Telematics Applications Project TE 2010

RESPECT
European
Usability
Support
Centres

Requirements Engineering And Specification In Telematics

WP3
Deliverable D3.2

Methods for User-Orientated Requirements Specification

J.Kirakowski, HFRG, Ireland (Ed.)

Version 1.6

17 April, 1997

Summary

This document presents methods to be used in user-based requirements engineering. It is intended primarily for the benefit of the Human-Computer Interaction specialist in order to supplement much existing material on methods and approaches for user-based system evaluation. The methods are organised into a framework developed in the 'Frameworks and Development Strategy' work package, and the process of how these methods may be used in a development environment is described in a companion document from this workpackage.

All methods presented have been used in industrial situations and have been seen to integrate with software development practice.

The methods are also classified according to their suitability for development projects involving users with special needs (young, aged, and physically handicapped).

Keywords: user-centred design, requirement specification, methods, special needs

Table of Contents

[redacted]

Document Versions and Acknowledgements

1.1 Based on D3.1.

1.2 Revised with reference to D5.1 draft.

1.3 Sent for review by project peer reviewer.

1.4 Extra sections added following reviewer's comments, tables tidied.

1.5 Comparison with software engineering methods reduced following decision to create new work task specifically for this.

1.6 Public version.

Thanks to Mr K Hurley and Ms N Vereker from HFRG who worked with me on earlier editions of this document. Many members of the RESPECT project have made valuable contributions to the methods reviewed and to the way they are described, for which we at the HFRG are extremely grateful. A number of participants in the UTEST discussion forum made extremely helpful comments on plans and drafts of earlier versions, contributed documents, and gave very much appreciated moral support and encouragement.

Summary and Context of Use

This document is the second edition of a collection of methods and approaches (together loosely called *practices*) for user-based requirements analysis produced by the RESPECT project. *The objective is to produce a comprehensive list of user-based requirements specification practices which have been adopted by industry and which are therefore likely to be of practical benefit to the Human Computer Interaction (HCI) specialist in this area.*

Part of the task of the HCI specialist is to manage the end-user input to the design process. We assume that in this task the HCI specialist is a **manager** and a **facilitator**, not a **representative** or **spokesperson** for the users or, worse still, an **interface between opposing world-views**. Essential to his or her expertise is a knowledge of a wide range of methods and approaches that can be pressed into service in different contexts as required.

We assume furthermore that the HCI specialist is going to be working in an inter-disciplinary team consisting of software designers, graphics artists, interface design specialists, marketing personnel and so on, and that he or she may well have *multiple roles* within this design team.

Three classes of readers are targeted by the current document:

1. The RESPECT project partners, in order to continue the discussion leading to a definition of the essential user-based requirements engineering methods necessary to support the process developed in Work Package 5;

2. Personnel charged with user-based requirements engineering in Telematics Applications Programme projects, to supplement extant collections of methods (which are mainly focused on user-orientated evaluation later in a project lifecycle) and to give an indication of the kinds of methods in which the European Usability Support Centres network can provide advice, training and consultancy;

3. HCI specialists in the Telematics and software industry in general, to present a state of the art collection of best practices in user-based requirements engineering and to stimulate the development of better methods.

The document is divided into two unequal parts and an appendix.

The first fairly brief part defines the concept of user-based requirements engineering and compares the software engineering view of user involvement in the requirements specification process with the human centred view of how users should be involved. A terminological distinction is made between *methods*, which are recipes for conducting successful requirements activities with users; and *approaches* or techniques in need of specific instantiations in order to be of practical use. This is of course more of a continuum than a dichotomy.

In the second part, a list of methods and approaches to facilitate user-based requirements engineering is given, organised under headings corresponding to the three main stages of the user centred model:

Stage 1 User Context Analysis

Stage 2 Feasibility and Prototyping

Stage 3 User Requirements Synthesis

These three stages are more of a spiral process than a waterfall one. The model which co-ordinates progress through these stages is not the concern of the present document: this is worked out and made explicit in the 'RESPECT User Requirements Framework Handbook' emerging from Work Package 5 of the RESPECT project, with which the current document will be ultimately merged.

In the appendix to this document is a classification of the user-based requirements methods and approaches by applicability to three groups of users with special needs: the aged, the young, and the physically handicapped. This has been developed as a result of the review of an earlier draft of the current document by the special needs task force in RESPECT.

At an early stage in the gestation of this document we thought this would be the right place to give an account of how requirements engineering from an HCI perspective differs from the activity as seen from a software engineering perspective. Several editions later, we realise that this is a problem of a different kind to that of cataloguing practices, and we have identified a new task within RESPECT and will produce a special report which will be concerned only with this topic later in 1997.

More information about the European Usability Service Centres network can be obtained from URLs mentioned on the RESPECT project web site at:

http://info.lut.ac.uk/research/husat/respect

or by contacting the project manager, Dr Nigel Bevan, at

n.bevan@npl.co.uk

Introduction

This document is a collection of approaches and methods (together loosely called *practices*) for **user-based requirements engineering** produced by the RESPECT project. We intend this term to mean: *eliciting from users what they want, developing with assistance from users a representation of what the system under design should look like, and verifying or confirming with users that this is what will satisfy their needs.*

A recent review of software engineering orientated requirements engineering practices by Loucopoulos and Karakostas [6] stresses that requirements engineering still lacks consensus as to what exactly constitutes software requirements engineering as a whole but presents a model in terms of three stages: **elicitation, representation,** and **validation.** These terms are likely to be more familiar to the software engineer but it will be seen that they map onto the terms used in the definition of user-based requirements engineering given above. At present there is no commonly accepted interdisciplinary framework which takes account of the wide range of criteria and practices relevant to studies of both software engineering and human-computer interaction [9].

We will argue elsewhere that some of the problems with requirements engineering raised by Loucopoulos and Karakostas can be resolved with reference to a greater focus on the needs of the end user. However, the methods described in section 2 do *not* deal with important issues such as data modelling [6], and the specification and validation of requirements such as Reliability, and Efficiency [3]. For these, a thorough grounding in software engineering principles and methods has always been, and continues to be essential.

Users within the software engineering model

A standard text on information systems design, current throughout the 1980s, writes:

> The bottom line to systems work, and in particular the gathering of study facts and their analysis, is to extract from a group of people their knowledge, ideas, and needs [2].

The text advocates a number of ways in which facts may be gathered; but as far as people are concerned, it leans on **interviews**, **surveys**, and **naturalistic observation** (the emphasis on ‘desk research’, ie looking at documents generated by the commissioning company is typical of the period). The extremely concise text by Olle and colleagues, first published at the end of the 1980s and current through the early 1990s [8], views design methodologies as processes that take the design team at least part-way through an intricate information systems methodology framework of which the **system design** is a part-way product. Olle et al write that “The system design stage consists of preparing a prescriptive statement about an information system...”

and that furthermore

> The user acceptor must, in order to fulfil his or her role, be able to judge whether the information system to be built will meet his or her needs.

The role of the user into the lifecycle of such a document is classically viewed as (1) to provide input and (2) to sign off the end result. But as Olle et al write: “the lack of comment (to this process) from users has all too frequently been accepted as approval."

Comment from end users may not achieve its purpose for a number of reasons; chief among them being that the non-technical end user may not understand the notations and language used in a software engineering requirements specification, and be unable to visualise or predict the end result of a development process based on such a specification.

This state of affairs may not matter too much if systems are designed for use in batch mode, where the end users' interaction with the system is limited or at least well-buffered by specialists (it is useful to remember that it was WW Royce who first proposed the so-called waterfall life cycle model in the batch-computing days of the early 1970s), but as interactive computing became increasingly ubiquitous, and as the level of involvement of the users with the system became increasingly intensive, this mind set became increasingly pernicious. Two issues raise themselves with particular force:

1. How can the requirement specification *process* be re-designed so that end users can be used as a resource throughout the process?

2. What *tools and techniques* are available to facilitate end users' involvement in the process?

For in addition to the increased amount of interactivity in contemporary information systems, we have to face the fact that such systems are also going to embody, to increasingly greater extents, options for end user originated customisation and fine tuning and may have to cope with data processing demands that cannot be foreseen by the design team or even the user representatives themselves.

Thus requirements engineering is no longer a case of analysing the 'tasks' the users will have to 'complete' with the software, but of exploring the latent possibilities of the medium and approach being adopted by the development project, and ensuring that these possibilities will be easily available to the user.

The iterative RESPECT model

Rubin [7] emphasises that it is not enough simply to *focus on users*. He writes:

> In the worst cases, I have seen entire engineering teams trek across several continents on a tour of customer sites, only to return with a myriad of conflicting information documented and filed away on unstructured trip reports. Unfortunately, this volume of information and huge effort served little useful purpose.

The key solution is to collect perhaps less information to start with, but to verify it more frequently. Although the word 'prototyping' has by now a specialised technical meaning as in 'rapid prototyping', in fact any representation of the system is technically speaking a prototype. This can be anything from a sketch on a back of an A4 pad to a miniature working model. Distinctions are usefully made between 'horizontal' and 'vertical' prototypes [5], where a 'horizontal' prototype is an overview of a large part of the system, and a 'vertical' one is a more detailed view of a smaller self-contained segment.

The essence of true iterative design which is open to feedback from end users is that the team is able, following Rubin, to "shape the product" through a process of design, test, redesign and retest.

Isensee and Rudd [5] summarise studies that claim that 60% to 80% of all system problems can be traced back to inaccurate requirements definitions, so it is not surprising to read that in 1992, 87% of developers from a large US sample reported using some kind of iterative design strategy. So obviously prototyping makes sense, commercially. These authors also review an up-to-date collection of software based tools which may be used in the stages of prototyping where computer representations have become necessary, and the reader is referred to this invaluable resource.

The RESPECT model of user-based requirements engineering is detailed in [7], and is presented in outline here. It is an iterative process with three main stages, described as follows. Each stage may be returned to, but in practice we would expect to see considerable iteration *within* stages, possibly using a time-boxing principle.

Stage 1 User Context Analysis

This consists of understanding the initial project requirement and performing an analysis of the main user groups, their tasks and working environments. Design constraints and relevant standards to be applied are identified.

Stage 2 Feasibility and Prototyping

This stage is concerned with concept development and representing possible concepts by means of scenarios, paper prototypes, software prototypes etc. This allows users to assimilate each concept easily and to comment on its feasibility. Different concepts are evaluated by comparing their characteristics, by assessing their cost/benefits for different user groups, and by testing them with users. The requirements specification is validated against user experience as it is developed.

Stage 3 User Requirements Synthesis

This stage integrates the user requirements identified in stages 1 and 2, and groups them ready for input into the design phase of system development. The different requirements groups include: concept description, general system characteristics, system functions (based on the tasks that must be performed), user-system interfaces, user support needs, physical and organisational characteristics, usability goals, and the approach for installing the system. A plan for implementing the user requirements is also developed.

For the purpose of the current document, the methods collected are summarised under each of these stages at the start of Part 2.

Added value to software engineering approaches

The following user-centred methods are current in software engineering treatments of user-based requirements work:

- interviews
- surveys
- naturalistic observation
- brain storming
- focus groups
- ethnographic approaches

These methods are all relevant to the early *user context analysis* phase of user-based requirements work. However, they do not support in any strong way the *feasibility and prototyping phase*, nor do they support the important *user requirements synthesis* phase outlined in the current framework.

A view of how different aspects of requirements (business and market; user; and functional) can be successfully integrated as a spiral process in a realistic commercial environment is given in [4], which describes how users, usability engineers, system engineers, and developers are able to rapidly co-ordinate their efforts to produce a GUI design using an iterative methodology known as *the Bridge*. Of special interest here is how the Bridge methodology efficiently addresses the integration between the software product's deep functionality in addition to the GUI 'look and feel'.

Approaches such as these, as well as STUDIO [1] and GUIDE [10] will become increasingly prevalent as the software industry takes up the opportunities afforded by involving the end users in requirements engineering in order to create quality software products.

Terminology

There is some confusion in terminology in this area, and the purpose of this section is to present a systematic terminology that the RESPECT project have found useful when dealing with these issues.

Models are generic descriptions of strategies in which development work may be carried out. It has become commonplace to talk of the 'waterfall' model of software development, while understanding that this model has many instantiations in different organisations—some instantiations are 'official' and some are 'de facto'. The document by Maguire and Kirakowski emerging from Work Package 5 of the RESPECT project [7] presents a *model* of the user-based requirements engineering framework.

Approaches are generic in nature: they are simply broad classes of ways of doing things which may be instantiated in dozens of different ways. Thus for instance **Wizard of Oz prototyping**, or **interviews**. Approaches are ideas, sometimes quite extensively commented upon and discussed, but these ideas have to be developed and instantiated in order to be of any practical value. Very often, an approach only has practical value when it is instantiated in a particular company methodology.

Methods are ways of doing things, like recipes. Methods should be prescriptive, otherwise they are approaches, as discussed above. Examples of methods are **Video Prototyping**, or **Usability Context Analysis**. Methods have the characteristic that they can be employed in a variety of organisational circumstances, and they may have some necessary materials associated with them that can either be copied or purchased from the method developer, or developed anew. However, anybody should be able to apply a well-described method once it has been explained to them.

The reader is referred to [5] for a recent collection of software-based *tools* that may be pressed into use for prototyping.

References

[1] Browne, D (1994) *STUDIO: STructured User-interface Design for Interaction Optimisation.* Prentice Hall, NY.

[2] Burch, JG, FR Strater, and G Grudnitski (1983) *Information Systems: Theory and Practice (3rd Ed).* J Wiley & Sons.

[3] Davis, AL (1993) *Software Requirements: Objects, Functions and States*. Prentice Hall International.

[4] Dayton, T, A McFarland, and J Kramer (in press, 1997) Bridging User Needs to OO GUI Prototype via Task Object Design. To appear in: L Wood and R Zeno (Eds) *User Interface Design: Bridging the Gap from Requirements to Design*. CRC Press, Boca Raton, Fl.

[5] Isensee, S and J Rudd (1996) *The Art of Rapid Prototyping.* International Thomson Computer Press.

[6] Loucopoulos, P and V Karakostas (1995) *System Requirements Engineering.* McGraw Hill International.

[7] Maguire, M and J Kirakowski (1997) RESPECT User Requirements Framework Handbook (RESPECT Project deliverable D5.1).

[8] Olle, TW, J Hagelstein, IG Macdonald, C Roland, HG Sol, FJM van Assche, AA Verrijn-Stuart (1991) *Information Systems Methodologies.* IFIP - Addison-Wesley.

[9] Preece, J and HD Rombach (1994) A taxonomy for combining software engineering and human-computer interaction measurement approaches: towards a common framework. Int J Hum Comput Stud, 41.2, 553-583.

[10] Redmond-Pyle, D and A Moore (1995) *Graphical User Interface Design and Evaluation: A Practical Process*. Prentice Hall, Hemel Hempstead.

Methods and Approaches

Summary of methods and approaches by Stage

Stage:	User Context Analysis	Feasibility and Prototyping	User Requirements Synthesis
Group Discussion	x	x	x
Usability Context Analysis	x		
Brainstorming	x		
Naturalistic Observation	x		
Surveys	x		
Ethnographic Approach	x	x	
Focus Groups	x	x	
Interviews	x	x	
Functionality Matrix	x	x	
Task Allocation Charts	x	x	
Task Analysis	x	x	
Wizard of Oz		x	
Paper Prototyping		x	
Scenario Building		x	
Storyboarding		x	
Empathic Modelling		x	x
Video Prototyping		x	x
Diary Methods		x	x
Rapid Prototyping & RAD/JAD		x	x
Lab-based Observation			x
Co-operative Evaluation			x
Parallel Design			x
Walkthrough			x

Group Discussion/ Future Workshops

Primary Reference Source

Kensing, F and Madsen, KH (1991) Generating visions: future workshops and metaphorical design. In J Greenbaum & M King (Eds) *Design at work -- Co-operative Design of Computer Systems.* Lawrence Earlbaum, Hillsdale, NJ.

Summary description

Group discussions are based on the idea of stakeholders within the design process discussing new ideas, design options, costs and benefits, screen layouts etc., when relevant to the design process. Group discussions help to summarise the ideas and information held by individual members. The general idea is that each participant can act to stimulate ideas in the other people present, and that by a process of discussion, the collective view becomes established which is greater than the individual parts. The 'future workshops' concept is designed specifically to allow actors who are not used to having a voice in the discussion process to step forward.

A future workshop goes through three phases:

- Critique: the participants voice current problems and structure them so that a constructive outcome may be expected;
- Fantasy: the participants generate visions of the ideal future scenario;
- Implementation: the 'visions' are evaluated and a plan for future action is set up.

Typical Application Areas

Useful for obtaining opinions from a range of people who may not feel secure about voicing their opinions.

Benefits

Group discussions help to summarise the ideas and information held by individual members. The general idea is that each participant can act to stimulate ideas in the other people present, and that by a process of discussion, the collective view becomes established which is greater than the individual parts.

Limitations

Some individuals may not get the chance to air their views or may be inhibited by other group members, particularly colleagues or more senior staff. Some people may also not always think creatively in a group setting and prefer to be interviewed or to complete a survey form in their own time.

Resource requirements

Requires preparation on the part of the facilitator to make sure that the meeting focuses on the issues at hand. The facilitator should ideally be outside the process being discussed, but familiar with the issues being raised. A 'safe' situation should be set up so that participants may not be inhibited in voicing opinions which may contradict company policy.

Procedure

1. Decide on the objectives of the meeting and the participants required to take part in it.

2. When contacting the participants explain clearly what topics are to be discussed and the meeting format. Discuss the issue of confidentiality of the proceedings. Obtain agreement beforehand if any particular recording techniques are to be used e.g. video or audio recording.

3. Produce a timetable for the session and run a pilot session to check that the timetable is realistic. If background information is required from the group individuals, prepare a suitable questionnaire for administration either before or after the session.

4. During the session the facilitator should be active in formulating the themes for the discussion, and summing up the results at the end of each topic.

Practical guidance

¥ Create an atmosphere which is person-centred and non-evaluative.

¥ Provide participants with a simple form to complete personal details before the meeting starts. This can help provide an activity while any last minute setting up is required or if some participants are late arriving.

¥ Suggest some rules for the discussion and enforce these rules.

¥ Support the participants in the formulation of the problem, and guide the participants when necessary.

¥ Quash destructive behaviour as soon as you can identify it.

¥ Protect individuals whose ideas and comments differ from others in the group.

¥ Do not suggest solutions to the problem.

¥ Avoid evaluating proposed solutions.

¥ Ensure that all participants get an opportunity to contribute and that the proceedings are not dominated by any one person or group.

¥ If the group includes people with severe visual impairments, the group leader should wear bright clothes to make sure that he can be seen by all the participants.

Usability Context Analysis

Primary Reference Source

Bevan, N (Ed), (1997) *Usability Context Analysis: a Practical Guide.* NPL Usability Services, Teddington, UK

Summary description

Usability Context Analysis (UCA) is a structured method for eliciting detailed information about a product and how it will be used, and for deriving a plan for a user based evaluation of a product. For this method stakeholders meet to detail the actual circumstances (or intended use) of a product. This is produced in a document called the Context Report Form, which is then examined by a usability consultant who decides who decides if each factor is indeed important for the usability of the product. Following this inspection a summary list, called the Context of Evaluation, of these factors is produced. This list specifies important characteristics of the products' users, their tasks, their environment, and also lays the foundation for an observational evaluation.

Typical Application Areas

Context Analysis meetings should take place as early as possible in the design of a product. However the results of these meetings can be used throughout the lifecycle of the product; being continually updated and used for referral. The method is also of general relevance to a wide range of systems and processes. The objective of the method is to collect information -- this should be done preferably through a well-prepared meeting, but it may also be done through as series of personal interviews (for instance, if schedules in a busy company do not permit essential personnel to attend a general meeting).

Benefits

Context analysis offers a number of benefits. It provides a framework to ensure that all factors which may affect the usability of a product are considered. It also helps to ensure that user-based evaluation produces valid results, by specifying how important factors are to be handled in an evaluation, and by defining how well the evaluation reflects real world use.

Limitations

The success of this method depends upon competent chairing to keep the meeting focused on the relevant issues. Familiarity with the method by the chairperson is essential.

Cost of use

No specific hardware or software is required for this method. A word processor is needed to generate a report. However the facilitator of a context meeting should be familiar with the system to be evaluated and so may require access to a functional prototype or a finished product.

Costs of Acquisition

For a person leading a Context Analysis, training or prior experience is highly recommended, and familiarity with the Context Report Form is essential.

Suitability for requirements engineering in Telematics:

There is considerable experience of this method within the RESPECT consortium. Usability context analysis particularly well integrated with performance measurement methods later in the project lifecycle, and user based questionnaires such as HFRG's SUMI and MUMMS questionnaires.

How to get it:

Contact NPL Usability Services at `nigel@hci.npl.co.uk.`

Detailed description of method:

There are a number of steps required to conduct the UCA method:

1. First identify all relevant individuals who will be involved in the process and organise the context meeting. Examine the context report form to identify in advance any factors which do not need to be discussed at length at the meeting.

2. Complete the Product Report Form to define the scope of the term "product" for the analysis.

3. At the meeting, fill in the Context of Use column of the Context Report.

4. After the meeting identify the critical components which may affect usability.

5. If Context Analysis is to be conducted as preparation for an evaluation, identify how the critical components will be handled during the evaluation.

6. Produce an evaluation plan detailing the users, the tasks they will carry out, and the circumstances of the evaluation.

When applying this method it is important that the relevant stakeholders are present.

Brainstorming

Primary Reference Sources

Osborne, A. F. (1963), *Applied Imagination*, Schribener and Sons, NY.

Summary Description

Brainstorming is one of several group approaches, probably the oldest and best known. The idea is to let people come together and inspire each other in the creative, idea generation phase of the problem solving process.

It is used to generate new ideas by freeing the mind to accept any idea that is suggested, thus allowing freedom for creativity. The approach has been widely used in design.

Brainstorming is used at an early stage in the elicitation process for the rapid generation of ideas and problems in a specific domain, and is focused on the quantity of the response. Clustering methods may be used to enhance the outcome of a group session.

Typical Application Areas

Early in the development phase when little of the actual design is known, and there is a need for new ideas.

Benefits

The group process as such is usually perceived as rewarding in itself, and it creates a feeling of ownership of the result. In the brainstorming process everybody in the group can take credit for the good ideas.

Also when any experts are present, it has the advantage of preventing early criticism of ideas and reduces negative evaluative comments at an early stage of system development.

Limitations

There has been a wide range of studies intended to evaluate the efficiency of the approach, and the majority of these studies shows that people working in isolation produce more and better ideas than when working as a group.

Costs of Use

The human resources are the most important for succeeding with this approach. The more creative people with a variety of experiences in the field, the better the result. 5 - 12 people may participate, and the session need not take more than one hour.

Costs of Acquisition

This is not a proprietary approach and may be self-taught from the primary reference source.

Suitability for requirements engineering in Telematics

Very relevant, however, new developments making use of creative drawing techniques should also be considered when designing graphical and multi-media interfaces. See for instance Verplank (1991).

References

Verplank, W. (1991), Graphical Invention for User Interfaces,Tutorial, ACM CHI'91 Conference on Human Factors in Computing Systems, 1991.

Naturalistic Observation

Primary Reference Source

Loucopoulos P and V Karakostas (1995) *System Requirements Engineering.* McGraw Hill International.

Summary description

Observational methods involve an investigator viewing users as they work and taking notes on the activity which takes place. Observation may be either direct, where the investigator is actually present during the task, or indirect, where the task is viewed by some other means such as through use of a video camera.

Typical Application Areas

Useful early in specification for obtaining qualitative data. This method is an alternative (non-involving) version of Contextual Inquiry. It is useful for studying currently executed tasks and processes. It has been extensively advocated in the past.

Benefits

Allows the observer to view what users actually do in context. Direct observation allows the investigator to focus attention on specific areas of interest. Indirect observation captures activity that would otherwise have gone unrecorded or unnoticed.

Limitations

Observing can be obtrusive and subjects may alter their behaviour due to the presence of an observer. Co-operation of users is vital and so the interpersonal skills of the observer are important. Notes and video tape need to be analysed by the note-taker which can be time consuming and prevents the task being split up for analysis by a number of people. If events or behaviours which occur at unpredictable intervals are of interest, this kind of observation can become extremely time-consuming.

Cost of use

Analysis usually takes 5 to 7 times the amount of time spent recording events unless a substantial amount of analysis is done in real-time, during off-peak moments.

Indirect observation requires access to audio visual recording and playback equipment.

Costs of Acquisition

Observers require training and practice in order to take accurate and complete notes.

Suitability for requirements engineering in Telematics:

Some partner experience (SINTEF, NPL, HFRG, NOMOS). Technique widely used in industry by HCI consultants, and portable video kits are popular.

How to get it

Widely documented in the literature.

Detailed description of method

Naturalistic Observation as a field method involves the following steps:

1. Establish objectives and information requirements. Should the coverage be in breadth or in depth? It is extremely important at this stage to find out what will happen to the end-product of this process, and therefore to tailor the whole process to the requirements of those who will receive the results.

2. Gain contacts and especially their co-operation with the process of Naturalistic Observation that you intend to carry out. Establish the times, places, and people who will be observed. Note that in some countries, the law may prohibit you from taking video films of people without their explicit written consent.

3. Decide on the recording technique you will use. Will you rely on hand-written notes (traditional), audio, or video and audio records? Note that the more complete your record, the longer it takes to analyse. It is useful to be able to make some kind of first-cut analysis during observation

4. Analyse, summarise, and report in relation to the objectives set out at the start.

Observation as an approach in a laboratory setting is instantiated quite specifically using the Laboratory Based Observation approach. A variation of single user observation is two-user observation where pairs of users are invited to work together and the above process is carried out on the pair. One of the 'users' in two-user observation may be a member of the design team, and this is particularly useful in situations where there may be an unstable prototype.

Surveys

Primary Reference Sources

Kirakowski, J and M Corbett (1990) *Effective Methodology for the study of HCI.* North Holland Elsevier.

Summary description

A survey involves administering standard questionnaires to a large sample population. Surveys can help determine customer preferences, work practice and attitudes. There are two types: Closed, where the respondent is asked to select from available responses and Open, where the respondent is free to answer as they wish.

Typical Application Areas

Open surveys are applicable in the same kinds of situations as semi-structured interviews, closed surveys as structured interviews. Interviews are preferable in situations where there is a lot of person-power available for requirements elicitation and the user group is easily accessible. Surveys should be used, by preference, in situations where the user group is spread out geographically or where person-effort is small. Unless the user group are especially motivated, response rates of 20% or less are common with questionnaires.

Benefits

Quick and relatively inexpensive to administer. Results can be subjected to statistical analysis

Limitations

Questionnaire design is not straightforward. It may be hard to follow up on interesting comments as it is often not desirable or possible to keep records of who wrote what comment.

Cost of use

Depends very much on the complexity of the survey and the number of respondents needed. Mailing costs are likely to be high, as are telephone costs if a follow-up is to be used.

Costs of Acquisition

Survey design requires some expertise and can be costly if inexperienced staff attempt to use it for the first time.

Suitability for requirements engineering in Telematics:

Many RESPECT partners have expertise in the area. Widely used in industry.

How to get it

Surveys have to be tailored to individual requirements. Techniques are widely documented in the literature.

Detailed description of method

Initial steps are the same as for interview design, keeping in mind that semi-structured interviews are similar to open-ended surveys (ie, the issues are known, but the range of user responses to them is not); and structured interviews are similar to closed-ended surveys (ie, the ranger of user responses is pretty well understood, but the strength of each response category needs to be determined).

Questions should be posed in as factual way as possible. Evaluative questions about feelings and interpretations lead to attitude questionnaires and opinion surveys, which are all notoriously difficult to develop by the researcher with little knowledge of psychometrics.

User sampling should be used, and if done properly, surveys should employ a rigorous statistical sampling method to ensure that results are not biased. However, this recommendation is rarely if ever observed in industry. It is sometimes done to offer respondents a little gift in exchange for a returned survey: if chosen appropriately, this can raise response rates to 80% and above. A low response rate may be followed up with either a re-posting or better still a telephonic contact. However, these methods require that users be identified by name to the researcher at least: some surveys may require total anonymity. It is usual to include a short covering letter requesting the respondent to reply and a stamped addressed envelope if possible to make the return as easy for the respondent as can be.

If user information is being kept on computer (as is almost inevitable these days) care should be taken to ensure that the data privacy legislation in your country is not breached, and respondents should be assured of this in the covering letter.

Ethnographic Approach / Contextual Inquiry

Primary Reference Sources

Blomberg, J., Giacomi, J., Mosher, A. & Swenton-Hall, P. (1993) Ethnographic field methods and their relation to design. In: Schuler, D. & Namioka, A. (eds.) *Participatory Design: Principles & Practices.* New Jersey: Lawrence Erlbaum.

Good, M (1989). Seven experiences with contextual field research. SIG-CHI Bull, 20(4) 25-33.

Whiteside, J., J Bennet, and K Holzblatt (1988) Usability engineering: our experience and evolution. In M Helander (Ed) *Handbook of Human Computer Interaction,* 791-817. NY: North Holland Elsevier.

Summary description

The ethnographic approach emphasis the understanding of behaviour in context through the participation of the investigator in the situation being studied as an active member of the team of users involved in the situation. It provides a descriptive report, utilising a range of approaches, mainly informal interviews and observational techniques. The ethnographic approach is essentially the traditional systems analysis approach enriched by contact with sociology and social anthropology. A close variant has been adopted by Digital, called 'Contextual Inquiry'.

Typical Application Areas

In situations with the subject domain is unclear/unfamiliar to the team or where context of work may be expected to have a significant effect.

Benefits

Provides a holistic view of tasks being studied with an emphasis on context and as such helps in understanding work setting. Doesn't require any prior domain knowledge on the part of the investigator. Produces a written document detailing work and culture.

Limitations

Only describes current work practice. The analysis of field notes, video tapes and transcripts is time consuming and must often be completed by the person who recorded them. Ethnography is not a rigorous approach as it relies on inference by the ethnographer as to what is going on in a particular situation. May be difficult to integrate results into a specification.

Cost of use

Requires the personal involvement of an investigator who must be open minded, possess good social skills and have been trained in observational techniques. May require considerable investment of time and effort in order to integrate investigator into the environment to be studied.

Costs of Acquisition

Training or experience in observational techniques would be a prerequisite for investigators.

Suitability for requirements engineering in Telematics:

Some RESPECT partners have experience in this area. Useful as a first step in order to identify issues and work practice in an unfamiliar domain.

How to get it

Detailed directions are given in Blomberg et al (1993), field experiences are referenced in the other reading cited above.

Detailed description of method

Observation and contextual inquiry is best done by a group of researchers who develop a medium- to long-term relationship with a group of organisations who are interested in providing data. The following steps are modelled on the Contextual Inquiry method of Holtzblatt et al:

- Identifying the customer: identify the groups that will be using the new technology or are using similar technology, and arrange to access organisations within the groups that give a cross section of the (potential) market.

- Arranging the visit: write to the targeted organisations identifying the purpose of the visit, a rough time-table, and how much of the employees time will be taken up by the exercise. Ensure that some feedback from the day is possible before leaving. Ensure that the participating organisations understand how many visits you intend to make over the time period of the evaluations.

- Identifying the users: a software product will affect many people throughout the organisation, not just the management or the end users. Ensure that you understand the key users in the organisation whose work will be affected by a new system or changes in the current one.

- Setting the focus: select what aspects of the users' work you wish to make the focus of each visit, and write down your starting assumptions. Make a statement of purpose for each visit, and after the visit, evaluate to what extent you have achieved your purpose.

- Carrying out the interview / observation: stay with the selected users until you have managed to answer the questions you have raised in 'setting the focus'. Very often this may involve inviting the user to directly share and comment on your notes and assumptions.

- Analysing the data: the process of analysis is interpretative and constructive. Your conclusions and ideas from one round of observations are input to the next round, and an evaluation of the results so far should be one of the purposes of subsequent visits.

Focus Groups

Primary Reference Sources

Blomberg, J., Giacomi, J., Mosher, A. & Swenton-Hall, P. (1993) Ethnographic field methods and their relation to design. In: Schuler, D. & Namioka, A. (eds.) *Participatory Design: Principles & Practices*. New Jersey: Lawrence Erlbaum.

Caplan, S (1990) Using Focus Group methodology for ergonomic design. *Ergonomics,* 33.5, 527-537.

Summary description

A focus group brings together a cross-section of stakeholders in an informal discussion group format. Views are elicited by a facilitator on relevant topics. Meetings can be taped for later analysis.

Typical Application Areas

Useful early in requirements specification. Helps to identify issues which may need to be tackled and provides a multi-faceted perspective on them.

Benefits

Allows the analyst to rapidly obtain a wide variety of views from a range of people with widely differing but relevant perspectives.

Limitations

Social factors such as peer pressure may lead to inaccurate reports. Techniques such as Delphi groups can be used to compensate for this.

Cost of use

Meeting facilities and A/V recording facilities if a video record is desirable.

Costs of Acquisition

Facilitators require training in order to keep meetings focused.

Suitability for requirements engineering in Telematics:

Useful, in that focus groups can be applied to a wide range of situations. However, the role of the facilitator is relatively constrained in this kind of technique and more interactive techniques are available.

How to get it

Widely practised. There are a few simple ground rules that the facilitators must adhere to (see below).

Detailed description of method

The facilitator is selected from technical personnel who have a stake in the successful development of the product. A range of issues to be addressed is drawn up. A group of between 6 - 8 representative users is invited to attend. Each focus group meeting should last between 45 and 60 minutes. If the product exists in a demonstrable version, the users should be given a chance to experience it before the meeting.

The facilitator introduces the issues to be discussed, and clarifies his role as an observer and facilitator of free discussion between the users. He may attempt to 'draw out' users who say little, and to suggest that users move to another topic. However he should not intervene directly in the discussion, should not attempt to 'explain' issues which have arisen, and should certainly not be seen in an evaluative role. He should stress that his primary role is 'to listen'.

It is common to tape-record the meeting, but an experienced facilitator should be able to reconstruct a meeting of this length from memory with a few notes to guide him.

Focus groups are useful to enable the design team to understand the vision the user community has of the product being developed, of the kind of uses the product could be put to, and the image the product should have. They can also bring to light annoying features of a product that have not been suspected and could have been missed out completely. It is usual in focus group work that the group itself undergoes a process of change as a result of meeting and discussing the issues. Focus groups are therefore often used when it is planned that new technology will be brought into an organisation in order to find out how the employees envisage that the technology will be used.

Multiple focus groups are frequently used (12 - 20 groups) with the proviso that no user should be present in more than one group to get as wide a range of views as possible. If different facilitators are used for some of the groups, then the result is more convincing still.

Interviews

Primary Reference Sources

Blomberg, J., Giacomi, J., Mosher, A. & Swenton-Hall, P. (1993) Ethnographic field methods and their relation to design. In: Schuler, D. & Namioka, A. (eds.) *Participatory Design: Principles & Practices.* New Jersey: Lawrence Erlbaum.

Fowler, FJ (Jr), Mangione, TW (1990) *Standardised Survey Interviewing*. Sage Publications, Newbury Park.

Summary description

Commonplace technique where domain experts are asked questions by an interviewer in order to gain domain knowledge. Interviewing is not as simple as it may appear and comes in 3 types: unstructured interviews, semi-structured interviews and structured interviews. The type, detail and validity of data gathered vary with the type of interview and the experience of the interviewer.

Typical Application Areas

Interviewing is still the most widely used and abused method of finding out what users want. The apparent simplicity of an unstructured interview lies in the fact that interviewing appears to be a skill which most adults feel they posses from their experience of social conversation. It is characterised be an unconstrained attitude to the agenda and is a technique that is conducted in practically any human endeavour.

Semi-structured interviewing is useful in situations where broad issues may be understood, but the range of respondents' reactions to these issues is not known or suspected to be incomplete.

Structured interviewing should only be carried out in situations where the respondents' range of replies is already well known and there is a need to gauge the strength of each shade of opinion.

Benefits

Useful for identifying possible areas for more detailed analysis. The data gathered provides information on general rules and principles and is faster than observational methods. Interviews are useful for investigating events which occur infrequently and are popular, well known and widely accepted.

Interviews, whatever their limitations, have the extremely positive feature that they give the interviewee the feeling that their input to the process has been taken account of, Extremely useful for getting 'buy in' from the interviewee, and promoting a good image for the representative of the company doing the interview.

Limitations

There is room for considerable bias in what questions are asked and how the answers are interpreted. The interviewer may need to acquire domain knowledge in order to know what questions to ask. What people say often differs from what they really do.

In addition it has been argued that questions with content involving “prestige, social gain and personal circumstances” could bias the response. In a knowledge elicitation context, where the “informant” has the assumed title of “expert”, these factors could play a critical role in the ultimate success of the enterprise.

Cost of use

Requires considerable preparation on the part of the interviewer.

Suitability for requirements engineering in Telematics:

Many partners will have experience of this technique. It is widely accepted and used in industry.

How to get it

Widely covered in the literature.

Detailed description of method

In an elicitation context, the semi-structured interview is generally most fruitful. However, Fowler and Mangione give an excellent account of how to conduct structured interviews should these be needed. There are typically four phases in the interview:

1. The “nurturing” phase. This is the initial warm-up to the interview with pleasantries exchanged, and introductions made.

2. The “energising” phase. Here the area of discourse, and any existing problems are identified.

3. The “body” of the interview. This is the peak phase of activity, where the interviewer is continually probing, ideally asking open-ended questions about issues to understand the range of responses the users produce. It is important at this stage for the interviewer to remain analytical and neutral.

4. The “closing” phase. Summaries may be given as to what has taken place. Subsequent actions are noted, and future planning is made.

Before the interviews, the interview team should decide on a list of issues that will be brought up with each user, and identify strategies and ‘for examples’ in case the users find it difficult to answer to some topics. After the interviews, the design team should pool their notes and present a summary of user reactions to each topic. If more than one interviewee is present, the interviewers may be increased in number but should never exceed the number of interviewees by more than one.

Functionality Matrix

Primary Reference Source

Catterall, B (1990) The HUFIT functionality matrix -In D. Diaper, G. Cockton, D. Gilmore, B. Shackel, (Eds) *INTERCHI'90*, Amsterdam, North-Holland

Summary Description

This is a way of specifying which functions each user type needs. Identifying which tasks are critical allows more time to be paid to these during design and usability testing.

Typical Application Areas

Applicable mainly to systems where tasks are well defined.

Benefits

Enables a realistic consideration of early user and task issues on product functionality specification.

Tailorable to suit varying design processes and in-house styles.

Allows different user types to be considered together in a single process.

Superfluous functions are identified.

Represents a reference in subsequent product lifecycle stages and may be updated in the light of prototyping.

Limitations

Prime focus on functions and features rather than interface appearance.

Can be cumbersome for large numbers of functions.

Cost of Use

Resources required fairly small. Requires input from different user types to complete matrix fully.

Costs of acquisition

Compulsory training not specified although the method may require some degree of practice in order to acquire the basic skills.

Suitability for requirements engineering in telematics

Useful for the specification phase of design process.

Detailed Description of Method

- Identify user groups and enter into matrix rows.
- Identify tasks per user group and enter into matrix rows below respective user group.
- List potential functions and features and enter into matrix columns.
- Identify functions which are critical to task.
- Identify functions which are only occasional used.
- Add new functions or features as required to support gaps in tasks.
- Remove functions with are not required.
- Develop prototype to help create more detailed user requirements specification.

STRUCTURE FOR FUNCTIONALITY MATRIX (EXAMPLE)

Users and Tasks	**Functions**					
User A						Key: ☒ = Critical to Task ☒ = Occasional Use
Task A	☒					
Task B			☒			
Task C					☒	
User B						
Task A		☒				
Task B			☒			
Task C				☒		

Task Allocation Charts

Primary Reference Source

Ip, W. K., Damodaran, L., Olphert, C. W., and Maguire, M. C. (1990) The use of task allocation charts in system design: A critical appraisal. In D. Diaper, G. Cockton, D. Gilmore, B. Shackel, (Eds), *Human-Computer Interaction - INTERCHI'90,* Amsterdam, North-Holland,

Summary Description

A range of task allocation options are established between different users and the computer system to identify the optimal division of labour to provide satisfying job and efficient operation of the whole work process.

Typical Application Areas

Most useful for systems which affect whole work processes rather than single user, single task products.

Benefits

Counteracts the tendency to try and computerise the whole of a working system leaving users to carry out the remaining tasks regardless of the kinds of jobs this produces.

Limitations

Requires some concept of the new system for users to contribute to the process and generate new options. Analysts need good understanding of existing job roles.

Cost of Use

Resources required fairly small. Requires input from people with knowledge of different user types in existing work process.

Costs of acquisition

Compulsory training is not specified although the method may require some degree of practice in order to acquire the basic skills.

Suitability for requirements engineering in telematics

Useful for the specification phase of design process when it is possible to identify the tasks that will be carried out with the application to be designed.

Detailed Description of Method

- Each work process is identified

- For each process, draw up task flow diagram to show existing split of tasks between users and computers and interactions between them.

- Generate at least two charts to show task allocation between user(s) and computer for the new system.

- Comment on the implications for job satisfaction and efficiency for each chart.

- Select chart which is most acceptable to the users or generate new charts.

Standard flow chart symbols may be used to represent process, decision points, system inputs, output, etc. However the actual notation used is not important and an alternative set of symbols may be used if preferred.

Task Analysis

Primary Reference Sources

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

D Poulson (Ed), (1996) *USERfit. A practical handbook on user-centred design for Assistive Technology*. European Commission, DGXIII TIDE Project 1062.

Summary Description

Task analysis is a term covering several techniques. Common elements for the techniques is that they all describe interactions between humans and their environments in a systematic way. Task analysis can be defined as the study of what a user is required to do, in terms of actions and/or cognitive processes, to achieve a system goal. Task analysis therefore is a methodology which is supported by a number of techniques to help the analyst collect information, organise it, and then use it to make various judgements or design decisions.

Typical Application Areas

Task Analysis has classically been used in areas where it is possible to define users' actions at the computer in terms of specific goals for the satisfaction of which the users interact with the software. For situations where the users' interest is likely to be unfocussed, such as browsing or searching, task analysis may still be useful, but yield less tangible results.

Benefits

It should be noted that even if not used in any formal way the principles of task analysis can be of considerable value in focusing attention to relevant issues to consider when designing products from the users' point of view.

Limitations

In the hands of inexperienced practitioners, too much level of detail may be entered into; for systems with diffuse objectives, time may be wasted by attempting to apply task analysis to intractable material.

Cost of Use

Although task analysis may be done cheaply by one experienced practitioner, this is seldom recommended, and a combination of task analysis and some other group-orientated method will be found to yield most benefit.

Costs of acquisition

Task analysis is still largely self-taught, although annual conferences in the human factors area sometimes include tutorials on varieties or applications of task analysis. Within the EUSC a number of partners are specialists in this area (e.g. HUSAT and SINTEF) and they would be able to offer skills training and guidance.

Suitability for requirements engineering in telematics

The potential benefits are so great that all projects are recommended to do at least some task analysis, even if at a relatively high level of abstraction. See also Usability Context Analysis which contains elements of task analysis.

Detailed Description of Method

Tasks can be analysed and described from different perspectives. They might be described in terms of frequencies, logical dependencies, temporal dependencies and in terms of both the human and contextual limitations. The relations between the different constraining factors is also often rather complex. This make a proper understanding of tasks without tools to guide the analysis very difficult.

To design computer systems in a task based way four techniques are of particular relevance:

- Hierarchical Task Analysis (HTA) and Tabular Task Analysis (TTA)
- Timeline Analysis (TA)
- LINK analysis

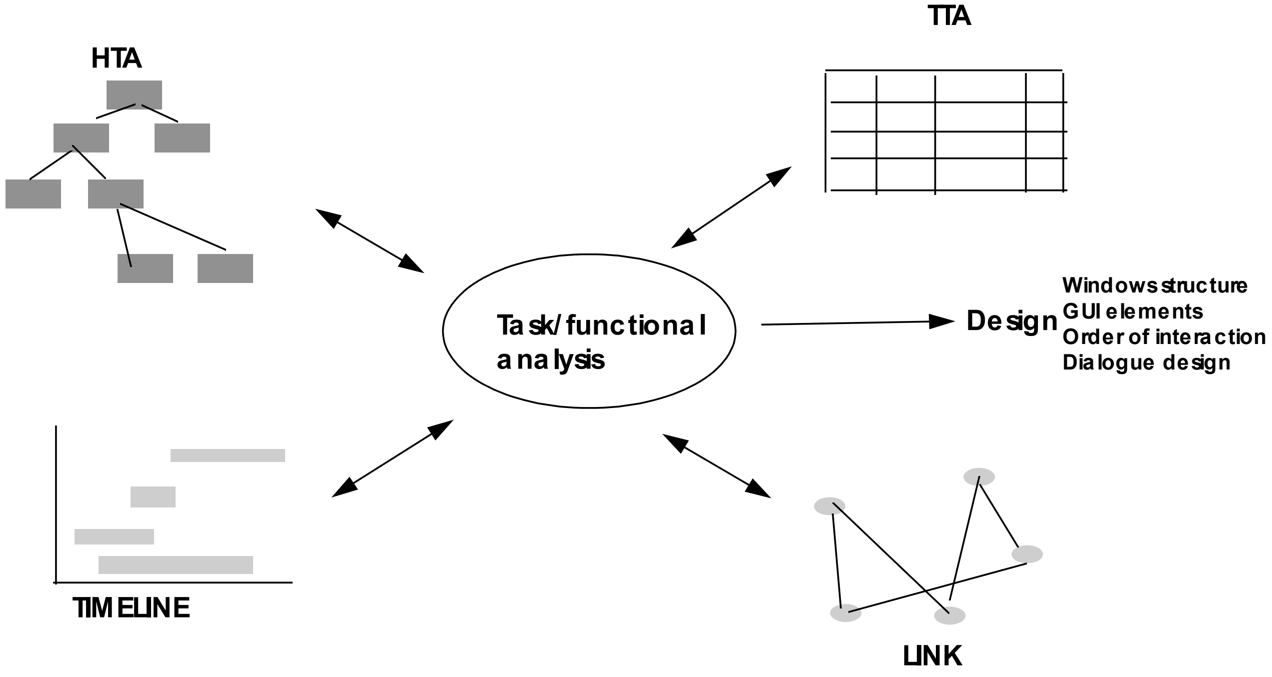

Figure 1. Task analysis varieties

Hierarchical Task Analysis (HTA) and Tabular Task Analysis (TTA)

Hierarchical Task Analysis (HTA), the most commonly used task analysis technique, represents the relationships between tasks and subtasks. It records system requirements and how these can be achieved, including the order in which tasks and subtasks can take place. It can be recorded in a tabular form and/or pictorially. If recorded pictorially it resembles a tree with branches and sub branches as required.

Task analysis involves three closely linked stages: information collection, information recording and analysis.

Information collection procedures include assessing existing documentation (e.g. operations manuals, procedures, safety reports, previous task analysis studies, drawings, screen pictures), establishing what people do in specified circumstances (in both normal and abnormal situations), questioning/interviewing (getting experienced personnel to describe what and how they do things, what information they need, and how they determine if a task has been carried out satisfactorily). Questioning can be formal/informal.

The results from the above information collection procedures should be verified by other experienced personnel.

Information recording will initially be in note form. It will eventually be translated into the final report. One information recording technique is to draw a tree diagram of tasks in their relation to sub-tasks.

The aim of the HTA is to re-describe the task progressively as more information is collected. Tasks are described in terms of elements which are recursively broken down into sub elements and so on. The entries in the boxes of the diagram are the names of the operations and consist only of a few words.

Some tasks may be broken down into greater detail in sequence as a *plan*. A plan describes how the task operations are put together as a complete activity, or shows the circumstances under which one operation is performed rather than another. Plans may be cross referenced and added to the hierarchical table.

In addition, where operators are time dependent or time critical, e.g. small time intervals or simultaneous actions required from an operator, time may be added at the beginning of each action, starting at zero to form a *timeline analysis.*

Information analysis is the final phase. The information gained is used to yield the basic data for design decisions. For example, if a set of displays is to be designed, then the focus will be on information flow across the interface. These sources may then be used to guide design activity.

Another example would be the layout of a set of controls, where the sequence of operations would be important. The sequences illustrated by the task analysis would be isolated and used to guide decisions on optimal layout, for example to minimise the amount of physical movement needed.

Timeline Analysis (TA)

TA is used to document the temporal aspects of tasks. It is well suited to represent task dependencies and possible resource problems in performing tasks. It is often difficult to collect the information needed to produce a TA. In absence of observational data estimates or guesses of temporal relations are used.

LINK analysis

LINK analysis is the simplest of the techniques. It simply demonstrates the frequency of linkage between tasks. It might be used to identify how and how often a user need to navigate from one interface element to the other. The technique can easily be performed. It is most often based on observational data.

The process of designing a system

The HTA methodology is a useful tool to support the design of computer systems. This is particularly linked to the fact that the HTA provides a consistent logical description of the interdependencies of tasks and therefore provide a rational framework for description of possible user interfaces. A typical process of task based design would imply several steps.

First it is important to establish a description of the situation here and now, This will serve as a communication tool with the users and the user organisation and ensure that the starting point is a proper understanding of the tasks as they are performed in the work system.

This preliminary HTA is transformed into a Tabular Task Analysis (TTA). In this format judgements about the allocation of function between man and machines is done. At the same time new tasks that are created by the introduction of a new IT system is written in. We should like to stress that this is not a detailed analysis down to keystroke level. It is possible to construct the basic commands for the new system at this stage.

Based on the TTA a new HTA describing the dependencies of tasks interacting with the new system is developed. This graphical description is used to identify tasks that are performed many places in the system, tasks that are linked to specific goals (i.e. start up, calculate data etc.) and tasks that need further analysis. If done properly this format can be used to describe the windows structure and dialogue elements of an interface.

Wizard of Oz Prototyping

Primary Reference Sources

Maudsley, Greenberg & Mander (1993) Prototyping an intelligent agent through Wizard of Oz. *Interchi '93 Conference Proceedings,* pp 277-284.

Summary description

This approach involves a user interacting with a computer system which is actually operated by a hidden developer - referred to as the "wizard'" The wizard processes input from a user and simulates system output. During this process the user is led to believe that they are interacting directly with the system. This form of prototyping is beneficial early on in the design cycle and provides a means of studying a user's expectations and requirements. The approach is particularly suited to exploring design possibilities in systems which are demanding to implement.

Typical Application Areas

The approach is highly applicable to "intelligent interfaces" which feature agents, advisors and/or natural language processing.

Benefits

This approach allows usability requirements and issues to be explored at an early stage in the design process, particularly for systems which go beyond readily available technology.

The member of the design team who plays the wizard can gain valuable insights from the close involvement in the user's activity.

Limitations

The person playing the role of the wizard must appreciate the functionality of the proposed system in order to provide a convincing representation.

This approach requires a higher commitment of resources than other approaches to prototyping such as those that rely on simple paper-based materials.

It may be difficult to carry this out effectively in situations where there is a large graphic element in the interface.

Cost of use

Two computer systems would be required, one each for the user and the wizard. Two staff are required to conduct the evaluation - one to play the wizard, another to instruct the user and record the session. The wizard should be an experienced member of the design team so that system responses are logical and not beyond the realms of possibility. The time overhead largely depends upon the task domain and the number of users exposed to the prototype.

Costs of Acquisition

Compulsory training is not specified. However, many reports on the approach stem from research papers where the details of the technique are sketchy. The approach is however, likely to require experience and / or training for the practitioner to reach competency.

Suitability for requirements engineering in Telematics:

Some existing expertise is available from RESPECT partners. Industry could adopt the principles, it is already applied within Apple. While the approach may lack the general applicability of other prototyping approaches it is particularly suited to multimedia and telematics applications.

How to get it

See primary reference source for more information.

Detailed description of approach

The general procedure for implementing this approach is outlined in the following.

1. Firstly, allow enough time to fabricate the wizard of Oz prototype, design some tasks, recruit users, conduct the evaluation of the prototype and report the results.

2. Allocate the role of wizard and the role of facilitator to the relevant staff.

3. Assemble the necessary equipment and inter-connecting software.

4. Select appropriate users to test the prototype, try to cover the range of users within the target population.

5. Prepare realistic task scenarios for the evaluation.

6. Pilot the evaluation procedure and ensure the wizard is well practised in playing the role of the computer.

7. Ensure recording facilities are available and functioning.

8. Conduct each session. The facilitator instructs the user to work through the allocated tasks interacting and responding to the system as appropriate.

9. Conduct post-session interviews with the users, drawing upon pre-set questions and issues raised during the use of the prototype.

10. Debrief the users and thank them for their co-operation.

11. Analyse information obtained, summarise observations and user evaluations. Consider the themes and severity of the problems identified.

12. Summarise design implications and recommendations for improvements and feed back to design team. Video recordings can support this.

13. Where necessary refine the prototype and repeat the above process

The evaluation of wizard of Oz prototypes yields information regarding user expectations and requirements for novel technologies. The data gained from this approach may be used to refine the initial prototype, and inform subsequent evaluations.

Paper Prototyping

Primary Reference Source

Rettig, M (1994) Prototyping for tiny fingers. *Communications of the ACM*, April, Vol.37,No.4.

Summary description

This method features the use of simple materials and equipment in order to create a paper-based simulation of an interface or system. Paper prototypes provide a valuable and cost-effective means to evaluate and iterate design options before a team gets committed to one implementation. Interface elements such as menus, windows, dialogues and icons can be sketched on paper or created in advance using card, acetate, pens etc. The result is sometimes referred to as a low-fidelity prototype. When the paper prototype has been prepared a member of the design team sits before a user and 'plays the computer' by moving interface elements around in response to the user's actions. The user makes selections and activates interface elements by using their finger as a mouse and writing 'typed' input. A further person facilitates the session by providing task instructions and encouraging the user to express their thoughts and impressions. Notes may be made by other observers or a video record may be created.

Typical Application Areas

The method has wide applicability. However, it is most suitable in contexts where it is easy to simulate system behaviour or when the evaluation of detailed screen elements is not required. Paper prototyping is appropriate for the early stages of the design cycle where changes can be readily made before there is a commitment to one implementation.

Benefits

Usability problems can be detected at a very early stage in the design process (before a commitment to code has been made).

Communication and collaboration between designers and users is encouraged.

Paper prototypes are quick to build and refine, and thus support iterative design and multiple evaluations.

Only minimal resources and materials are required to convey product feel.

The technique can be utilised by those with little or no human factors expertise.

Limitations

Because of their simplicity, paper prototypes do not support the evaluation of fine design detail.

Due to the use of paper and a human operator, this form of prototype can not be reliably used to simulate system response times.

The individual playing the role of the computer must be fully aware of the functionality of the intended system in order to simulate the computer.

Cost of use

The technical resources required are minimal. Materials such as paper, card, adhesives and markers are needed to create the actual prototype. In addition, some means of recording the interactions between user and prototype is required (e.g. video camera). The method also needs one individual to play the role of the computer or system, and another to act as a facilitator. Costs may also be incurred when recruiting users and allocating time to manage each evaluation session.

Costs of Acquisition

Compulsory training is not specified although the method may require some degree of training in order to acquire the basic skills.

Suitability for requirements engineering in Telematics:

This method has a role in RESPECT and is perhaps one of the most accessible approaches to prototyping.

How to get it

Some RESPECT partners are able to offer courses and hands-on experience.

Detailed description of method

The following material outlines firstly a general procedure for implementing this method, and also indicates the kind of information that is produced. Then a more detailed overview of two activities that can be carried out with paper prototyping is given: *sketching* and *user testing*.

1. Firstly, allow enough time to create the prototype, design some tasks, recruit users, conduct the evaluation of the prototype and report the results.

2. Assemble the necessary materials. Construct the paper prototype, using separate stock for menus, dialogue boxes and any element that moves or changes appearance.

3. Select appropriate users to test the prototype, try to cover the range of users within the target population.

4. Prepare realistic task scenarios for the evaluation.

5. Pilot the evaluation procedure and practice playing the role of the computer.

6. Ensure recording facilities are available and functioning.

7. Conduct each session, by manipulating the paper prototype as the users work through the tasks.

8. The facilitator provides the task instructions and explores the user's impressions and intentions through appropriate questions.

9. If observers are present they can make notes of problem areas and potential solutions on cards during the session for later scrutiny and prioritisation.

10. Conduct post-session interviews with the users, drawing upon pre-set questions and issues raised during the prototype evaluation.

11. Debrief the users and thank them for their co-operation.

12. Analyse information obtained, summarise observations and user evaluations. Consider the themes and severity of the problems identified.

13. Summarise design implications and recommendations for improvements and feed back to design team. The video recordings can support this.

14. Where necessary refine the paper prototype and repeat the above process

The evaluation of paper prototypes provides an opportunity to collect early design feedback. This results in recommendations for the refinement of the initial prototype, which can form the basis for the evaluation of further prototypes.

Sketching

This technique involves members of the design team and potential users, producing sketches or designs of the ideas that they wish to input to the design process. The objective is to enhance user participation in the design process and collaboration between designers and users. The easiest way to set up a sketching exercise is to use a flipchart or whiteboard with everyone sitting around it presenting and reacting to ideas. An electronic whiteboard has the advantage of producing printouts of the 'screen' which can then be photocopied to the group before rubbing out the earlier ideas to consider new ones (e.g. SILK Landey 1994). Alternatively, users may sketch their own ideas individually which they can then each present in turn to the group.

A more sophisticated method involves presenting users with a set of paper, cardboard or plastic interface elements which they can lay out, on a flat surface, in what they feel is an appropriate way. Again designs may be discussed and developed as a group or individually. An example of such a kit is PICTIVE created at Bellcore by M. Muller (1992). This method is effective when the basic control types of a future interface are known but user feedback on a suitable layout is required. The method does not require users to draw the interface although they can supplement the design with additional elements or annotations to add contents.

The success of the exercise relies on the presence of a facilitator chairing the meeting. The main role of this person is to ensure that the group stay focused upon the design problem and ensuring that every member of the group is given the opportunity to stand up and present his or her own ideas. Another role is to summarise all the ideas after the session for presentation to a design team meeting.

The outcome of a participatory design exercise will be a series of ideas for screens, layouts, navigation structure, that can be evaluated by the design team to assess their technical feasibility and usability. They will thus serve as a first draft of design specifications. Various techniques are possible to preserve the designs : the sessions can be recorded on video, the paper mockups may be stuck down onto a base sheet, covered with clear plastic, photographed or simply photocopied. They may then be mocked-up on screen or in hardware form to further test the ideas. The paper screen designs can also be used as a 'walk-through' exercise to get reactions from other end users.

User testing

Early pilot studies of a system idea can be carried out using paper versions of screen displays. These tests can be run to compare design alternatives, or to contrast with current procedures. The paper-prototype should be designed to contain the screens or interactive sequences needed to perform a series of typical tasks. During the test, a member of the design team sits in front of a user and 'plays the computer' by moving interface elements around in response to the user's actions. Alternatively they may write messages on 'post-its' to represent elements such as pull down menus or dialogue boxes. The user makes selections and activates interface elements by using their finger as a mouse and writing 'typed' input. A further person facilitates the session by providing task instructions and encouraging the user to express their thoughts and impressions. Notes may be made by other observers or a video record may be created.

A variant of the paper walkthrough is to produce the screens as a set of cards. Users are asked to order the cards in the sequence that seems most appropriate for the activities they must carry out. The objective is to focus on the flow of user tasks and identify the appropriate structure of the task-sequence. (e.g. CARD: Collaborative Analysis of Requirements and Design, M.Muller 1992). Cards may also be used to elicit data or menu structures from the user. Each of the data elements or menu options may be written out on cards and laid out infront of the user. The user then places the cards into piles to represent suitable groupings. Common groupings between different user subjects can be used to structuring the system data or menus.

The method is particularly useful for assessing user reactions to layout, data structures, and sequencing of screens. However it is hard to convey to the user the feeling of interacting with the new system.

The tests allow usability problems to be detected and recommendations be made at a very early stage in the design process, before a committing the design to code. Thus it supports iterative design and multiple evaluations. Further redesign can be carried out on paper, or the design can be developed on screen to test the dynamic interactive features.

References

Landey J., (1995) *Interactive Sketching for User Interface design* Proceedings of ACM CHI 95 Conference pp.63

Muller, M. J. (1992) *Retrospective on a year of participatory design*. Proceedings of ACM CHI 92 Conference pp.455

Remington R.,(1994) *CHIRP : The computer human interface Rapid Prototyping and design assistant toolkit*. Proceedings of ACM CHI 94 Conference pp.455

Rettig, M (1994) *Prototyping for tiny fingers*. Communications of the ACM, April, Vol.37, No.4.

Scenario Building

Primary Reference Sources

Clark, L. (1991) The use of scenarios by user interface designers. In Daiper, D. & Hammond, N (Eds.) *HCI'91 conference proceedings*, pp.103-115.

Nielson, J.(1991) Paper versus computer implementations as mock up scenarios for heuristic evaluation. In Daiper et al. (Eds) *HCI-Interact '90 conference proceedings,* pp. 315-320

Requirements acquisition through scenario building, Spool J., M. (1994) *Chi'94 Tutorial Notes.*

Summary description

Scenarios are characterisations of users and their tasks in a specified context. They offer concrete representations of a user working with a computer system in order to achieve a particular goal. The primary objective of scenario building is to generate usability requirements or targets. The scenarios are created by the members of a development team who then role play what it is like to be a user in order to form a group-wide user model based on consensus. Scenarios also offer the opportunity to explore the implications of design options and communicate interface issues to colleagues for comment and critical feedback. Actual users are not involved in this process. Scenarios may be utilised in the early phases of a development cycle as a means of defining end user requirements. The method may also substitute for user and task analysis which is rarely carried out in practice.

Typical Application Areas

Scenarios have generic relevance for the design of a broad range of systems.

Benefits

Scenario building encourages designers to consider the characteristics of the intended users, their tasks and their environment.

Usability issues can be explored at a very early stage in the design process (before a commitment to code has been made).

Scenarios can help identify usability targets and likely task completion times.

The method promotes developer buy-in and encourages a user-centred design approach.

Scenarios can also be used to generate contexts for evaluation studies.

Only minimal resources are required to generate scenarios.

The technique can be used by developers with little or no human factors expertise.

Limitations

Scenarios are of particular relevance when considering user and task characteristics rather than the detail of interface design and layout.

Cost of use

The resources required are minimal and scenarios should be quick to produce (perhaps just a few hours?). An experienced moderator is recommended for the sessions in which the scenario is explored, and up to 2 hours per session may be required.

Costs of Acquisition

There are no purchase costs and although running costs should be minimal a user may require a limited amount of experience in order to become familiar with the technique.

Suitability for requirements engineering in Telematics:

RESPECT partners may well have existing experience of employing scenarios, whilst the transfer of this method to industry should be readily achieved. The method also bears a close relationship to Usability Context Analysis as used by NPL. In light of the benefits of the method it should receive the attention of RESPECT.

How to get it

The method is not proprietary and appears to be widely used in both industry and academia.

Detailed description of method

The principle steps for this method are as follows.

1. Gather together the development team and other relevant stakeholders under the direction of an experienced facilitator.

2. Identify intended users, their tasks and the general context. This information will provide the basis for the scenarios to be created by the development team.

3. Functionally decompose user goals into the operations needed to achieve them.

4. Assign task time estimates and completion criteria as usability targets

5. The session can be video-taped for later review or transcribed for wider distribution

6. The results from scenario building sessions can be used to plan user-based evaluations

Storyboarding / Presentation Scenarios

Primary Reference Sources

Androile S (1991) *Storyboard Prototyping.* QED Information Sciences, Wellesley, MA.

Summary description

Storyboards are sequences of images which demonstrate the relationship between individual screens and actions within a system. A typical storyboard will contain a number of images depicting features such as menus, dialogue boxes and windows. The formation of these screen representations into a sequence conveys further information regarding the structure, functionality and navigation options available within an intended system. The storyboard can be shown to colleagues in a design team as well as potential users, which allows others to visualise the composition and scope of an intended interface and offer critical feedback. This method can be used early in the design cycle where the use of storyboards supports the exploration of design possibilities and the early verification of user requirements.

Typical Application Areas

This method is of general relevance, especially to products in which there a complex structure of information is being developed.

Benefits

Feedback can be gained on system functionality, style and also navigation options early on in the development cycle where changes can be more easily implemented.

- The method promotes communication between designers and users.
- Storyboards can be created quickly and easily.
- Only minimal resources and materials are required.
- The technique can be utilised by those with little or no human factors expertise.

Storyboards provide a platform for exploring design options via a static representation which can be shown to both potential users and members of a design team. This can result in the selection and refinement of particular design options. This filtering process can be a valuable precursor to prototyping activities.

Limitations

Storyboards may lack the interactive quality of prototyping methods.

Because of their simplicity, storyboards do not support the evaluation of fine design detail.

Storyboards do not accurately convey system responsiveness.

Cost of use

The technical resources required to create storyboards are minimal, and include drawing tools (both computer and non computer-based), paper, card, pens and adhesives. Furthermore, the time and human resources are low.

Costs of Acquisition

There are no explicit training costs, as with other methods some initial exposure to the method may be required to acquire the basic principle.

Suitability for requirements engineering in Telematics:

The method appears to be simple and could be readily adopted by consortium partners and industrial concerns. This method is valuable as a bridging activity between early data gathering methods such as focus groups and interviews and the various prototyping methods.

How to get it

This method is well documented in the literature.

Detailed description of method

The general procedure relating to this method follows.

1. Give consideration to the scenarios of use which the storyboard will reflect. A storyboard may represent several activities such as entering, saving or printing information. Alternatively a separate storyboard may be created to represent each distinct theme.

2. Construct the storyboard as a sequence of screen representations. using separate images to reflect changes in system appearance. Thus the storyboard indicates the availability and purpose of dialogue windows, menu items, toolbars and icons.

3. The elements of a storyboard can be annotated with explanatory captions to aid audience understanding and evaluation.

4. The completed storyboard can be shown to design teams as well as intended users to solicit evaluative feedback. Several storyboards can be created and shown to an audience in order to explore different design options.

5. It may be useful to video or audio record the feedback sessions for later review or to show to other colleagues.

6. Further storyboards can be created and evaluated in light of feedback.

Empathic Modelling

Primary Reference Sources -

D Poulson (Ed), (1996) *USERfit. A practical handbook on user-centred design for Assistive Technology*. European Commission, DGXIII TIDE Project 1062.

Summary Description -

This method has so far been mainly developed for use with disabled users. With empathic modelling the designer/developer tries to put themselves in the position of a disabled user. This is done by simulating the disability by various techniques. Similar methods is widely used in many application areas, but rarely referred to as empathic modelling.

Typical Application Areas -

General method which could be applied to a broad range of applications.

Limitations -

A disability that is "simulated" in this way will have a lot of differences from a real disability. There is a big difference between the situation an investigator puts himself in, and the situation of an actually disabled person. To make the "simulation" real a lot of work need to be invested in research on the characteristics of the user group.

Benefits -

One of the advantages of the method is that those that need the information, can get it directly without the mediation of a user. The method can provide useful information when it is difficult to involve real users.

Costs of Use -

The costs involved will vary considerably, it all depends on the ambition of the project. Well designed, longitudinal studies, may demand a lot of resources, while small, informal, "impressionistic" studies are practically free.

Costs of Acquisition -

Not yet known.

Suitability for requirements engineering in Telematics -

The approach can be used in early problem definition stages of design to increase the awareness of designers of the implication of design for specific disabilities

How to get it -

SINTEF Unimed Rehab, PO Box 124, Blindern, N-0314 Oslo, Norway.

References

Pastalan, L.A. 1982, Environmental design and adaptation to the visual environment of the elderly. LISS, A.R. Aging and visual functions. 1982, pp323 - 333.

Video Prototyping

Primary Reference Sources

Vertelney, L (1989) Using video to prototype user interfaces. *SIGCHI Bulletin*, V21 (2), pp 57-61.

Summary description

This method allows designers to create a video-based simulation of interface functionality using simple materials and equipment. Interface elements are created using paper, pens, acetates etc. For example a start state for the interface is recorded using a standard camcorder. The movements of a mouse pointer over menus may then be simulated by stopping and starting the camcorder as interfaces elements are moved, taken away and added. Users do not directly interact with the prototype although they can view and comment on the completed video-based simulation.

Typical Application Areas

Wide application potential, particularly suited for simulating interface functionality. However, it must be possible to simulate the interface elements with basic materials. The method is relevant in the early stages of the design cycle to demonstrate design options and concepts.

Benefits

Usability problems can be detected at a very early stage in the design process (before a commitment to code has been made). Provides a dynamic simulation of interface elements that can be viewed and commented on by both design teams and intended users. Minimal resources and materials are required to convey product feel. The technique can be utilised by those with little or no human factors expertise.

Limitations

Staff familiar with the functionality of the intended system are required to create the video prototype. The method does not actually capture a user interacting with the prototype. Because of the use of simple materials, video prototypes do not support the evaluation of fine design detail.

Cost of use

As with paper prototyping only simple materials are required to create the elements of the prototype to be committed to video-tape. These include paper, acetate, pens and adhesives. A video camera is also required to capture and replay a simulation. The method calls for two people, one to manipulate the interface elements and so operate the computer and the other to control the camera. Little video production expertise is required, although it could be time-consuming to create more complex sequences using stop-motion animation. Regular use would incur basic costs for paper materials and video-tapes.

Costs of Acquisition

Compulsory training is not specified, although the technique may require some experience to reach competency in the basic principles.

Suitability for requirements engineering in Telematics:

This is a novel and interesting method, the principles of which are adopted in some sectors of industry. However, it lacks the interactive element of other prototyping methods and would perhaps be most suited for demonstration purposes where larger audiences are involved.

How to get it

This is not described as a proprietary method. It has been practised within America, particularly at US West and Apple. A good start is the reference cited above; SINTEF has experience of this method in the EUSC.

Detailed description of method

The general procedure relating to this method is outlined below. The method parallels that described for paper prototyping in several respects, although the emphasis on creating a video-based simulation is a distinct feature, as is the absence of users who directly interact with the prototype.

1. First, allow enough time to create the prototype, design some scenarios of use for demonstration purposes, and produce the video-based simulation. It should be remembered that even brief sequences of stop-start animation can be time consuming.

2. Assemble the necessary materials. Construct the paper prototype, using separate stock for menus, dialogue boxes and any element that moves or changes appearance. A paper-based mouse pointer for instance can be attached to the end of a strip of acetate so that it can be moved without the operators hands appearing on the video recording.

3. The person manipulating the interface elements should practice playing the role of the computer.

4. Ensure recording facilities are available and functioning. Ideally the camera should point directly at the prototype, perhaps by being mounted above a table where the materials are placed.

5. One person should manipulate the elements of the paper prototype while another person controls the video camera. For example a menu selection can be captured by initially filming a shot of the paper desktop and subsequently filming a brief sequence as the mouse pointer is moved on a transparent arm to a menu item. The video camera is then paused while a paper representation of a menu is placed under the camera, filming then continues while the mouse pointer passes down the menu items.

6. Once completed the video-based prototype can be shown to design teams as well as intended users to solicit evaluative feedback. Several video prototypes can be created and shown to an audience in order to explore different design options.

7. Where necessary the prototype can be refined and the above process repeated. The use of video prototypes supports the exploration of design options by providing a dynamic simulation which can be shown to both potential users and colleagues. This can result in recommendations for the refinement of the initial prototype.

Diary Methods

Primary Reference Source

Kirakowski, J and Corbett, M (1990) *Effective Methodology for the Study of HCI.* North Holland Elsevier.

Summary Description

Diary methods require the informants to record activities they are engaged in throughout a normal day. The structure of diaries varies from unstructured or open ended where the informant writes in with his own words, to highly structured tick-box questionnaires. The technology involved ranges from paper and pencil based techniques, to the use of tape and video diaries, to computerised on-line questionnaires administered by computer.

Providing users with a portable cassette voice recorder and asking them to 'make a note' of their experiences with whatever is the focus of the diary study is a useful way of capturing the users' interest and to get on-the-spot notations. The data usually has to be content analysed. Kirakowski & Corbett give an account of this process.

Typical Application Areas

They can be used during the initial phases of design, to provide a picture of the activities the user group engage in. The frequency, duration and difficulty of performing tasks can be recorded in order to establish an understanding of the users difficulties and the context wherein a technical aid will be introduced. They can also be used to provide design feedback, or evaluate products.

Limitations

The methods demands some effort from the users. If structured methods are used, as with questionnaires, it can be difficult to interpret results, due to lack of context information. A lot of data can be collected, this may lead to a lot of work to analyse the information. The method relies on the informants ability and motivation for filling in the information.

Benefits

Provides information that may be forgotten in an interview, or by means of other methods. Involves the users in an active manner. Provides an understanding of the activities the users engage in. The method is useful to obtain real life measures about the activities and technology the users are surrounded by.

Costs of Use

Relatively low cost. May take some time to analyse results, especially when an informal approach is used. A structured diary or computer based inquiry will demand some resources for preparation.

Suitability for Telematics Engineering

Provides vital information about the users, tasks and environment in an efficient and fluid way.

Rapid Prototyping (software or hardware based)

Primary Reference Source

Isensee, S, & J Rudd (1966) *The Art of Rapid Prototyping.* International Thomson Computer Press, London.

Summary Description

This method is concerned with developing different proposed concepts through software or hardware prototypes, and evaluating them. In general the process is termed ÔrapidÕ prototyping. The development of a simulation or prototype of the future system can be very helpful, allowing users to visualise the system, and provide feedback on it. Thus it can be used to clarify user requirements options.

Rapid prototyping is described as a computer-based method which aims to reduce the iterative development cycle. Interactive prototypes are developed which can be quickly replaced or changed in line with design feedback. This feedback may be derived from colleagues or from the experiences of users as they work with the prototype to accomplish set tasks.

Within software engineering circles the method is closely associated with user interface management systems and various design support tools. The latter tools offer the designer libraries of procedures and graphical interface elements for defining the softwareÕs logical structure and Ôlook-and-feelÕ. Here the title refers to an approach adopted by software developers in which the prototypes exhibit a higher fidelity with the end product than those created as part of other methods such as paper prototyping.

The method requires more sophisticated technical resources than is the case with low-fidelity prototyping methods which rely on paper materials. An additional cost of use is the level of human expertise required to master the supporting development tools, along with the time necessary to implement a software prototype. If user-based testing will be carried out with the developing prototype, then resources for this activity will also be needed.

Benefits

¥ Permits the swift development of interactive software prototypes.

¥ Prototypes created by this method have a high fidelity with the final product.

¥ The prototypes created under this method support metric-based evaluations.

Limitations

¥ The method requires software development skills.

¥ Although rapid, the method is more time consuming than other approaches.

¥ The resources required are greater due to the need for software and hardware rather than paper and pens.

Cost of use

Requires programming or model building skills to produce the prototypes. A number of prototype iterations may be carried out or a parallel design approach may be adopted. The process of obtaining user feedback will also incur a certain amount of cost in terms of time and effort.

Suitability for requirements engineering in Telematics

Insofar as this is a generic approach which involves end users in the development process, Rapid Prototyping is to be recommended. The necessary tools for adopting this approach in a cost-effective way throughout the development lifecycle may not be available for some application areas. See Isensee & Rudd for more details about available tools.

Procedure

A general procedure for adopting the rapid prototyping method is outlined below.

1. Firstly, allow enough time to create the prototype. If the prototype is to be evaluated with users then allow time to design relevant tasks, recruit the users, evaluate the prototype and report the results.

2. Assemble the necessary equipment, including the hardware and software tools necessary to create the interactive prototype.

3. Develop the prototype itself.

4. Select appropriate users to test the prototype, trying to cover the range of users within the target population. A facilitator will also be required to instruct the users and run the evaluation.

5. Prepare realistic tasks to occupy the users as they work with the prototype.

6. Pilot the evaluation procedure and ensure the prototype can be used to accomplish the tasks.

7. Ensure recording facilities are available and functioning.

8. Conduct each session. The facilitator instructs the user to work through the allocated tasks, interacting with, and responding to, the system as appropriate.

9. If necessary additional information can be obtained by interviewing users following their use of the prototype.

10. Debrief and thank the user.

11. Analyse the obtained information and then summarise the observations and user evaluations. Determine the themes and severity of the problems identified.

12. Summarise design implications and recommendations for improvements and feed back to design team. Video recordings can support this.

13. Where necessary refine the prototype and repeat the above process.

RAD (Rapid Application Development) and JAD (Joint Application Design) Workshops

Primary Reference Sources -

Andrews, D.C. (1991) JAD: A crucial dimension for rapid applications development'. Journal of systems management, March 23-31.

Summary Description -

RAD:- Workshops are set up in which 8-20 individuals make decisions through the consensus building leadership of a trained, unbiased facilitator who is not a stakeholder in the future system. A number of different formats for the method are offered. One variation produces formal outputs such as entity-relationship models, which can be input directly into the system specification.

JAD is a specific variation developed within IBM. Here users and information systems professionals are drawn together to design a system jointly in a facilitated group session. Six roles are defined including: session leader, user representative, specialist, analyst, an information systems representative and an executive sponsor. A 20% to 60% increase in productivity over traditional design methods is claimed.

Benefits

- Improves quality and speed of system specification and design.

- Is integrated with current structured methods and CASE tools.

- Promotes co-operation, understanding and teamwork among the various user groups and information systems staff.

- Can be used at various levels of detail, including: concept, requirements, specification and design.

Limitations

- Workshops tie up stakeholders often for several days at a time.

- Appears to have only been used effectively for smaller systems.

- Group sessions needs skilful running to ensure group they are productive and reach a consensus.

Costs of Use

Workshops require up to 3 weeks of planning. Post workshop activities may also be required.

Suitability for requirements engineering in Telematics

The essential structure of method may be appropriate to adopt within projects where there is a need to integrate the interaction of various groups or Centres who may be working to different goals.

How to get it

High level method documented in the literature. Specific details could be obtained from author. Cost unknown.

Main stages of the method

1. JAD Planning session (1-5 days).This session includes: participation orientation, definition, definition of high-level requirements, bounding system scope, identify designs to be produced, identify participants, schedule design sessions, deal with documentation issues.

2. JAD Design session (3-10 days). This session includes :participation orientation, review and refine requirements and scope, develop workflow diagrams and descriptions, identify system data groups and functions, specify processing requirements, deal with documentation issues.

3. JAD teams are given guidance and proformas which can be used as a basis for the agenda for group sessions. The proformas may be compiled into a workbook, preferably customised to the problem situation, for the team to complete either during session or as part of a follow-up. They may include participant matrix forms, issues forms, estimating assumptions forms, screen layout forms, report layout forms, interface description forms, and function description forms.

Laboratory-Based Observation

Primary Reference Sources

Rubin, J. (1994) Handbook of Usability Testing. John Wiley, NY.

Nielsen, J (1993) Guerilla HCI: Using discount usability engineering . In R Bias and D Mayhew (Eds) *Cost Justifying Usability*. Academic Press, Boston.

Summary description

This is an approach to studying user behaviour in the laboratory, and may be used at practically any stage in the development process when there is a representation of the software that users can interact with. The book by Rubin is cited on account of its clarity of exposition, but this approach is documented in many sources. J Nielsen advocates an approach he calls 'discount usability engineering'.

Typical Application Areas

These approaches may be used at any stage in the process, although 'discount usability engineering' assumes the existence of a prototype that can stand on its own.

Benefits

The method can be seen as an alternative to more extensive trials, which may take place later, prior to a release of a product.

Limitations

If planning and preparation work is underestimated, results may be of little value.

Cost of use

Basic needs for this approach are paper forms for the observers, and equipment to show the interface on a screen or wall. This equipment could include an overhead projector, a computer screen, a barco, or a video based system with monitor. Optional materials needed would be a video camera to record the test session, log software and a computer with a simple data logging program.

Costs of Acquisition

This approach is public domain. The work of Rubin is particularly useful as a starting point.

Suitability for requirements engineering in Telematics:

This method deserves consideration as it directly involves end users. While it can be applied early in the design with a paper and pencil prototype, it also integrates well with Performance Measurement and other metrics oriented methods that can be applied at a later stage of the design.

Detailed description of method

For this approach to be successful it is crucial that thorough pilot testing be conducted by the investigator before introducing users into the lab. This pilot testing has a number of steps:

1. The description of the tasks

2. A simple test procedure with written instructions

3. A predefined format to identify problems

4. A debriefing interview guide

5. A procedure to rank problems

6. An estimation of the proportion of problems identified (optional step)

If pilot testing is underestimated then results may be of little value. This planning will identify usability goals. Roles may then be distributed within the design team, and planning for the test can be administered. The number of subjects needed is then estimated, and a written procedure is made, with at least one pilot test being performed. Useful results can be obtained with three or four users; some data gathering techniques may need larger user sample sizes in order to minimise bias.

Using the test material, realistic scenarios will be recorded, along with demonstrations and instructions. After the session, data may then be collected from the subjects using interviews and questionnaires. Once trials are run, data is analysed, metrics (if any) are calculated, and problem severity is prioritised in an implications report.

Co-operative Evaluation

Primary Reference Sources

Monk, A., Wright, P., Haber, J., and Davenport, L. (1993) Improving your human-computer interface: A practical technique. Prentice Hall International (UK) Ltd.

Summary description

This is a cost-effective technique for identifying usability problems in prototype products and processes. The technique encourages design teams and users to collaborate in order to identify usability issues and their solutions. Users work with a prototype as they carry out tasks set by the design team. During this procedure users explain what they are doing by talking or Ôthinking-aloudÕ. An observer records unexpected user behaviour and the userÕs comments regarding the system. The observer also actively questions the user with respect to their intentions and expectations. This provides qualitative information concerning any difficulties that the users experience and the features or interface elements which give rise to these problems. The method is applicable in the early phases of the design cycle once a prototype is available for evaluation or at a later stage when an existing product is to receive further development.

Typical Application Areas

This method can be employed to evaluate any working prototype or implemented system. It provides a valuable means of verifying and eliciting user requirements.

Benefits

Usability problems can be detected early in the design process.

Information on the userÕs thought processes as well as their actions can be obtained.

Communication between designers and users is promoted.

The method can be utilised by those with little or no human factors training.

Limitations

The technique depends upon the use of appropriate tasks and is therefore unsuitable in the very early stages of design when the task domain has not been clearly defined.

The close involvement of designers in this evaluation technique makes it unsuitable in circumstances which require an independent assessment, such as quality assurance

The recordings can contain large amounts of information which may be time consuming to analyse.

Cost of use

This method requires a functional prototype to evaluate and one or more users to take part (five is the recommended maximum). A location for the evaluation and realistic tasks to occupy the users will also be needed. At least one person will be required to act as observer and prompt the user with questions. This person does not have to have expertise in human factors as the method is well documented. Some means of recording the userÕs comments and actions will also be needed (e.g. audio or video recording, system logs, notebooks). The number of days required will depend on the number of users taking part, the duration of each user session and the time required to analyse the information and produce the report. Assuming the participation of five users (for two hours each) a co-operative evaluation could be completed within 5-10 days.

Costs of Acquisition

Compulsory training is not specified and the primary reference source contains much useful background and examples.

Suitability for requirements engineering in Telematics

This method may prove valuable as an adjunct to prototyping activities and prescribes a means for eliciting user feedback.

How to get it

The method is very fully documented (see the primary reference source).

Main steps of the method

The general steps for this method are shown below. This is followed by an indication of the kind of output produced by this method.

1. Recruit users. Identify the target user population, i.e. those who will use the system in the future and select users directly from or akin to this target population.

2. Prepare tasks. Select realistic and appropriate tasks which test those features of the system that people will actually use in their work, and those features implemented in the prototype. Write task instructions for the users. Ensure that tasks are clearly expressed, perhaps by conducting a pilot session and working through the tasks and instructions. This will also indicate how much time is required from each user.

3. Prepare any pre-set questions to be asked and clear instructions for those running the evaluation.

4. Ensure prototype is ready to support the tasks

5. Ensure recording facilities are available and functioning

6. Conduct each session, by observing the users as they work through the tasks and recording what they say, and by exploring their impressions and intentions through relevant questions.

7. Make notes of unexpected events and user comments but keep up a dialogue so that the user is encouraged to explain their actions and expectations of the system

8. Conduct post-session interviews with the users, drawing upon pre-set questions and issues raised during the evaluation

9. Debrief the users and thank them for their co-operation

10. Analyse information obtained, summarise unexpected behaviour and user comments. Consider the themes and severity of the problems identified.

11. Summarise design implications and recommendations for improvements and feed back to system/process design team. The tape recordings can be used to demonstrate particular themes.

The primary deliverables from this method consist of a summary of the usersÕ comments and evaluations and their observed experiences whilst using the system. In addition the method results in a report which lists the identified problems with associated severity ratings and makes recommendations for improvements to the product.

Parallel Design

Primary Reference Source

Nielsen, J (1993) *Usability Engineering.* Academic Press, Boston.

Summary description

It is often helpful to develop possible system concepts with a parallel process in which several different designers work out possible designs. The aim is to develop and evaluate different system ideas before settling on a single approach as a basis for the system.

In parallel design it is important to have the designers working independently, since the goal is to generate as much diversity as possible. Therefore the designers should not discuss their designs with each other until after they have produced their draft design concepts.

When designers have completed their designs, it is likely that that they will have approached the problem in radically different ways that will give rise to different user systems. It is then possible to combine designs and taking the best features from each.

It is important to employ parallel design for novel systems where they is no established guidelines for how best the system should operate.

Although parallel design might at first seem like an expensive approach, since many ideas are generated without implementing them, it is in fact a very cheap way of exploring the range of possible system concepts.

Typical Application Areas

Parallel design is useful when there are quite different system concepts under discussion and it is required to make a decision on one and one only. It of course implies that some kind of working model or prototype is produced by all the independent groups – see for instance Paper Prototyping or Wizard of Oz.

Benefits

¥ Allows a range of ideas to be generated quickly and cost effectively.

¥ Parallel nature of the approach allows several approaches to be explored at the same time, thus compressing the concept development schedule.

¥ The concepts generated can often be combined so that the final system benefits from all ideas proposed.

¥ Only minimal resources and materials are required to convey product feel.

¥ The technique can be utilised by those with little or no human factors expertise.

Limitations

Requires a number of design team members to be available at the same time to produce system concepts. May become expensive if it becomes too elaborate.

Cost of use

The costs are mainly in time over the short period that design work is being carried out. Time is also needed to compare parallel design outputs properly so that the benefits of each approach are obtained.

Suitability for requirements engineering in Telematics

Partner NOMOS reports some successes using this method; it should only be started if there is a shared understanding in the project of how much resources will be put into the activity by each partner/ competing solution provider.

How to get it

The best way to start is by reading Nielsen's book. This approach is basically a way of combining different prototyping approaches with some comparative evaluation between candidate systems.

Procedure

The following procedure may be adopted for implementing this method:

1. Define clearly the boundaries for the parallel design i.e. goal of system, tasks that it should support, user characteristics etc.

2. If possible agree on the format that the design will be produced in e.g. on paper, in software.

3. If design teams rather than individuals are being used, select groups that have roughly equivalent skills.

4. Set a clear time limit on the design phase.

5. Agree on the criteria by which the designÕs will be assessed.

6. Allow sufficient time to carry out a fair comparison of the designs produced.

7. Discuss each design separately and then discuss how different aspects of the designs may be combined.

Walkthrough

Primary Reference Source

J Nielsen and RL Mack(1994) *Usability Inspection Methods.* J Wiley, NY.

Summary Description

A walkthrough is a process of going step by step through a system design getting reactions from relevant staff, typically users or experts role-playing the part of users. Normally one or two members of the design team will guide the walkthrough, while one or more users will comment as the walkthrough proceeds.

Typical Application Areas

This technique is most often used where there is a relatively unstable prototype or a written procedural specification. The ubiquitous presence of design team members usually inhibits free-floating comments from all but the most outspoken of users.

Benefits

¥ Obtains reactions to a design in an informal manner.

¥ Flexible means obtaining reactions, allowing the discussion to range over issues not originally considered.

Limitations

¥ Requires some form of prototype to show for user to react to.

¥ Results are opinions rather than objective data.

¥ Users may tend to react positively on seeing some prototype in operation.

¥ It may be difficult to imagine how the system will operate in the real environment.

Cost of use

Requires a prototype to be developed. The time overhead in holding the walkthrough sessions largely depends upon the task domain and the number of users exposed to the prototype.

Suitability for requirements engineering in Telematics

The tendency of the design team to ‘protect’ their project and for users to accept verbal reassurances makes this a less than satisfactory method to use with end users; with experienced HCI personnel role-playing or simulating end user reactions, walkthroughs may be of use if they are carried out early enough in the development.

Procedure

The general procedure for implementing this method is outlined in the following.

1. Decide clearly what issues or task scenarios should be covered by the walkthrough.

2. Set up a good recording mechanism, e.g. one person to show the system and ask questions, another to take notes, or record on tape, peoples comments for transcription later on.

5. Select appropriate users to take part in the walkthrough, trying to cover the range of users within the target population or request some HCI experts not connected with the project to assist.

7. Pilot the walkthrough to work out how much time is needed for each session.

8. Ensure recording facilities are available and functioning.

9. Conduct the walkthrough sessions, making sure that all sessions cover the issues identified beforehand.

12. Analyse information obtained by issue and comments made. Try to determine how many users made the same comment. Consider the themes and severity of the problems identified.

Appendix: Use of different user-based approaches by special needs groups

Approaches	Disabled	Young	Elderly
Group Discussion	1	2	1
Usability Context Analysis	1	1	1
Brainstorming	2	3	1
Naturalistic Observation	1	1	1
Surveys	3	3	2
Ethnographic Approach / Contextual Inquiry	3	3	3
Focus Groups	1	3	1
Interviews	1	2	1
Functionality Matrix	1	2	2
Task Allocation Charts	2	3	2
Task Analysis	1	2	1
Wizard of Oz Prototyping	2	1	2
Paper Prototyping	2	1	1
Scenario Building	1	2	2
Storyboarding / Presentation Scenarios	2	1	1
Empathic modelling	1	2	1
Video Prototyping	1	1	1
Diary Methods	2	3	2
Rapid Prototyping & RAD/JAD	1	1	1
Laboratory-based Observation	1	2	1
Co-operative Evaluation	1	2	1
Parallel Design	1	1	1
Walkthrough	1	3	2

Key:		
	1	Easily used with the projected user group
	2	May be used with some special adaptation
	3	May be used with a lot of special adaptation