T. (1988) User Interface Design. Chichester: Ellis Horwood.

Rushinek, A., & Rushinek, S.F. (1986). What makes users happy? Communications of the ACM, 29, 594-598.

Rust, J. & Golombok, S. (1989). Modern Psychometrics: The Science of Psychological Assessment. London: Routledge.

Salvendy, G. (1987). What we know and what we should know about human-computer interaction: Strategies for research and development. In G. Salvendy (Ed.), Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems. Amsterdam: Elsevier.

Salvendy, G., Sauter, S.L., & Hurrell, J.J. (Eds.). (1987). Social, Ergonomic, and Stress Aspects of Working with Computers. Amsterdam: Elsevier.

Sanders, M.S., & McCormick, E.J. (1987). Human Factors in Engineering and Design. U.S.A.: McGraw-Hill.

Scriven, M. (1972). The methodology of evaluation. In C.H. Weiss (Ed.), Evaluating Action Program:Readings in Social Action and Education.Boston:Allyn and Bacon.

Shackel, B. (1986a). IBM makes usability as important as functionality. The Computer Journal, 29, 475-476.

Shackel, B. (1986b). Ergonomics and usability. In M.D. Harrison & A.F. Monk (Eds.), People and Computers: Designing for Usability. London: Cambridge University Press.

Shneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Massachusetts: Addison-Wesley.

Shneiderman, B. (1988). We can design better user interfaces: A review of human-computer interaction styles. Ergonomics, 31, 699-710.

Shneiderman, B. (1984). The future of interactive systems and the emergence of direct manipulation. In Y. Vassiliou (Ed.), Human Factors in Interactive Systems. Proceedings of the NYO Symposium on User Interfaces (May 1982). Norwood, N.J.: Ablex Publishing.

Shouksmith, G. (1978). Assessment Through Interviewing. Oxford: Pergomon press.

Simes, D.K. & Sirsky, P.A. (1985). Human factors: An exploration of the psychology of human-computer dialogues. In H.R. Hartson. Advances in Human-Computer Interaction (Vol 1). USA: Ablex Publishing.

Simon, H., & Stedry, A. (1969). Psychology of economics. In G. Lindzey & E. Aronson (Eds.), Handbook of Social Psychology. (Vol. 5). Reading Mass.: Addison-Wesley.

Sinclair, M.A. (1990). Subjective assessment. In J.R. Wilson & E.N. Corlett (Eds.), Evaluation of Human Work. London: Taylor & Francis.

Smith, M.C. (1982). The In-Basket Test as Practical Psychology. Unpublished Doctorial Dissertation. Massey University: New Zealand.

Smith, S.L. (1986). Standards versus guidelines for designing user interface software. Behaviour and Information Technology, 5, 47-61.

Smith, S.L., & Mosier, J.N. (1985). The user interface to computer-based information systems: A survey of current software design practice. In B. Shackel (Ed.), Human Computer Interaction, Amsterdam: Elsevier.

Smith, S.L., & Mosier, J.N. (1986). Design Guidelines for User-System Interface Software. Bedford, M.A.: MITRE Corporation.

Spinas, P. (1989). User oriented software development and dialogue design. In M.J. Smith & G. Salvendy (Eds.), Work with Computers: Organisational, Management, Stress and Health Aspects. Amsterdam: Elsevier.

SPSSX (1986). Chicago: SPSS Inc.

Suchman, E.A. (1967). Evaluation Research. New York: Sage.

Sutcliffe, A. (1988). Human-Computer Interface Design. London: MacMillan Education Ltd.

Sweeney, M., & Dillon, A. (1987). Methodologies employed in the psychological evaluation of H.C.I. In H.J. Bullinger & B. Shackel (Eds.), Human-Computer Interaction-INTERACT '87. North-Holland: Elsevier.

Tabachnick, B.G., & Fidell, L.S. (1989). Using Multivariate Statistics (2nd ed.). New York: Harper & Row.

Theaker, C.J., Phillips, R., Frost, T.M.E., & Love, W.R. (1989). HIMS: A tool for evaluations. In A. Sutcliffe, & L. Macaulay. (Eds.), People and Computers V. Proceedings of the Fifth Conference of British Computer Society Human-Computer Interaction Specialist Group. Cambridge: Cambridge University Press.

Thimbleby, H.T. (1984). Generative user-engineering principles for user interface design. In B. Shackel (Ed.), Human-Computer Interaction - INTERACT'84. North Holland: Elsevier.

Toffler, A. (1980). The Third Wave. London: Collins.

Turbo Pascal, Database Toolbox. (1987) Borland International: Scotts Valley, Canada.

Turbo Pascal, Editor Toolbox (1987). Borland International: Scotts Valley, Canada.

Turbo Pascal, Version 4. (1987). Borland International: Scotts Valley, Canada.

Tyldesley, D.A. (1988). Employing usability engineering in the development of office products. The Computer Journal, 31, 431-436.

Tynan, P.D. (1985). Randomly sampled self-report method for collecting field data on human-computer interactions. In B. Shackel (Ed.), Human-Computer Interaction - INTERACT'84. North Holland: Elsevier.

Weiss, C.H. (1975). Interviewing in evaluation research. In E.L. Struening & M. Guttentag (Eds.), Handbook of Evaluation Research. Beverley Hills: Sage.

Wexelblat, R.L., (1981). Design of systems for interaction between humans and computers. In R.D. Parslow (Ed.), BCS'81 Information Technology for the Eighties: Heyden & Sons Ltd.

Williges, R.C., Williges, B.H., & Elkerton, J. (1987). Software interface design. In G. Salvendy (Ed.), Handbook of Human Factors. New York: Wiley.

Wortman, P.M. (1975). Evaluation research: A psychological perspective. American Psychologist. May, 562-755.

Yamagishi, N., & Azuma, M. (1987). Experiments on human-computer interaction evaluation. In G. Salvendy (Ed.), Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems. Amsterdam: Elsevier.

Yang, Y. (1989). Survey steered design. Behaviour and Information Technology, 8, 437-459.

Yoder, E., McCracken, D. & Akscyn, R. (1985). Instrumenting a human-computer interface for development and evaluation. In B. Shackel (Ed.), Human-Computer Interaction - INTERACT '84. Amsterdam: Elsevier Science Publishers.

Zipf, G.F. (1965) Human Behaviour and the Principle of Least Effort: An Introduction to Human Ecology. New York: Hafner.

Appendices

Appendix 1. Experimental Subject Contact Procedure.

1. All subjects were contacted by telephone prior to the experiment. A telephone list was required for this.

2. After being contacted it was confirmed that they had agreed to be a subject.

3. Subjects were told the following:

> The experiment is aimed at examining different methods of evaluating the usability of business type computer software. The experiment will begin by giving a general overview of the study. This will be followed by an overview of the piece of software they will be using. Subjects will then complete a set of tasks on the computer, that will take about half an hour. After completion of these tasks, subjects will complete an evaluation of that piece of software. This evaluation will take about half an hour to complete. However, it may require coming back at a different time to do this. If so, a mutually convenient time will be arranged. The total time commitment should be under two hours. All data are treated as strictly confidential.
>
> It was emphasised that no previous computer knowledge or experience was required.

4. If necessary, the above was elaborated on to the subject's satisfaction.

5. Subjects were then be given a list of times that the experiment would run and asked to choose which time was most suitable.

R.D. Henderson.

Appendix 2. Informed Consent Form Used in the Study.

INFORMED CONSENT FORM
Massey University

Title of Investigation: Business Software Evaluation Study.
Investigator: Ronald D. Henderson.
Department of Psychology.

Date:____

This is to certify that I,_____________, hereby agree to participate as a volunteer in a scientific investigation as an authorised part of the psychology research programme of the Department of Psychology, Massey University, under the supervision of Dr. Mike Smith.

- The investigation, and my part in the investigation have been defined and fully explained to me by __________, and I understand his/her explanation.
- I have been given an opportunity to ask whatever questions I may have had, and all such questions and inquires have been answered to my satisfaction.
- I understand that I am free to deny any answer to specific items or questions in interviews or questionnaires.
- I understand that any data or answers to questions will remain confidential with regard to my identity.
- I understand that I will not be informed as to which evaluation group I will be in until I have completed the computer task.
- I FURTHER UNDERSTAND THAT I AM FREE TO WITHDRAW MY CONSENT AND TERMINATE MY PARTICIPATION AT ANY TIME.

____ ____________________
Date. Subjects Signature.

I, the undersigned, have defined and fully explained the investigation to the above subject.

____ ____________________
Date. Investigator's Signature.

Appendix 3. General Overview Information Sheet Used in the Study.

General Introduction

Thank you for agreeing to participate in this study. As has been explained to vou over the telephone, this part of the study will take approximately a half hour to complete. Some of you will be required to come back, at a time convenient to you, and complete a second part of the study, which will take approximately half an hour.

As more and more people are using computers in their everyday work, there has been an increasing demand for software that is easy to use and understand. It is therefore vital for commercial software developers to evaluate the ease of use of software during the initial stages of development. Techniques that have been advocated to do this have included; interviews, questionnaires, logged data, and verbal protocol analysis.

The purpose of the study is to examine the strengths and deficiencies of these evaluation methods across the three software domains of word processing, spreadsheet management and database management.

Each of you will be required to complete one only evaluation methodology on one only software package. In this case you will all be using a __________ programme. You will be given a brief introduction to the computer, followed by the concepts involved with this type of programme. Following this, you will then perform a series of tasks which will be followed by the completion of an evaluation of the programme.

Those in the interview and verbal protocol groups will need to make a time, that is convenient to them, to come back and complete that part of the study. Those in the questionnaire group will complete a questionnaire before leaving, those in the logged data group will be free to leave as the data will already have been collected by the computer.

You have not been told which group you are in because individuals may focus there attention differently, depending on which evaluation they were to later complete. That is, if you know that you are going to complete a questionnaire you may focus on different things than if you knew you were to have an interview.

Please remember that the sole purpose of this study is to examine the different methods of evaluating software, it is these methods that are being evaluated, not your performance on the task.

Are there any questions at this stage.

Appendix 4. Introduction to the Spreadsheet Information Sheet Used in the Study.

Introduction to the Spreadsheet

Source: Chilton's Software Directory (1984).

The microcomputer revolution in business would probably not have occurred without the advent of spreadsheet software. The actual functions of spreadsheet software are quite simple, yet the concept behind the idea is brilliant.

The best way to think of spreadsheet software is to picture a giant matrix, or grid, of rows and columns. A partial blank spreadsheet is illustrated below, as Table one. The columns in this illustration are labelled with letters, and the rows are labelled with numbers. Some spreadsheets use numbering systems, such as R1 and C1, to indicate the Row or Column. All of the locations (at any given intersection of a row and a column) in the matrix are referred to as "cells" and are named by their respective positions. For example, the cell that is found at the intersection of the third column and fourth row is cell "C4."

Table one

Partial blank spreadsheet.

:	A :	B :	C :	D :	E :	F :	G :
1:							
2:							
3:							
4:							
5:							
6:							
7:							
8:							

Here is where the magic of spreadsheet software comes in. Each cell can contain numbers or letters and words. Also, each cell can be defined as a function of any other cell, or group of cells. This means that you can make the value of a given cell dependent upon the value of other cells. For example, suppose you have one number in cell B2 and another number in cell B3. If you wish to find the average of these two numbers and place the answer in cell B4, you would enter the formula [(B2 + B3)/2], into cell B4. The spreadsheet would automatically reflect the average in cell B4.

This does not seem an exceedingly difficult task. However, if instead of a small matrix, such as in Table one, you were dealing with a matrix of several hundred columns and rows and wished to add a constant to each Figure, say 10% for G.S.T., instead of redoing the entire sheet you could just enter a formula.

Figure one is a schematic outline of the spreadsheet you will be using in this study. You can get to the main menu by pressing the "/" key. You enter text in the spreadsheet in much the same way as you enter text on a typewriter, and most of the keys on the keyboard behave in the similar fashion. However, there are also many differences.

The highlighted block will always indicate which cell information will be entered. This information will be entered by pressing the ENTER key. You can move the highlighted block by using the arrow, PgDn, PgUp, Home and End keys on the numeric keypad to the right of the keyboard. You can also correct mistakes by retyping the highlighted block. Both rows and columns may be inserted and deleted. Rows and columns may also be formatted in the desired fashion.

Please attempt all of the tasks below but **remember, do not worry if you have problems, the aim of the experiment is to evaluate the software, not you.**

Appendix 5. Spreadsheet Experimental Task Used in the Study.

Spread Sheet Task

1/ Create a spreadsheet and enter the following headings and values. The numbers down the left hand side represent the rows that should be used. The letters along the top represent the columns that should be used.

A B C D E F G

1			Pie Sales			
2						
3		Total	Month	Income	Profit	P/U
4						
5	Appul Light	4000	2009	16000	6000	1.50
6	Carob Crunch	3600	1000	14000	4000	1.11
7	Coconut Crust	2800	700	12000	3000	1.07
8	Kumquat	600	300	3600	1000	1.66
9	Pure Peach	1000	400	3900	2000	2.00
10	Strawberry	400	200	660	400	1.00
11	Very Berry	2000	1029	4000	1000	0.50
12	Yogurt Crunch	300	128	300	200	0.66

2/ Next save the spreadsheet as PIESALES. If the programme states "File already exists. Do you want to overwrite it" press the "Y" key for yes.

3/ Move to the menu system and obtain a printout of the file by pressing the ENTER key. Answer "N" to the question "Do you want to print in 132 columns". Answer "Y" to the question "Print the boarder?"

4/ Clear the spread sheet.

5/ Load PIESALES.

6/ Insert a row between Pure Peach and Strawberry on the spread sheet.

7/ Enter the following data in the new row.

Squash Bloss 400 200 1600 400 1.00

8/ Save PIESALES as before.

9/ Clear the spreadsheet.

10/ Load PIESALES.

11/ Change Appul light to read Apple light.

12/ Delete the Carob Crunch row.

13/ Add the following row at the end of the spreadsheet.

Yogurt Yummy 400 600 2000 800 2.00

14/ Print the spreadsheet as before.

15/ Save PIESALES as before.

16/ Clear the spreadsheet.

17/ Load PIESALES.

18/ Delete the "Total" column.

19/ Obtain a printout as before.

20/ quit the programme.

When you have finished please notify the experimenter.

Appendix 6. Introduction to the Word Processing Information Sheet Used in the Study.

Introduction to Word Processing

Source: Chilton's Software Directory (1984).
Tubo Pascal, Editor Toolbox (1987).

Word processing is the entering, storing, editing, and retrieving of information - words or numbers or both. A word processor is a computerised version of a typewriter, in which everything that is typed in is saved on some kind of medium (usually a floppy disk) for present or later use. Stored information can be printed out, sent to another computer via a modem and telephone lines, merged into another file or files containing other information, or reworked when more creative energies emerge.

In the business context the micro based word processor offers several advantages over the more conventional type writer. Specifically, there is an increasing trend for word processor packages to be imbedded in larger integrated packages. This offers more control over data and information manipulation.

There is more flexibility in document editing and presentation with the output being generally more attractive. Also when word processing is done by the originator of the words, there is a much faster turnaround than typing performed by a secretary.

Features offered by most word processor packages typically include: block copy, merging and moving, rapid cursor control, spelling verification, document formatting, and varying choices of presentation.

In the following experiment you will be performing several tasks on a micro based word processor. Figure one is a schematic outline of the word processor. To get to the main menu you press the "F10" key. You enter text in the word processor in much the same way as you enter text on a typewriter, and most of the keys on the keyboard behave in the same fashion (press Enter to end each line, for example). But there are many important differences as well.

The cursor always indicates where new text will be entered. You can move the cursor in a number of ways, and the commands to do so are described in Table one. You can correct mistakes quickly and easily using the delete commands; you can copy and move text with the block commands; you can locate a particular string of text with the find command, and optionally replace it with another using the find-and replace command; and in most cases you can even undo your last few changes with the restore line or undo commands.

For a quick glance at all of the commands and their respective keystrokes refer to Table one. In this Table the word processor menu commands are displayed in boldface under the menu in which they appear. Non-menu commands are printed in plain type. They are broken down into basic movement commands, basic editing commands, and commands related to a particular menu. These last are printed after the menu commands under the appropriate menu.

Please attempt to complete the tasks. **Remember do not worry if you have problems, the aim of the experiment is to evaluate the software, not you.**

Appendix 7. Word Processing Experimental Task Used in the Study.

Word Processing Task

1/ Please type in the following passage exactly as it is written here.

M. Simon,
22 Farview Drive,
Wellington.

Ms. Simon, as you've requested, here is a bulletin reflecting our annual performance and outlining projections for next year's performance.

Good news from the market analysts. Sales in the Organic Pie line are booming! Yogurt Yummy is really taking off, as are Very Berry and Appul Light.

By the way, we've changed the spelling of Granola Pudding Delight, as you've suggested.

Elmer Body in the Crust Department won this month's Suggestion Box award. He'll get a month's supply of Appul Pie Filling as his reward for suggesting that we open up a Pie Tasting Room. Such a tasting room could pep up our national sales, as tourists carry news of our fabulous pies all over New Zealand.

All in all, it's been a good year, and projections show that next year should be even better. With the new factory in place, our extra capacity should make us able to keep up with the stratospheric demand for Mom's Appul Pies that we anticipate.

2/ Please save the data with the following file name: Appul.

3/ Close the file.

4/ Open the Appul file.

5/ Please make the following alterations.

1. Please add the following to the end of the third paragraph:

   Calling it Granola Pudding Delight adds pizazz to our lineup and will surely gain us a strong foothold in the youth market.

2. In the fourth paragraph change the name of Elmer Body to read Elmer Bodey.

3. Delete the second paragraph.

4. Delete the sentence "Good news from the market analysts."

5. Move the second paragraph to the bottom of the passage.

6. In all cases change Appul to Apple.

6/ Obtain a printout of the passage.

7/ Exit the programme.

Appendix 8. Introduction to the Database Information Sheet Used in the Study.

Introduction to Database Management Systems

Computers have always had the capacity to store large amounts of data. Initially, data were stored on magnetic tapes. The main drawback to tape storage is its slow processing time. To examine a record at the end of the tape, the program must read past all the records before it. The development of direct access devices (the disk drive, diskettes, and disk packs) brought an additional feature to the computer - the ability to access the data very quickly. But although the hardware technology has advanced, it has taken a while for the software to catch up.

When software and system designers turned their attention to developing a new technology to take advantage of disk capabilities, the result was the database concept. A database integrates separate, but related, data files into a single source of information and reduces the data redundancy that inevitably occurs when there are many separate files.

In the following experiment you will be using a database system that has been designed to keep track of a company's customers. Information is stored in records, where a record contains all the information relevant to one customer. Within each record is a set of fields. A field contains one aspect of information, for example an address, or phone number. Figure 1 illustrates this concept.

Code: Date:

First Name:
Last Name:

Company:

Address 1:
Address 2:

Phone: Extension:
Remarks 1:
Remarks 2:
Remarks 3:

Figure 1: Blank record illustrating the fields in the database.

The cursor will indicate where information will be entered into the record. You may move between the fields in a record by pressing either the ENTER or ARROW keys. Information can also be updated, deleted, sorted, and printed.

Please attempt all tasks, do not worry if you have problems, it is the software that is being evaluated, not you.

Appendix 9. Database Experimental Task Used in the Study.

Database task

Please add the following records to the database.

Code: 100 Date: 1.4.89

First Name: Karen
Last Name: Smith

Company: Home Bake.

Address 1: 105 Gray Street,
Address 2: Palmerston North.

Phone: 75 149 Extension: 8589

Remarks 1: Call twice per week.
Remarks 2: Main order product # 105.
Remarks 3: If not available ask for John.

Code: 206 Date: 2.5.88

First Name: Ron
Last Name: Brown

Company: Fresh Bake.

Address 1: 22 Mulgrave Street,
Address 2: Palmerston North.

Phone: 62 907 Extension: 71

Remarks 1:
Remarks 2:
Remarks 3:

2/ Move to the List section. List the records on the screen, using the name option when it becomes available.

3/ Move to the Find menu and Find the record with the code number 206.

4/ Update this record by adding the following information.

Remarks 1: Call Daily.
Remarks 2: Product 100.
Remarks 3: If not available ask for Jean.

5/ Move back to the Main menu and choose the List option.

6/ List the records on the Printer using the Code option when it becomes available.

7/ Move to the Find option. Locate the record with the Name Karen Smith.

8/ Change the first remark to read.

Remark 1: Call twice per month.

9/ Move to the Main menu, then Add the following record:

Code: 207 Date: 4.2.88

First Name: Helen.
Last Name: Sims.

Company: Marine Cakes.

Address 1: 152 Main Street,
Address 2: Napier.

Phone: 47257 Extension:

Remarks 1: Call 10th & 25th of each month.
Remarks 2: Product 105, 242, 203.
Remarks 3:

10/ Move to the List section and obtain a Printout by Name.

11/ Move to the Find section and delete record number 206.

12/ Move to the main menu and then Find record with the name Karen Smith.

13/ Change the extension number to read: 8599.

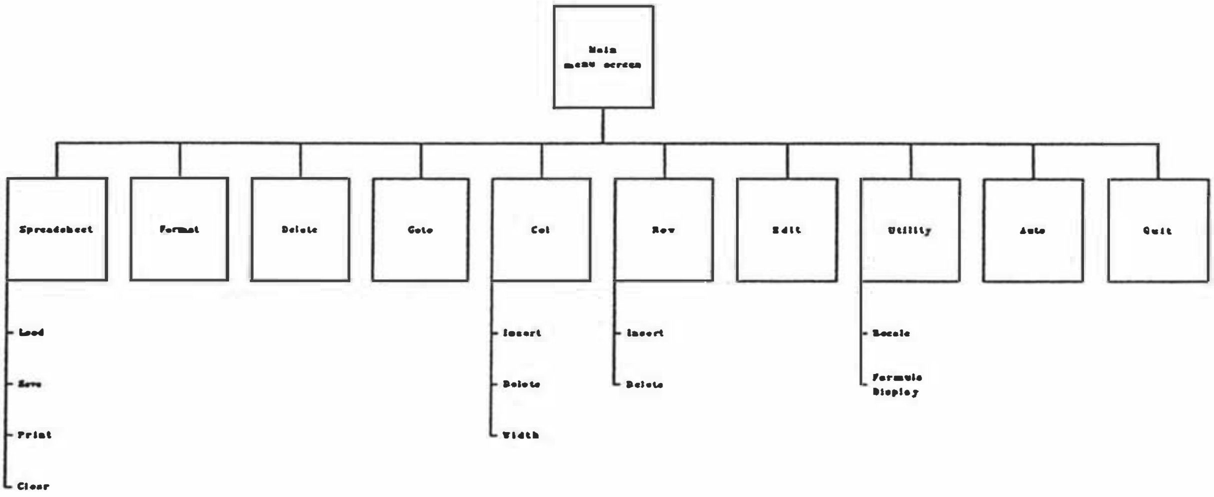
14/ Delete the record with the Code number 100.

15/ Move to the List system and List the records to the Printer using the Unsorted option.

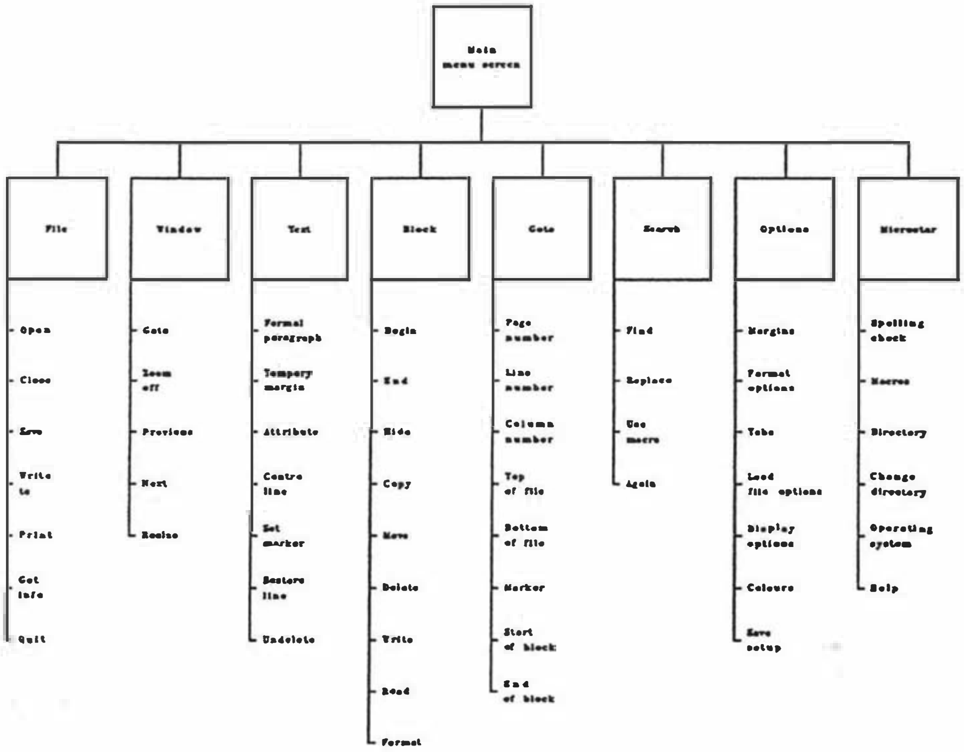
16/ Move to the Find section, Locate and then Delete the record with the code number 207.

17/ Quit the program.

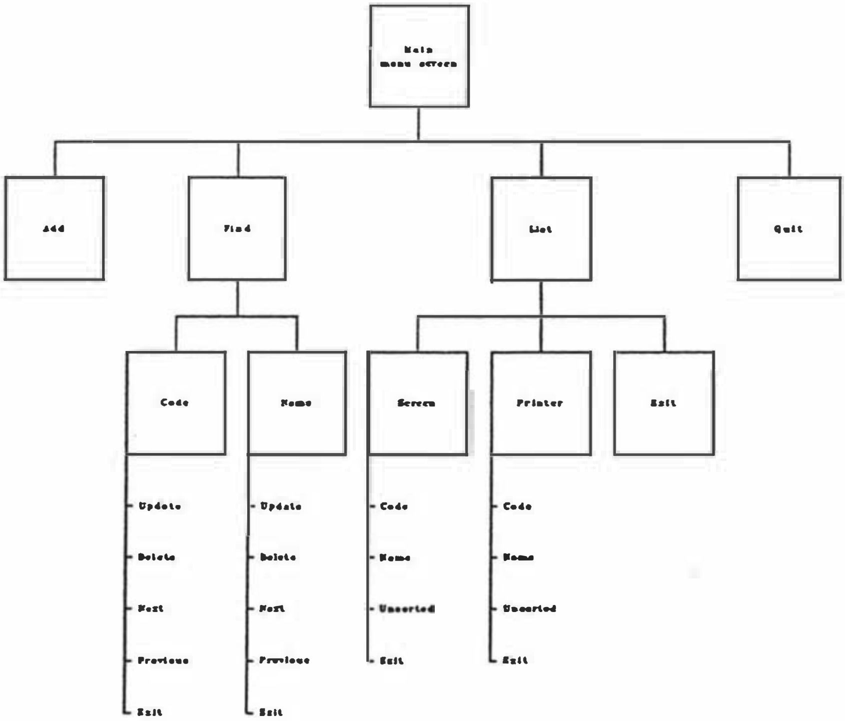
Appendix 10. Diagrammatic Outline of the Spreadsheet Used in the Study.



Appendix 11. Diagrammatic Outline of the Word Processor Used in the Study.



Appendix 12. Diagrammatic Outline of the Database Used in the Study.



Appendix 13. Base Logged Data Collection Routine.

Note: Some cosmetic portions of this program have been removed. These included error routines and screen boundaries

```
program Log (input, output);
uses crt, turbo3,dos;

const max = 1500;
         blank = '   ';

type     datetime  = record
              hour,min,sec,sec100:word;
         end;

         data = record
            First_name: string [10];
            Last_name : string [30];
            Gender    : char;
            wp        : integer;
            ss        : integer;
            db        : integer;
            gp        : integer;
            overall   : integer;
            keys      : array [1..max] of integer;
            number    : string[3];
            times_1   : array[1..max] of datetime;
         end;

var      key: char;
         enter :data;
         i:integer;
         file1 :file of data;

procedure check (var key:char);
begin
   if key = #0 then
   begin
      key:=readkey;
      enter.keys[i] := ord (key)+1000;
   end
   else
      enter.keys[i]:= ord (key);
   end;
```

```
procedure convert;
begin
    for i:= 1 to 3 do
    begin
        key:= readkey;
        check(key);
        with enter.times_1[i] do
        begin
            gettime(hour,min,sec,sec100);
        end;
    end;
end;

procedure overview;
begin
    clrscr;
    frame_it (1,1,78,25);
    gotoxy (10,2); Writeln ('Please answer the following questions');
    gotoxy (10,4); writeln ('You are going to be asked a few questions about how');
    gotoxy (10,5); writeln ('confident you feel using various computer programmes.');
    gotoxy (10,6); writeln ('In each case rate yourself on a seven point scale as to');
    gotoxy (10,7); writeln ('how confident you feel about using the programme.');
    gotoxy (10,8); writeln;
    gotoxy (10,9); writeln ('1 = not confident, 4 = quite confident, 7 = very confident');
    writeln;
    just_pause;
end;

procedure confidence;
begin
    gotoxy (10,12);
    writeln ('How confident do you feel using a Word Processor (1 - 7) ......');
    gotoxy (73,12); test_integer (73,12,enter.wp,1,7);
    gotoxy (10,14);
    writeln ('How confident do you feel using a Spread Sheet (1 - 7) .......');
    gotoxy (73,14); test_integer (73,14,enter.ss,1,7);
    gotoxy (10,16);
    writeln ('How confident do you feel using a Data Base (1 - 7) ..........');
    gotoxy (73,16); test_integer (73,16,enter.db,1,7);
    gotoxy (10,18);
    [illegible]
    [illegible]

    gotoxy (10,20);
    writeln ('In general how confident do you feel using computers (1 - 7) .');
    gotoxy (73,20); test_integer (73,20,enter.overall,1,7);
end;
```

```
procedure get_data;
begin
    clrscr;
    frame_it (1,1,78,25);
    gotoxy (10,2); writeln (’Please answer the following questions’);
    gotoxy (10,4); writeln (’Enter your first name ............’);
    gotoxy (48,4); readln (enter.First_name);
    gotoxy (10,6); writeln (’Enter your Surname ................’);
    gotoxy (48,6); readln (enter.Last_name);
    gotoxy (10,8); writeln (’Enter your gender (M/F) ...........’);
    gotoxy (48,8); test_in (48,8,enter.gender);
    gotoxy (10,10); writeln (’Enter the number you were assigned’);
    gotoxy (48,10); readln (enter.number);
    overview;
    confidence;
end;


procedure save_file (information: data);
var    filename: string[12];
begin
    filename:=’F’+information.number +’.dat’;
    assign(file1,filename);
    rewrite(file1);
    write(file1,information);
    close(file1);
end;


begin
    get_data;
    convert;
    save_file (enter);
    readln;
end.
```

Appendix 14. Software Usability Questionnaire Used in the Study.

Software Usability Form

Instructions

The following questionnaire is designed to find out what you feel about a software programme. There are eight sections in all, with each section being divided into two parts.

Part one is a series of statements about the programme. In each case think about the programme. Then rate the programme on the seven point scale by placing a circle around the number that applies (1 = Low, 7 = High).

If the question is NOT APPLICABLE then place a circle around the NA to the right of the scale. Also, if you DO NOT UNDERSTAND the question please place a circle around the letters DU to the far right of the scale.

For example

	Low High		
The package was enjoyable	1--2--3--4--5--6--7	NA	DU

In this case if you felt the package was enjoyable you would circle a score in the high end of the scale, if you felt the package was not enjoyable you would circle towards the low end, and if the question was not applicable to the programme you would circle the NA. If you did not understand the question you would place a circle around the letters DU.

Part two is an open ended section that allows you to make specific comments and suggestions about the package.

Please answer all questions.

Please print your name here:____________________________

Please print the number you have been assigned:__________

A. Programme Self-Descriptiveness

Low High

1 Instructions describing the tasks are clear. 1–2–3–4–5–6–7 NA DU

2 Meaningful prompts are provided. 1–2–3–4–5–6–7 NA DU

3 The programme tells you what to do next. 1–2–3–4–5–6–7 NA DU

4 Instructions always have a consistent tone. 1–2–3–4–5–6–7 NA DU

5 Instructions have a constant position on the display. 1–2–3–4–5–6–7 NA DU

6 Instructions for commands or choices are clear. 1–2–3–4–5–6–7 NA DU

7 Instructions for getting more help are clear. 1–2–3–4–5–6–7 NA DU

8 The programme explains requests to you if and when necessary. 1–2–3–4–5–6–7 NA DU

9 Instructions for correcting errors are clear. 1–2–3–4–5–6–7 NA DU

10 Decision aids are given if the task cannot be done as desired. 1–2–3–4–5–6–7 NA DU

11 The presentation of what the programme can do is clearly arranged. 1–2–3–4–5–6–7 NA DU

12 You can become thoroughly acquainted with the programme without human assistance. 1–2–3–4–5–6–7 NA DU

13 Commands that are available are shown. 1–2–3–4–5–6–7 NA DU

14 The programme always provides a command glossary. 1–2–3–4–5–6–7 NA DU

15 The programme shows valid input choices. 1–2–3–4–5–6–7 NA DU

16 Default choices are shown. 1–2–3–4–5–6–7 NA DU

17 The programme described what had to be done. 1–2–3–4–5–6–7 NA DU

18 Lengths of entry fields are shown. 1–2–3–4–5–6–7 NA DU

19 The programme provides a list of abbreviations. 1–2–3–4–5–6–7 NA DU

If specific problems arose please elaborate. Also, if possible, please provide suggestions for improvement.

B. User Control of the Programme

Low High

20 Moving between menus is easy. 1–2–3–4–5–6–7 NA DU

21 Going back to previous screens is easy. 1–2–3–4–5–6–7 NA DU

22 Undoing operations is simple. 1–2–3–4–5–6–7 NA DU

23 The user can control what the programme does. 1–2–3–4–5–6–7 NA DU

24 Accessing help is easy. 1–2–3–4–5–6–7 NA DU

25 The help files are clear. 1–2–3–4–5–6–7 NA DU

26 Error messages are appropriate. 1–2–3–4–5–6–7 NA DU

27 Speed of the programme is appropriate. 1–2–3–4–5–6–7 NA DU

28 Data entry operations are displayed on the screen appropriately. 1–2–3–4–5–6–7 NA DU

29 The programme allows interruptions of a task to start or resume another task. 1–2–3–4–5–6–7 NA DU

30 The programme is still usable while it is doing other tasks e.g printing. 1–2–3–4–5–6–7 NA DU

31 The programme tells you what it is doing. 1–2–3–4–5–6–7 NA DU

32 Information about the current programme status is obtainable. 1–2–3–4–5–6–7 NA DU

33 Commands are easy to understand. 1–2–3–4–5–6–7 NA DU

Low High

If specific problems arose please elaborate. Also, if possible, please provide suggestions for improvement.

C. Ease of Learning of the Programme

Low High

34 Getting started is easy. 1–2–3–4–5–6–7 NA DU

35 Learning the operations is easy. 1–2–3–4–5–6–7 NA DU

36 The programme makes few assumptions about your prior knowledge of the programme. 1–2–3–4–5–6–7 NA DU

37 Commands are easy to remember. 1–2–3–4–5–6–7 NA DU

38 Human memory limitations are respected. 1–2–3–4–5–6–7 NA DU

39 Information to complete the tasks is available. 1–2–3–4–5–6–7 NA DU

40 Information patterns are clear. 1–2–3–4–5–6–7 NA DU

41 Reference materials are clear. 1–2–3–4–5–6–7 NA DU

42 The programme is easy to learn how to use. 1–2–3–4–5–6–7 NA DU

43 User manuals are not required. 1–2–3–4–5–6–7 NA DU

44 Tutorials are clear. 1–2–3–4–5–6–7 NA DU

45 You can become acquainted with programme use with-out human assistance. 1–2–3–4–5–6–7 NA DU

46 Exploring features is encouraged. 1–2–3–4–5–6–7 NA DU

47 Relearning after intermittent use would be easy. 1–2–3–4–5–6–7 NA DU

48 Use by different levels of experience is accommodated. 1–2–3–4–5–6–7 NA DU

49 Experts can add features/shortcuts easily. 1–2–3–4–5–6–7 NA DU

50 You can use the programme without special computing knowledge. 1–2–3–4–5–6–7 NA DU

51 The programme provides information about what it can and cannot do. 1–2–3–4–5–6–7 NA DU

52 The programme explains each command and sub command on request. 1–2–3–4–5–6–7 NA DU

53 Learning new features would be easy 1–2–3–4–5–6–7 NA DU

If specific problems arose please elaborate. Also, if possible, please provide suggestions for improvement.

D. Completeness of the Programme

Low High

54 Terminology closely relates to the task area. 1–2–3–4–5–6–7 NA DU

55 The next screen in a sequence is predictable. 1–2–3–4–5–6–7 NA DU

56 Operations are closely related to tasks. 1–2–3–4–5–6–7 NA DU

57 Computer related terms are used appropriately. 1–2–3–4–5–6–7 NA DU

58 Terms on the screen are precise. 1–2–3–4–5–6–7 NA DU

59 Abbreviations are comprehensible. 1–2–3–4–5–6–7 NA DU

60 The number of operations per task is appropriate. 1–2–3–4–5–6–7 NA DU

61 Work proceeds from top to bottom. 1–2–3–4–5–6–7 NA DU

62 The programme maintains a sense of position. 1–2–3–4–5–6–7 NA DU

63 The programme appears as a whole package. 1–2–3–4–5–6–7 NA DU

64 The programme makes repetitive or routine input unnecessary. 1–2–3–4–5–6–7 NA DU

If specific problems arose please elaborate. Also if possible please provide suggestions for improvement.

E. Correspondence with User Expectations

Low High

65 Display rate for screen displays are appropriate. 1–2–3–4–5–6–7 NA DU

66 Response time for operations is appropriate. 1–2–3–4–5–6–7 NA DU

67 The programme provides the same response times for equal activities. 1–2–3–4–5–6–7 NA DU

68 The link between operations and results is clear. 1–2–3–4–5–6–7 NA DU

69 The programme acknowledges successful completion of a task. 1–2–3–4–5–6–7 NA DU

70 Terminology is consistent. 1–2–3–4–5–6–7 NA DU

71 Task related terms are used consistently. 1–2–3–4–5–6–7 NA DU

72 Computer related terms are used consistently. 1–2–3–4–5–6–7 NA DU

73 Informative feedback is appropriate. 1–2–3–4–5–6–7 NA DU

74 The amount of feedback is appropriate. 1–2–3–4–5–6–7 NA DU

75 Amount of feedback is controlled by you. 1–2–3–4–5–6–7 NA DU

76 The programme behaves similarly in similar situations. 1–2–3–4–5–6–7 NA DU

77 The programme requests similar user actions to similar tasks. 1–2–3–4–5–6–7 NA DU

78 The programme does what you expect it to. 1–2–3–4–5–6–7 NA DU

79 The programme allows you to see what is happening at any time. 1–2–3–4–5–6–7 NA DU
Low High

If specific problems arose please elaborate. Also, if possible, please provide suggestions for improvement.

F. Flexibility in Task Handling

Low High

80 The programme allows for alternative entry devices e.g. mouse, light pen. 1–2–3–4–5–6–7 NA DU

81 Fixed function keys are used for common tasks. 1–2–3–4–5–6–7 NA DU

82 The cursor starts at the first entry point. 1–2–3–4–5–6–7 NA DU

83 The programme allows you to do the task in different ways. 1–2–3–4–5–6–7 NA DU

84 Excess cursor movement is minimal. 1–2–3–4–5–6–7 NA DU

85 The programme allows for the use of the numeric keypad for massed entry of numbers. 1–2–3–4–5–6–7 NA DU

86 The programme provides reduced input/ output according to your training level. 1–2–3–4–5–6–7 NA DU

If specific problems arose please elaborate. Also, if possible, please provide suggestions for improvement.

G. Fault Tolerance

Low High

87 The programme is tolerant to user errors. 1–2–3–4–5–6–7 NA DU

88 The programme tolerates typical typing errors. 1–2–3–4–5–6–7 NA DU

Low High

89 The programme immediately detects errors. 1–2–3–4–5–6–7 NA DU

90 Destructive operations are protected. 1–2–3–4–5–6–7 NA DU

91 User errors do not abort a session. 1–2–3–4–5–6–7 NA DU

92 User errors do not destroy data. 1–2–3–4–5–6–7 NA DU

93 Error messages clarify the problem. 1–2–3–4–5–6–7 NA DU

94 Requests for help do not cause data loss. 1–2–3–4–5–6–7 NA DU

95 Context sensitive help is provided. 1–2–3–4–5–6–7 NA DU

96 Error messages are displayed on the entry screen. 1–2–3–4–5–6–7 NA DU

97 Error correction is at the point where the error occurred. 1–2–3–4–5–6–7 NA DU

98 The programme highlights where the errors are. 1–2–3–4–5–6–7 NA DU

99 Error messages are removed after correction. 1–2–3–4–5–6–7 NA DU

100 Auditory signals are used appropriately. 1–2–3–4–5–6–7 NA DU

101 The programme allows partial retyping if previous input was erroneous. 1–2–3–4–5–6–7 NA DU

102 Error messages contain correction hints. 1–2–3–4–5–6–7 NA DU

103 The programme provides messages with different levels of detail dependent on you experience. 1–2–3–4–5–6–7 NA DU
Low High

If specific problems arose please elaborate. If possible, please provide suggestions for improvement.

H. Formatting

Low High

104 Characters in the displays are readable. 1–2–3–4–5–6–7 NA DU

105 Space surrounding characters is adequate. 1–2–3–4–5–6–7 NA DU

106 Highlighting facilitates performing the task. 1–2–3–4–5–6–7 NA DU

107 Levels of intensity and boldfacing are apparent. 1–2–3–4–5–6–7 NA DU

108 Letter or shape size changes is apparent. 1–2–3–4–5–6–7 NA DU

109 Underlining is appropriate. 1–2–3–4–5–6–7 NA DU

110 Reverse video is used appropriately. 1–2–3–4–5–6–7 NA DU

111 Blinking is used appropriately. 1–2–3–4–5–6–7 NA DU

112 Colour changes are used appropriately. 1–2–3–4–5–6–7 NA DU

113 Display lay outs simplify the task. 1–2–3–4–5–6–7 NA DU

114 Screens are pleasing to look at. 1–2–3–4–5–6–7 NA DU

115 Displays are uncluttered. 1–2–3–4–5–6–7 NA DU

116 Displays are orderly. 1–2–3–4–5–6–7 NA DU

117 A title always identifies the display. 1–2–3–4–5–6–7 NA DU
Low High

If specific problems arose please elaborate. Also, if possible, please provide suggestions for improvement.

If you have any further comments, or suggestions that may improve the programme please elaborate.

Appendix 15. Usability Rating Sheet Used in the Study.

Software Evaluation Study

Please answer the following questions. All information is confidential.

1. Please print you first name: ______________________________

2. Please print your second name: ____________________________

3. Please print the number you were assigned: _________________

4. Please rate the programme on a 1 to 10 scale as to how usable you felt the programme was. 1 = low, 10 = high.

1----2-----3----4----5-----6-----7-----8-----9-----10
Usability of the programme

Thank you for your participation. If you would like more information about the study, and the results, please tick the box below and supply a contact address.

Yes I would like more information about the study. [ ]

My contact address is ______________________________________

____

____

Appendix 16. Means, Standard Deviations and Sample Sizes of the Questionnaire Ratings.

Questionnaire results: Sample size, means, and Standard deviations for the Word Processor, Database and Spreadsheet.

Factor one: Programme Self-Descriptiveness.

	Word Processor			Database			Spreadsheet		
Q1	14	4.79	1.25	12	4.83	1.27	13	5.77	1.69
Q2	14	4.79	1.25	12	4.83	1.27	13	5.77	1.69
Q3	12	4.25	1.48	12	4.50	1.45	12	3.75	1.48
Q4	13	3.31	1.84	11	3.64	2.06	12	3.67	1.78
Q5	8	4.00	1.31	12	4.75	1.60	12	4.75	1.60
Q6	10	4.70	1.95	12	4.92	1.93	11	6.18	0.75
Q7	14	3.79	1.48	12	4.67	1.37	13	5.15	1.28
Q8	14	2.79	1.31	11	3.18	1.78	13	3.15	1.68
Q9	12	2.67	1.50	10	3.30	1.77	9	3.67	1.80
Q10	14	3.14	1.66	10	3.10	1.52	11	2.91	1.81
Q11	9	1.89	0.93	6	3.50	1.87	7	2.43	1.62
Q12	14	4.43	1.95	11	5.00	1.48	12	2.92	1.68
Q13	14	3.07	1.73	12	4.75	1.91	13	4.31	1.55
Q14	13	4.46	1.45	11	5.36	1.50	13	5.15	2.03
Q15	11	4.45	1.86	9	5.00	1.73	10	3.70	2.16
Q16	7	3.00	1.41	10	4.70	1.34	9	3.56	2.01
Q17	7	2.86	1.21	6	4.00	1.79	7	2.86	1.77
Q18	12	3.00	1.60	10	3.40	2.22	12	2.92	1.83
Q19	3	2.00	0.00	8	5.38	1.92	7	2.86	2.12

Factor two: User Control of the Programme.

Q20	7	2.29	1.50	8	4.38	2.00	8	2.52	1.58
Q21	14	5.93	1.14	12	5.42	1.31	12	6.17	0.83
Q22	13	4.69	1.97	10	5.40	1.26	7	5.86	1.86
Q23	11	3.82	1.83	12	4.92	1.44	12	4.92	1.51
Q24	13	4.15	1.41	11	5.00	1.41	10	4.20	2.10
Q25	12	4.75	1.36	7	3.00	1.91	9	2.67	1.41
Q26	13	3.15	1.63	7	2.43	0.79	5	1.30	2.20
Q27	9	3.78	1.79	7	4.57	1.51	6	4.50	1.52
Q28	10	5.50	1.08	10	5.40	1.71	12	5.83	1.11
Q29	9	3.89	1.83	11	5.09	1.45	10	5.00	1.15
Q30	6	4.67	2.58	7	4.71	1.50	5	4.40	1.52
Q31	7	4.29	1.80	6	4.50	1.64	9	3.67	2.06
Q32	13	3.31	1.32	11	2.73	1.49	12	4.25	2.05
Q33	7	4.14	1.68	6	4.50	1.64	7	3.86	1.46

Factor three: Ease of Learning the Programme.

	Word Processor			Database			Spreadsheet		
Q34	14	4.21	1.25	12	5.08	1.56	13	5.77	0.73
Q35	14	5.00	1.57	12	5.75	1.22	12	5.92	1.56
Q36	14	4.07	1.77	12	5.00	0.95	13	5.85	1.28
Q37	14	5.00	1.75	11	5.00	1.55	12	4.25	1.91
Q38	14	3.79	1.37	12	5.42	1.00	13	6.08	1.19
Q39	12	3.17	1.34	11	4.82	1.25	8	4.88	1.81
Q40	9	2.89	1.45	10	4.50	2.01	11	3.73	2.00
Q41	6	4.00	1.26	10	4.60	1.43	5	4.80	2.28
Q42	5	3.80	1.48	7	4.71	1.80	4	4.50	2.52
Q43	14	4.21	1.48	12	5.50	1.24	13	6.08	1.44
Q44	10	3.70	1.95	12	4.33	1.72	10	5.00	2.11
Q45	3	6.00	1.00	5	4.40	1.67	6	5.00	2.10
Q46	14	3.21	1.48	11	4.73	2.10	11	5.36	1.03
Q47	9	2.44	1.13	8	4.63	2.07	8	3.50	1.31
Q48	13	4.69	1.75	12	5.92	1.16	10	6.30	0.82
Q49	11	3.91	1.04	11	4.27	1.42	9	4.22	1.79
Q50	6	4.17	1.47	7	4.71	1.70	6	4.50	1.97
Q51	14	3.64	1.82	12	5.33	1.37	13	5.46	1.05
Q52	11	1.91	0.83	8	2.38	1.51	9	2.67	1.58
Q53	12	2.42	1.00	7	3.29	2.50	9	2.78	1.39

Factor four: Completeness of the Programme.

Q54	14	4.00	1.47	12	4.83	1.53	11	5.64	0.92
Q55	8	4.25	1.39	8	5.00	1.31	10	5.40	1.26
Q56	7	4.14	2.12	10	4.50	1.72	3	6.00	1.73
Q57	9	3.89	1.45	10	5.10	1.45	9	5.00	1.12
Q58	10	6.10	0.99	12	5.75	1.22	8	4.63	1.92
Q59	11	4.36	1.50	12	4.92	1.31	13	5.08	1.44
Q60	9	4.44	1.81	10	5.10	1.60	10	5.50	1.08
Q61	9	5.11	1.05	11	5.55	1.04	8	5.00	1.31
Q62	10	6.10	0.99	12	5.75	1.22	8	4.63	1.92
Q63	8	4.50	0.93	9	5.44	1.13	5	5.60	1.14
Q64	11	4.82	1.40	8	4.63	2.13	6	4.50	1.38

Factor five: Correspondence with user expectations.

Q65	9	4.33	1.94	8	3.87	1.81	9	4.33	1.66
Q66	8	5.37	0.92	10	5.20	0.92	9	5.33	1.00
Q67	13	5.62	0.87	12	5.17	1.27	11	5.73	0.90
Q68	9	4.78	0.83	10	5.00	1.33	10	5.70	0.95
Q69	11	4.00	1.73	8	5.13	1.89	12	5.92	1.08
Q70	10	2.20	1.23	9	3.33	2.29	12	3.42	1.83
Q71	12	4.92	1.24	12	5.17	1.03	12	6.00	0.95
Q72	10	5.60	1.17	10	5.10	1.20	12	5.50	1.00
Q73	8	4.63	1.51	11	5.00	1.34	11	5.45	1.37
Q74	12	3.08	1.38	9	4.11	1.62	9	4.33	1.41
Q75	12	2.83	1.47	9	4.22	1.64	9	2.56	1.51
Q76	10	5.10	1.66	10	5.30	1.25	10	5.50	0.97
Q77	9	5.22	0.83	9	5.33	1.32	11	5.82	1.08
Q78	13	4.38	1.89	9	5.00	1.50	12	5.75	1.54
Q79	13	3.69	1.80	11	4.18	1.47	12	4.75	1.91

Factor six: Flexibility in task handling.

	Word Processor			Database			Spreadsheet		
Q80	4	1.00	0.00	1	5.00	0.00	1	6.00	0.00
Q81	12	5.25	1.71	9	5.78	0.97	8	5.00	2.27
Q82	12	6.50	0.52	11	6.27	1.01	9	5.67	1.99
Q83	7	4.00	1.63	7	4.29	1.70	9	4.33	1.73
Q84	10	4.60	1.90	11	5.55	1.51	9	5.11	1.27
Q85	1	5.00	0.00	5	6.00	0.71	7	3.14	2.12
Q86	7	2.14	1.21	4	5.00	1.41	4	2.00	1.41

Factor seven: Fault Tolerance.

Q87	13	3.54	2.11	10	3.50	1.84	10	4.20	1.48
Q88	11	5.82	1.54	10	5.40	1.35	11	4.36	2.16
Q89	11	2.64	1.50	10	3.40	2.07	11	4.27	1.85
Q90	5	5.20	1.92	5	2.80	1.10	6	4.00	1.79
Q91	9	6.00	0.87	9	4.33	2.60	9	5.56	1.33
Q92	10	4.80	2.04	10	3.50	2.22	9	3.67	2.29
Q93	9	3.33	1.94	10	3.50	2.32	9	3.33	2.18
Q94	10	6.10	0.74	3	5.00	1.73	2	6.00	1.41
Q95	4	3.00	2.00	4	2.50	1.91	4	1.25	0.50
Q96	11	4.73	2.45	9	5.11	1.69	11	4.55	2.16
Q97	7	4.14	1.86	8	5.75	1.75	9	5.22	1.79
Q98	6	2.33	1.51	8	3.25	2.38	7	4.57	2.23
Q99	7	5.57	1.72	9	5.67	1.41	5	5.60	2.07
Q100	2	1.50	0.71	3	4.67	2.52	6	5.50	0.55
Q101	9	5.89	1.17	10	5.40	1.43	8	4.63	2.07
Q102	8	4.13	1.46	6	1.83	1.17	6	2.50	1.76
Q103	7	1.71	0.49	6	1.67	1.21	5	2.20	1.10

Factor eight: Formatting.

Q104	13	6.15	0.90	12	6.33	0.89	13	6.31	0.95
Q105	13	6.15	0.99	12	6.33	0.89	13	6.00	1.68
Q106	11	5.45	1.86	10	5.80	1.55	13	5.92	1.32
Q107	9	5.11	1.96	12	5.67	1.61	9	5.11	1.96
Q108	8	5.00	1.85	10	5.60	2.01	8	4.50	2.14
Q109	4	6.00	1.15	10	5.40	1.58	3	5.67	1.15
Q110	2	6.00	1.41	4	5.50	1.29	0	—	—
Q111	3	6.33	1.15	6	5.67	1.37	3	4.67	0.58
Q112	5	5.60	1.34	10	6.00	0.94	8	5.88	0.83
Q113	9	4.89	1.27	12	5.67	1.67	12	5.83	0.94
Q114	13	4.92	1.80	12	5.58	1.88	13	5.00	1.87
Q115	13	5.31	1.32	12	5.50	1.38	13	5.31	1.70
Q116	13	5.69	0.95	12	5.75	1.29	13	5.92	0.95
Q117	12	5.75	1.14	11	5.18	1.89	12	4.42	1.56