RESPECT
User-Centred Requirements Handbook

16th July 1998

Version 3.3

Previously called:
"RESPECT User Requirements Framework Handbook"
(Version 2.21)

Telematics Applications Project TE 2010

Requirements Engineering and Specification in Telematics

WP5
Deliverable D5.3

User-Centred Requirements Handbook

Martin C. Maguire
HUSAT Research Institute

incorporating material from
D3.1 RESPECT Methods
by J. Kirakowski and N. Vereker

Version 3.3
16th July 1998

Abstract

This document describes the RESPECT framework for user requirements specification. The process is divided into a number of stages for gathering user requirements and developing the system concept. A range of data gathering methods are described to support the user requirements capture process.

Keywords: user-centred requirements, user-centred design, requirements engineering, usability

CONTACTS

This guide was produced by the RESPECT project. The production and distribution of this guide is supported by the Information Engineering area of the EC Telematics Applications program. Additional copies can be obtained from the project co-ordinator, or from one of the other Usability Support Centres listed below.

Serco Usability Services
(Project Co-ordinator)
4 Sandy Lane, Teddington
TW11 ODU, Middx, UK
Tel: + 44 181 614 3811
Fax: + 44 181 614 3765
Nigel Bevan
nbevan@usability.serco.com
respect@npl.co.uk

HUSAT Research Institute
Martin Maguire
Tel: + 44 1509 611088
m.c.maguire@Lboro.ac.uk

Human Factors Research Group
Jurek Kirakowski
Tel: + 353 21 902636
hfrg@ucc.ie

SINTEF Telecom and Informatics
Jan Havard Skjetne
Tel: + 47 2206 7927
JanHavard.Skjetne@informatics.sintef.no

Fraunhofer-IAO
Paulus Vossen
Tel: + 49 711 970 2315
paulus.vossen@iao.fhg.de

Lloyd's Register
Jonathan Earthy
Tel: + 44 181 681 4040
jonathan.earthy@lr.org

Nomos Management AB
Nigel Claridge
Tel: + 46 8 7536220
nigelc@nomos.se

SiESA - Sistemas Expertos S.A.
Belén Martínez
Tel: +34 1 859 98 44 / 60
sysnet@bitmailer.net

CB&J
Stefana Broadbent
Tel: + 33 1 45 44 57 15
s.broadbent@cbj.fr

SIEM
Maria Athousaki
Tel: +30 1 9249668
athous@siem.mob.forthnet.gr

http://www.npl.co.uk/respect

Executive Summary

This handbook is concerned with user-centred requirements specification. It has been produced as part of the Telematics Applications Programme RESPECT project (TE 2010). Its aim is to provide a formal basis for gathering user requirements equivalent to the specification of business requirements and technical requirements. The outcome is a documented set of user requirements and preliminary system design to meet those requirements. This version has been refined after its use by Telematics Applications projects and other organisations.

The handbook has a number of key characteristics:

- It starts from the point where system design goals are specified. It takes these goals and develops requirements from the end-user's point of view. Therefore it may be used alongside a process of analysing and developing the requirements of the Business (or organisation) and the Technical requirements.

- It is based upon an iterative process, drawn from the ISO 13407 draft standard for user-centred design. It comprises a cycle of three iterations: (i) User context and early design, (ii) Prototyping and user test, (iii) User requirements documentation. The outcome is a documented set of user requirements for the future new system or revised system.

- It is supported by a set of data collection methods and techniques for establishing user requirements, which are also described within this document.

- It can be used flexibly to fit in with different system design methods.

To use the handbook successfully, it is vital to work through the successive stages of Part B in a careful and comprehensive planned procedure.

Feedback on this document

If you have any comments on this document, please pass them to:

Martin Maguire
HUSAT Research Institute
The Elms, Elms Grove
Loughborough, LE11 1RG, Leics
UK
Tel: +44 1509 611088, Fax: +44 1509 234651
m.c.maguire@lboro.ac.uk

Feedback Form

RESPECT User-Centred Requirements Handbook Version 3.3

Please use this form to provide feedback on this document.

Completed by:
Date:

1) What application did you use the handbook for or what application areas are you most interested in?

2) How understandable do you find this document in general?

3) How well does it relate to the project you are working on, or to your organisation's specification activities in general?

4) How well do you feel that you could carry out the user requirements capture and specification process described in **Part B, Phases 1 to 3** ?

5) Are there any improvements that you would like to see made to **Part B**?

6) Are there any other improvements you would like to see made to the document as a whole?

7) Do you have any other comments about the document?

Please send, fax or email your comments to:

Martin Maguire, HUSAT Research Institute, The Elms, Elms Grove, Loughborough, LE11 1RG, Leics, UK
Tel: +44 1509 611088, Fax: +44 1509 234651, Email: m.c.maguire@Lboro.ac.uk

Audience for this document

This handbook is intended for use by project team members with responsibility for generating and maintaining user requirements.

It offers an overview of user requirements capture for project managers and provides background material for user representatives and technical designers. Staff concerned with requirements specification can use this document to guide them through the process. If they already have a procedure that they intend to follow, they may simply use the document to assist with specific activities such as running a discussion group or conducting interviews.

When using this document, requirements gathering personnel should be able to refer to someone with human factors, ergonomics or psychology skills, to ensure that the methods recommended in the document are being applied appropriately. Such a specialist would be able to identify potential problems with, for example, a survey form before it is administered, or an interview programme before costly interview time is used.

The extent of the work and reading needed to undertake user requirements specification is not as great as may first appear from the size of the document. **Part A** gives the background to the handbook, **Part C** is a reference source for guidance on a range of relevant methods, and **Part D** contains the references and blank master copies of the recommended forms. **Part B** is a guided process or framework for user-centred requirements and design, and is the central core.

To exploit the framework, design teams must work through the three successive stages of **Part B** in a steady and disciplined procedure. By careful and comprehensive use of the framework, you will certainly develop user requirements specifications and set usability goals of much greater accuracy and validity than are typical hitherto.

Acknowledgements

As with any integrative and prescriptive handbook in the modern world, this RESPECT User-Centred Requirements Handbook draws upon and owes much to the work of many others.

The Author therefore wishes to give full acknowledgement to significant contributors as follows:

At the *HUSAT Research Institute*, important contributions have been made by other members of the RESPECT team:

Professor Brian Shackel
Robert Graham
Colette Nicolle.

The authors of the RESPECT deliverable D3.1 'Methods for User-Oriented Requirements Specification', which Part C of this document draws from:

Dr Jurek Kirakowski, *Human Factors Research Group, University College Cork*
Keith Hurley
Natalie Vereker

The several RESPECT colleagues and other reviewers who have made valuable comments and inputs to previous drafts:-

Dr Nigel Bevan (RESPECT Project Co-ordinator) and
Owen Daly-Jones,*Usability Services, National Physical Laboratory, UK.*
Jan Heim, Tor Endestad, Jan Havard Sketjne, *SINTEF Unimed Rehab, Norway.*
Paulus Vossen, *Fraunhofer Institute, Stuttgart.*
Nigel Claridge, *Nomos Management AB.*
Sara Jones, *University of Hertfordshire.*
Michael J. Underwood.

The HUSAT developers of the Planning, Analysis and Specification (PAS) Toolset under the Esprit HUFIT Project, from which the basic approach of this Framework is drawn:

Margaret Flite-Galer
Bernard Catterall
Bronwen Taylor
Martin Maguire
Gordon Allison

The earlier work of other HUSAT experts including:

Leela Damodaran, Ken Eason, David Davies, Susan Harker, Wendy Olphert, Arthur Gardner, Jim McKenzie and Brian Shackel.

The developers of the Usability Context Analysis Handbook under the Esprit MUSiC project:

Dr Nigel Bevan, Rosemary Bowden, Richard Corcoran, Ian Curson, Mile Macleod, Jonathan Maissel, R. Rengger and Cathy Thomas, *National Physical Laboratory*
and Dr Andrew Dillon, Martin Maguire and Marian Sweeney, *HUSAT Research Institute.*

User-Centred Requirements Handbook

Contents

List of Tables

List of Figures

Part A

Introduction to the Handbook

User-Centred Design

User-centred design is an approach to interactive system development which focuses specifically on making systems usable and safe for their users. User-centred systems empower users and motivate them to learn and explore new system solutions. The benefits include increased productivity, enhanced quality of work, reductions in support and training costs and improved user health and safety. Although there is a substantial body of human factors and ergonomics knowledge about how such design processes can be organised and used effectively, much of this information is not yet widely applied.

Adopting a user-centred design process leads to more usable systems and products. It reduces the risk that the resulting system will under-deliver or fail.

User-centred design implies:

- early focus on users, tasks and environment;
- the active involvement of users;
- an appropriate allocation of function between user and system;
- the incorporation of user-derived feedback into system design;
- iterative design whereby a prototype is designed, tested and modified;

The process of user requirements specification described in this document is based upon these principles. In particular it is an iterative process based upon the design cycle presented in the user-centred design draft standard ISO 13407 (1997b) shown below:

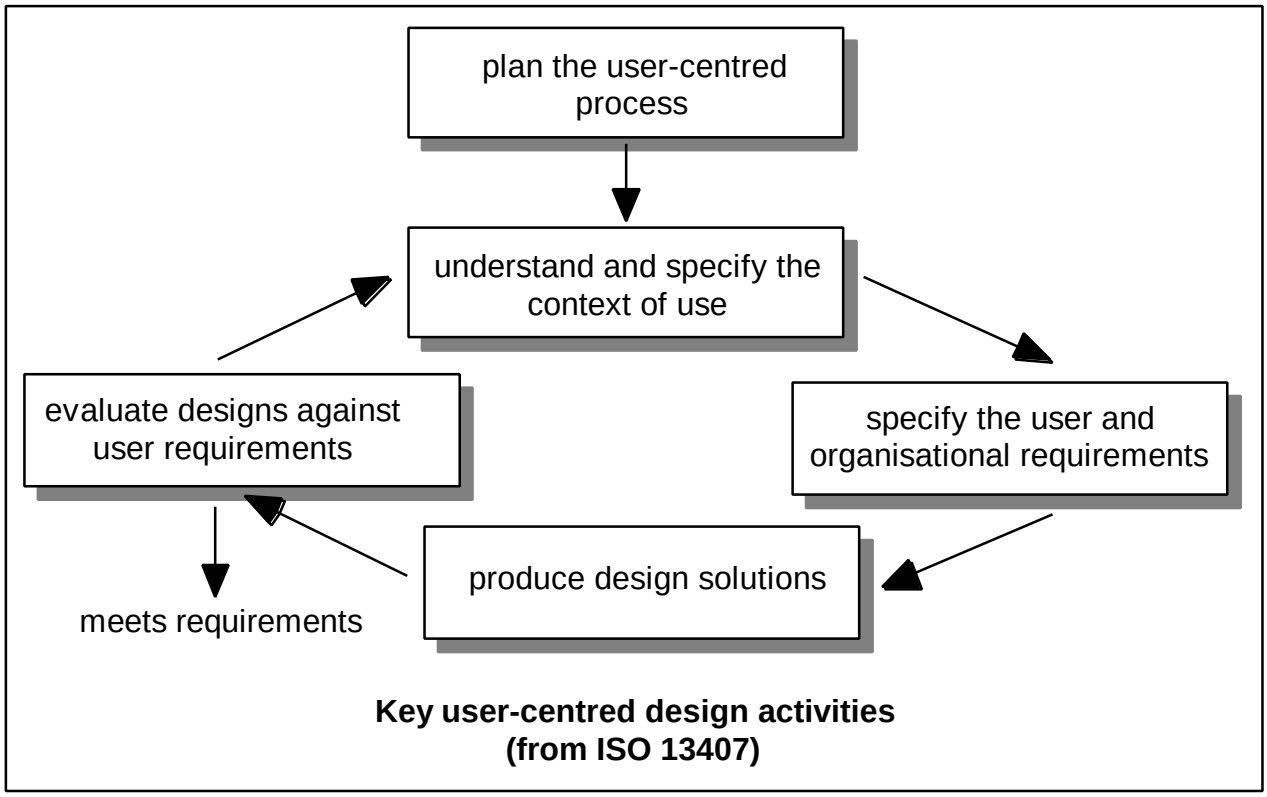

FIGURE 1. THE USER-CENTRED DESIGN CYCLE

The main cost of achieving the benefits of user-centred design is that the manager's task initially appears more difficult. Project planning has to allow for iteration and for incorporating user feedback. More time will also be required for effective communication between design team participants and for reconciling potential conflicts and trade-offs. However, project managers will benefit from the additional creativity and ideas from an extended development team and skill base. Users will also feel a strong sense of ownership of the system that results. Above all, proper consideration of usage issues early on in the project will result in a better design and significant savings at later stages when changes are much more costly.

Requirements analysis is the first stage in the user-centred design process. Later stages are described in the INUSE Handbook of User Centred Design (Daly-Jones, Thomas and Bevan, 1997). The Handbook also describes the principles of user-centred design, based on the ISO 13407 standard (ISO, 1997b).

Focus of the Handbook

Requirements analysis is the process of determining what is required of a future system or product. This may be a computer-based system for a particular customer or a product to be launched onto the open market. This document uses the term **'system'** to cover all classes of application including large scale computer-based systems, software packages and standalone electronic products.

Typically the process starts with a **project proposal** or **project brief**. This describes what is wanted from the proposed system in general terms. From this starting point, three kinds of requirement are developed:

- **Business requirements** – specify the needs of the business from a commercial point of view or, for systems being developed internally, the needs of the enterprise in general. A requirement may, for example, be that the system should sell at least 10,000 units within 2 years of launch. Another may be to divert staff from routine work to problem solving activities.

- **User requirements and functional specification** – specify the system |requirements from a user's point of view, including the functions required to support the user tasks, the user-system interfaces, user support required, physical and organisational requirements, equipment and hardware. They also include usability goals that must be achieved and the approach for installing the system. In the Telematics context, 'the user' may include both end users of an electronic service, and service providers who make use of the network infrastructure.

- **Technical requirements** – specify how the system will achieve the required functions and the structure of data that must be available for internal processing to be successful. For example, if a search function is to give a fast response time, the data may need to be indexed in a certain way to support rapid retrieval. Technical constraints will also be specified, such as the maximum data communication speed over a network.

All three of these sets of requirements must be carefully developed to ensure the success of the new system. Historically they may have been seen as separate, and the importance of user requirements is often overlooked. The three types of requirements should be developed logically in the order given above, and to achieve a successful design should be carried out as part of a common business framework. This guide is concerned with the capture and specification of user requirements and user functions, rather than organisation and business requirements or technical requirements. The complementary technical requirements will be of two types: specifications developed to meet the need of the user requirements, and constraints on how the user requirements can be achieved, related to the nature of available technology.

Relationship to, and use with, Traditional Requirements Engineering methods

Traditional approaches to requirements engineering concentrate on identifying functional requirements and ensuring that the developed product meets these requirements. Other non-functional requirements (efficiency, reliability, usability, maintainability and portability), have had less importance. Yet from a user perspective, non-functional requirements may be critical to successful implementation of a new system.

RESPECT provides a broad framework for requirements engineering which makes meeting user needs to achieve quality in use the overall objective of the design process (see Bevan and Azuma 1997, Bevan 1997, Bevan 1998). Neither functionality nor usability (in the narrow sense of an easy to use interface) is given priority: they are subservient to the objective of providing a system which enables the user to meet their goals in the real world. Depending on the context of use the user's goals may be business or personal objectives.

RESPECT emphasises the importance of obtaining a complete understanding of user needs, and validating the emerging requirements against potential real world scenarios of usage. Existing requirements engineering methods and tools can be used within this framework to document and trace the detailed requirements through the development process.

RESPECT also introduces additional requirements of two types: detailed contextual requirements associated with scenarios of use, and high level quality in use goals (also called usability goals) for the users to be effective, efficient and satisfied when carrying out their tasks.

The RESPECT approach to user requirements forms part of a broader framework for user centred design, documented in the INUSE Handbook of User Centred Design. The INUSE handbook describes the methods which can be used to implement the design from a user-centred perspective.

Using RESPECT in conjunction with existing methods

If your organisation does not already have a requirements engineering process, the RESPECT process will provide a business and user-oriented discipline and framework for identifying requirements. RESPECT does not deal explicitly with how to identify and document detailed requirements, e.g. relating to technical interfaces or internal data structures. Which methods are most appropriate for dealing with detailed requirements will depend on the nature of the system being developed, and the nature of any existing requirements engineering procedures.

The RESPECT process is entirely compatible with current good practice in requirements engineering, but enhances current approaches in several important ways:

- RESPECT recommends identifying high level quality and user requirements (and associated constraints) and obtaining feedback on these prior to elaborating more detailed technical requirements.
- RESPECT puts more emphasis on understanding detailed scenarios of use in order to elicit important non-functional requirements. The scenarios developed by RESPECT also require the definition of more detailed scenarios or "use cases" than required by some requirements engineering processes.

- RESPECT methods will help prioritise requirements from a user perspective.
- RESPECT recommends additional methods and techniques to elicit and validate system requirements from a user's point of view, including the functions required to support the user tasks, the user-system interfaces, user support required, physical and organisational requirements, equipment and hardware, usability goals that must be achieved and the approach to installing the system.

The RESPECT method does not include organisational and business requirements such as the needs of the business from a commercial point of view or, for systems being developed internally, the needs of the enterprise in general. Methods for identifying organisational requirements are described in Contextual Design: A Customer-Centered Approach to Systems Designs by Hugh Beyer and Karen Holtzblatt (1998).

Requirements Engineering – A Good Practice Guide

A good practical introduction to conventional requirements engineering is given in the book: Requirements Engineering – A Good Practice Guide by Sommerville and Sawyer (1997). This is used of an example of how to integrate RESPECT with conventional approaches.

The following guidelines in the book are compatible with and are directly support the RESPECT approach:

1. Requirements elicitation

- Assess system feasibility
- Be sensitive to organisational and political considerations
- Identify and consult system stakeholders
- Record requirements sources
- Define the system's operating environment
- Use business concerns to drive requirements elicitation
- Look for domain constraints
- Record requirements rationale
- Collect requirements from multiple viewpoints
- Prototype poorly understood requirements
- Use scenarios to elicit requirements

2. Requirements validation

- Use prototyping to animate requirements

3. Requirements analysis and negotiation

The RESPECT procedure provides a richer process for analysis and negotiation of requirements than recommended in the Good Practice Guide. RESPECT uses a form of checklists, and can be seen as an elaboration of the simpler spiral approach to elicitation, analysis and negotiation recommended in the book. The guidelines in the book include:

- Define systems boundaries
- Use checklists for requirements analysis
- Provide software to support negotiations
- Plan for conflicts and conflict resolution

- Prioritise requirements

Other aspects of requirements engineering

The book contains additional guidelines to support the broader requirements engineering process, which is not specifically covered in the RESPECT procedure. These guidelines are useful when planning or assessing the overall requirements engineering process. They include:

1. Requirements document

- Define a standard document structure
- Explain how to use the document
- Include a summary of the requirements
- Make a business case for the system
- Define specialised terms
- Lay out the document for readability
- Help readers find information
- Make the document easy to change

2. Describing requirements

- Define standard templates for describing requirements
- Use language simply and concisely
- Use diagrams appropriately
- Supplement natural language with other descriptions of requirements

3. System modelling

- Develop complementary system models
- Model the system's environment
- Model the system architecture

4. Requirements validation

- Organise formal requirements inspections
- Use multi-disciplinary teams to review requirements
- Define validation checklists
- Write a draft user manual
- Propose requirements test cases

5. Requirements management

- Uniquely identify each requirement
- Define policies for requirements management
- Define traceability policies
- Maintain a traceability manual

The Good Practice Guide book briefly describes how to implement each guideline, and gives the benefits, costs and problems. It can thus be used in conjunction with the RESPECT document to provide a comprehensive approach to requirements engineering.

Existing software engineering procedures

Where a system is being developed for a small company or is for a few internal users at a large firm, the RESPECT method (described in Part B) can be used as a free standing method. However for larger projects, the method will need to work within the structure of an established software engineering procedure.

Most large software projects are developed using some version of the "Waterfall method". This assumes that a system is developed through a clearly defined series of steps or "Phases".

The steps are typically:

- Requirements analysis
- Specification
- Planning
- Design
- Implementation
- Integration
- Maintenance

In its strictest version, the traditional approach states that each phase must be completed before the next phase can begin, and that the development team should not need to return to an earlier phase to redefine a system as it is being developed. However most software engineering specialists today realise that this approach is unrealistic and that for interactive system development, iterations in the process are needed.

Various modifications to the phases of the waterfall method and their interaction have been proposed. However software development environments still incorporate many steps of the method, partly for historical reasons and partly because the approach helps to define responsibilities and costs for various activities within a large software project. Thus the RESPECT method can supplement some of the stages of a waterfall environment.

Requirements Analysis Stage

The Waterfall method's initial "Requirements Analysis" phase describes the activity of defining the precise needs that the software must meet. These needs should be defined in terms of the users' needs, rather than how they will be met by the proposed system. However traditional software engineering tends to take an idealistic view. It essentially looks at requirements in an abstract way rather than contextualised as part of tasks to perform. The RESPECT process and similar methodologies, such as that by Beyer and Holtzblatt (1997), is centred upon the system end-user, the tasks they have to perform with the system, and the environment in which they work or operate.

The traditional requirements engineering approach defines what the user needs to do, not how it will be done. However the RESPECT process goes further and develops specifications of how the needs will be satisfied. It defines, for example, interactive procedures and organisational designs so that users become aware and are happy with the future implementation of the system. Normally needs analysis is like stamp collecting where ideas and wish list items are collected together. The RESPECT process is more comprehensive and structured, and provides a more complete view. Traditionally requirements are specified at different levels and may be unrelated or simply low level enhancements to the existing system. The RESPECT approach is based on representative tasks which are complete descriptions of what the user needs to do to achieve particular goals.

Thus an example task for a bank machine might be to "withdraw money". Thus the requirement would be for the user to be able to identify themselves, specify the amount they require, and to obtain the money. A more complete task description will perhaps describe the user, task and the possible environment at the time. For example: "a user in a wheelchair, wishes to withdraw £50 from the bank machine at night. However their current limit only allows £20 to be withdrawn". By considering all these aspects, the user requirements can be made more complete and take into account possible problems that may arise for the user.

A traditional requirements analysis will collect detailed partial tasks and functions that are needed to support those tasks. Corresponding to this within the RESPECT process, there are two phases: **Phase 1- User context and early design** which analyses current task processes, and **Phase 2 - Prototype and user test** which develops new processes and identifies new functions. Thus the RESPECT process and the traditional approach can complement each other. The traditional approach helps to ensure that all important functions of the system are recognised, while the representative task descriptions in the RESPECT process provide an integrated picture of the functions working together.

Specification

In the traditional "Specifications" phase of software engineering, the requirements are used to produce a description of the system that includes the details needed by the software designers and implementers. The customers or end-users can then sign off on this document, and the software team can begin to plan and design the actual system. However as users will typically not be experts at reading specification documents, they may have trouble imagining how the system will actually perform. Various alternatives to written specifications have been proposed, including prototypes and a more iterative approach to design, both of which form part of the RESPECT Framework. Thus a prototype of the system can be added to the requirements specification to illustrate how the new system will operate.

However the RESPECT Method also uses task scenarios (called 'usage cases') and documents how the user will interact with the new system in order to achieve the task goal. Thus customers will be able to understand this part of the specification. It will also force the specification writer to consider a complete task, which may catch problems that could otherwise be missed when functions are considered individually. Within **Phase 3 - User requirements documentation**, the RESPECT process will produce a user-oriented list of user requirements structured under different headings (e.g. functions and features, user support, and environmental requirements). These will provide a useful guide to the users to enable them to see what the future system will be like, and whether they wish to accept it or request changes.

Planning, Design and later stages

From this point on, the strict Waterfall method and the RESPECT method may take different paths, although both work in parallel. As the design and implementation progresses, it is likely that changes will be needed to make sure that the system meets user needs. Based on the user-centred design principle of iterative design, several iterations of testing and redesign are likely to be required, which may involve jumping back through the phases of the waterfall method. At the same time, changes should also be made to RESPECT **Phase 3 - User requirements documentation**. This will allow users to refer to the current status of the requirements expressed in a form they will be able to understand. This will also allow them to maintain a good grasp of the development process as the system develops.

There are also parts of the RESPECT process that will be applied during the later stages of the Waterfall process. Firstly a test plan is described which will include usability test goals. By referring to this plan, users may take a more active part within the test process. They will thus be able to see at first hand whether the system is able to meet its usability goals. Also the process describes the requirements for user support and implementation. These will be specified to ease the process of user awareness and uptake of the new system. Again users will be able to refer to these descriptions in order to ensure that user support and implementation needs are being addressed properly.

Relationship with other RESPECT documents

The RESPECT project has produced a number of interrelated documents. This document, D5.3, forms the RESPECT Framework for analysing and specifying user requirements. The figure overleaf shows the relationship of this document to other RESPECT documents.

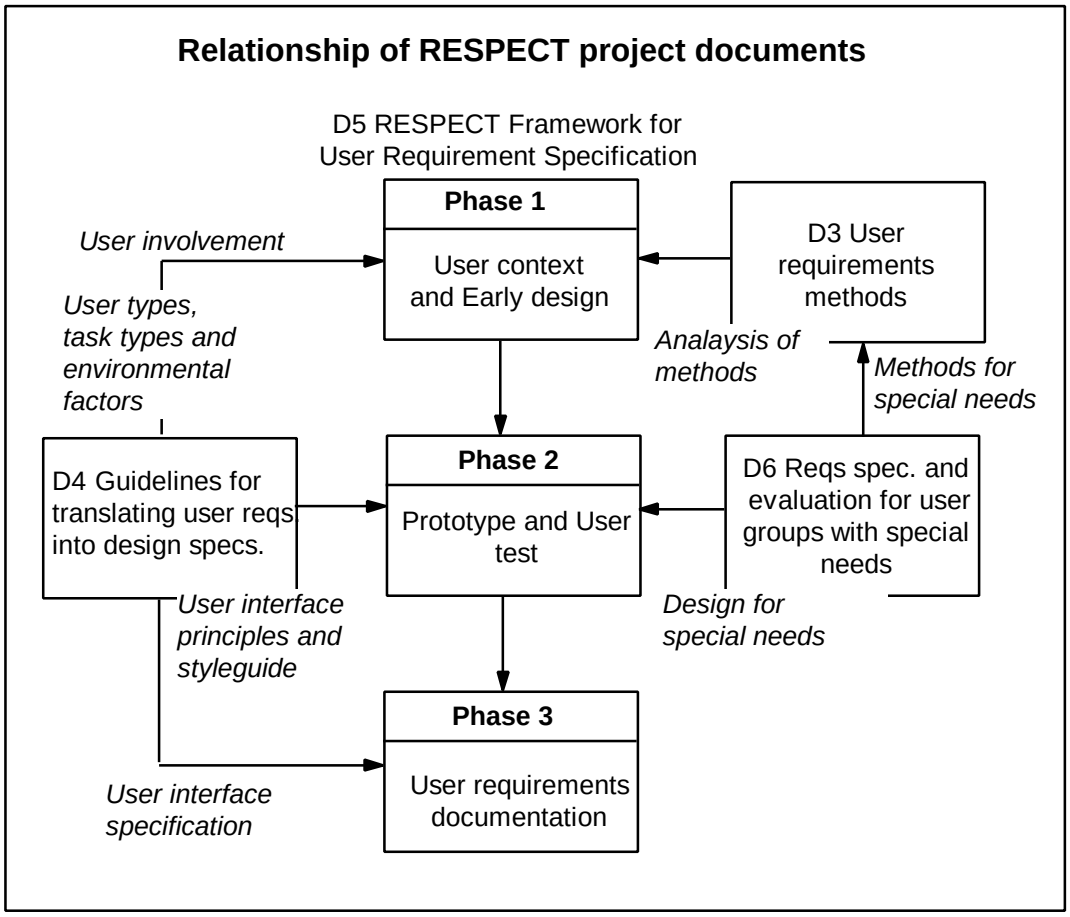

FIGURE 2. RELATIONSHIP BETWEEN RESPECT DOCUMENTS

Additional RESPECT documents supporting the user centred requirements engineering process are as follows:

D3.2 Methods for user-oriented requirements specification
This state of the art report on methods for user centred requirements elicitation and specification catalogues user-based requirements elicitation methods which have been shown to be of industrial relevance, and which can be adapted to the needs of the development of telematics applications.

D4.2 Guide to mapping requirements to user interface specifications
This guide (Vossen and Maguire, 1998) assists in the process of translating the requirements generated by this Requirements Handbook into user interface specifications, taking account of different aspects of user needs. It includes a list of factors and possible requirements related to users, tasks and environments which should influence the user interface design. It also provides advice on user involvement in the design process.

D6.2 Requirements specification and evaluation for users groups with special needs
This document (Heim et al, 1997) gives information on some of the basic requirements of young, old and disabled people and explains in detail how to design systems to take account of their needs. It also includes advice on carrying out methods for capturing user requirements and evaluating prototypes when applied to users with special needs.

Reference is made to these other documents at appropriate points in the requirements specification process (Part B of this document), indicated by the symbol:  .

Main stages of the Framework

Understanding user requirements is an integral part of information systems design and is critical to the success of projects such as those within the Telematics Applications Programme. However specifying these requirements is not so simple to achieve. How can a system be designed before the future situation is known? End-users may not appreciate the benefits that future technology can offer them. Once they understand the implications of a potential solution, their requirements may change. Similarly, the design team may find it difficult to integrate user opinions into the design process, and thus may concentrate only on the technical requirements of the system.

An important characteristic of the user requirements process is that user opinions of what they might want from a system will evolve. As a system concept develops, users will see new possibilities or potential problems and so the requirements will change.

A general approach to specifying user requirements needs to be flexible to meet different situations e.g. generic or custom system development, new systems or developments of existing systems. With these differences in mind, this framework document has been developed to offer a general structure from which relevant techniques can be selected and used as applicable.

*This document offers an approach to specifying user requirements that is **data driven**. This means that the basis of the approach is to collect and generate user requirements under a series of section headings A range of different methods are suggested to capture the required data. Once captured, these data form the basis of the user requirements specification document.*

The three main stages of the user requirements framework and their component sections can be represented as an iterative cycle, as shown in the figure overleaf. Each cycle contains an analysis of the context of use, the specification of user requirements, developing a design to meet those requirements and testing them against the requirements.

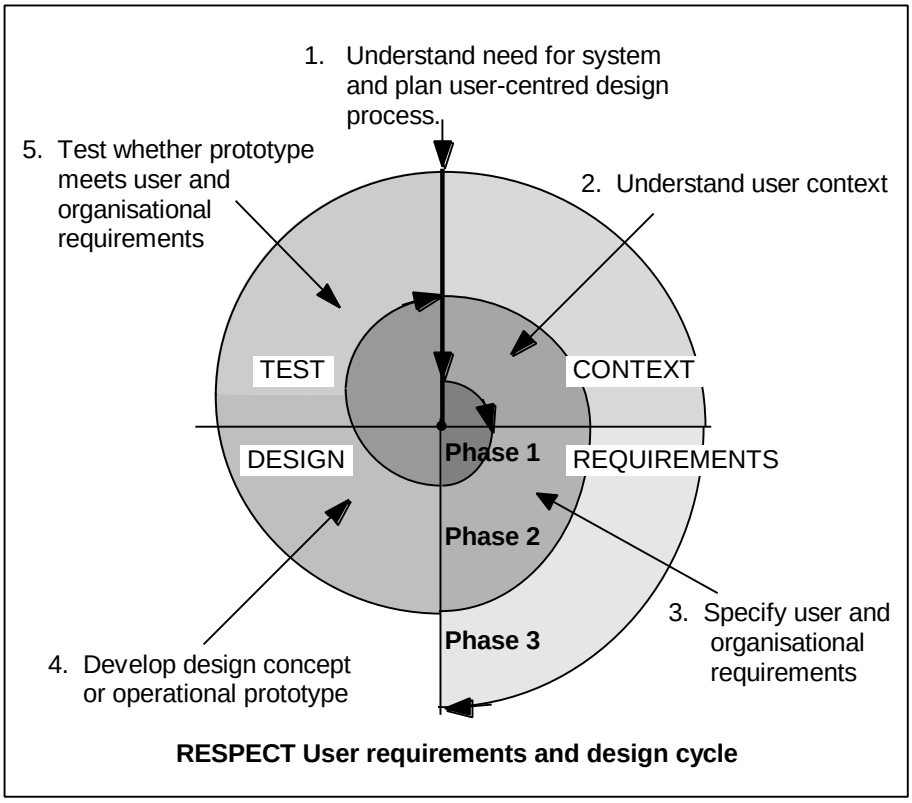

FIGURE 3. OVERVIEW OF RESPECT REQUIREMENTS AND DESIGN CYCLE

The main stages of this process are described below:

PHASE 1. USER CONTEXT AND EARLY DESIGN

1.1 Summarise project

This consists of recording details the initial project requirement and design context.

1.2 Identify users and stakeholders

Here the future users and stakeholders of the system are identified, and their roles in relation to the system are recorded.

1.3 Specify user characteristics

Here user characteristics such as skills, physical attributes and personal details are recorded.

1.4 Describe technical environment

Here the technical characteristics of the current system (e.g. displays, keyboards, telephones) and possible future requirements are recorded.

1.5 Describe physical environment

The characteristics of the physical environment (e.g. lighting, sound and thermal conditions) and possible future requirements are recorded.

1.6 Describe social and organisational environment

The characteristics of the social environment (e.g. organisational aims, staff and management structure, performance monitoring, group working) and possible future requirements are recorded.

1.7 Identify user goals and tasks

User goals for the system are listed. Where there is a current system in place, user tasks to achieve each goal are identified.

1.8 Review current processes

User goals for the system are listed. Where there is a current system in place, user tasks to achieve each goal are identified.

1.9 Review similar systems and products

Features of similar (possibly competitive) systems, which may need to be included or definitely excluded, are documented.

1.10 Produce design ideas and concepts

Design ideas and concepts for the new system are generated and represented in some way.

1.11 Perform expert review of designs

Expert reviews are carried out of the design ideas and concepts to assess their feasibility for use in the new system.

1.12 Move to Phase 2?

An assessment is made as to whether the design ideas and concepts form a sufficiently good basis for further development as a prototype. If so, then the process continues with phase 2.

PHASE 2. PROTOTYPE AND USER TEST

2.1 General usability goals and guidelines

To start the design process it is helpful to summarise general usability goals that the design is aiming to achieve. Also a checklist of guidelines may be considered to assist the process of user interface design.

2.2 Identify design constraints

Here constraints on the design process are listed.

2.3 Identify task scenarios

Here a number of task scenarios which represent common or important task situations are listed. These are used to test the success of the prototype. Usability criteria are also established to help judge the success of the system in relation to the tasks. At least one scenario is required for each user goal.

2.4 Propose new processes

For each scenario, a set of interaction steps are listed to demonstrate how the system should be used. A list of functions and features which support these steps are listed.

2.5 Develop prototype
An interactive prototype is developed using software and/or hardware which users can interact with. This will then be used to test the system design.

2.6 Test prototype with users
The prototype is tested with users and observations recorded by evaluators. Also performance scores and subjective ratings are recorded which may be used as a basis for a user requirements test plan.

2.7 Review user cost/benefits
The tasks that the each user group will have to perform are reviewed to see how acceptable they are and components of a job.

2.8 Move to Phase 3?
The results of the prototype test and the task acceptability review are considered. If the prototype appears to be successful, the design can be used as a basis for the user requirements specification which is carried out in Phase 3.

PHASE 3. USER REQUIREMENTS DOCUMENTATION

In this Phase, the user requirements identified within Phases 1 and 2, are documented. A process of prioritisation is also carried out to ensure that the most important requirements are achieved. Progress on the achievement of each requirement is also monitored during the design process.

3.1 General system characteristics
Here any general characteristics of the Design Concept are listed.

3.2 Organisational structure
This section describes the intended organisational structure implied by the new system.

3.3 Task scenarios and interaction steps
This section summaries the user goals and system related tasks that they will perform as part of the new system.

3.4 Technical environment
This section describes the user requirements for the future technical environment e.g. the software, hardware, other equipment, and other materials needed to perform the user tasks.

3.5 System functions and features
The main functions and features of the system are listed together with the user group each relates to.

3.6 User interface design
This section describes the characteristics of the proposed user interface illustrating the structure and example screen layouts. A reference is also made to the system prototype and how to run it.

3.7 User support

This section describes future user support requirements including on-line help, documentation, human support etc.

3.8 Physical environment

This section describes the user requirements for the future physical environment e.g. the lighting, sound, workstation layout etc.

3.9 Social and organisational environment

This section describes the user requirements for the future social environment e.g. group working, assistance provided etc.

3.10 Standards and styleguides to apply

Here all relevant standards and interface styleguides that should be referred to during the design process are listed.

3.11 Test plan

A plan for testing the system when it is implemented is developed. This is based on the previously identified task scenarios.

3.12 Implementation plan

A description of the system phasing is required to show the migration path to the new system from the user's points of view.

Part B

User Requirements Framework

The User Requirements Framework consists of three main **phases**. Each stage is broken down into a series of numbered **sections,** as shown below. The aim of the Method is to collect data (e.g. user characteristics) and to generate items (e.g. a usage case) for each section. The phases and sections within them are defined so that one naturally leads on to the next. However the process is flexible and the order in which each section is considered can vary as appropriate to the situation.

Phase 1. User context and Early design

1.1 Summarise project
1.2 Identify users and stakeholders
1.3 Specify user characteristics
1.4 Describe technical environment
1.5 Describe physical environment
1.6 Describe social and organisational environment
1.7 Identify user goals and tasks
1.8 Review current processes
1.9 Review similar systems and products
1.10 Produce design ideas and concepts
1.11 Perform expert review of designs
1.12 Move to Phase 2?

Phase 2. Prototype and User test

2.1 General usability goals and guidelines
2.2 Identify design constraints
2.3 Identify task scenarios
2.4 Propose new processes
2.5 Develop prototype
2.6 Test prototype with users
2.7 Review user cost/benefits
2.8 Move to Phase 3?

Phase 3. User Requirements Documentation

3.1 General system characteristics
3.2 Organisational structure
3.3 Task scenarios and interaction steps
3.4 Technical environment
3.5 System functions and features
3.6 User interface design
3.7 User support
3.8 Physical environment
3.9 Social and organisational environment
3.10 Standards and styleguides to apply
3.11 Test plan
3.12 Implementation plan

The process is iterative so that items of information are noted, then modified or firmed up. For example, a broad set of system functions may be described in the project specification, then refined after the tasks have been analysed and a prototype developed.

The method of data collection is also flexible. While in some situations requirements can be generated through discussion, in others, the use of particular techniques and methods (e.g. task analysis) may be needed.

Collecting user requirements using this framework will demonstrate that a user-driven approach to capturing user requirements has been adopted, and will support the requirements of the ISO standard 9241, Part 11 (ISO, 1997a).

When collecting requirements information, a set of forms are presented which may be used to structure the information. An example completed form is normally given with a blank version in the **Appendix 3** which may be photocopied. The forms can then be used as the basis for the user requirements document.

Phase 1

User context and Early design

PHASE 1 of the user requirements specification process consists of the first iteration around the user-centred design loop as shown below.

- **CONTEXT:** Background information is gathered about the project and what it is intending to achieve. A user and stakeholder analysis is then performed to identify the range of different users and stakeholders, and to document their characteristics. A description of the environment in which they are working is also produced. These contextual factors may highlight a need for particular user requirements.
- **REQUIREMENTS:** The next stage is to identify user goals and tasks for the system. The current process for each goal is reviewed, problems identified and ideas for overcoming the problems are listed. Similar systems or products on the market may also be reviewed to any identify functions or features that should be included or excluded from the new system.
- **DESIGN:** Using the information gathered thus far, a list of design ideas or a design concept may be produced and represented in different ways.
- **TEST:** The design ideas or concept may then be considered as part of an expert review in order to decide whether they form a good basis for meeting the user goals. If so then the process may move into Phase 2.

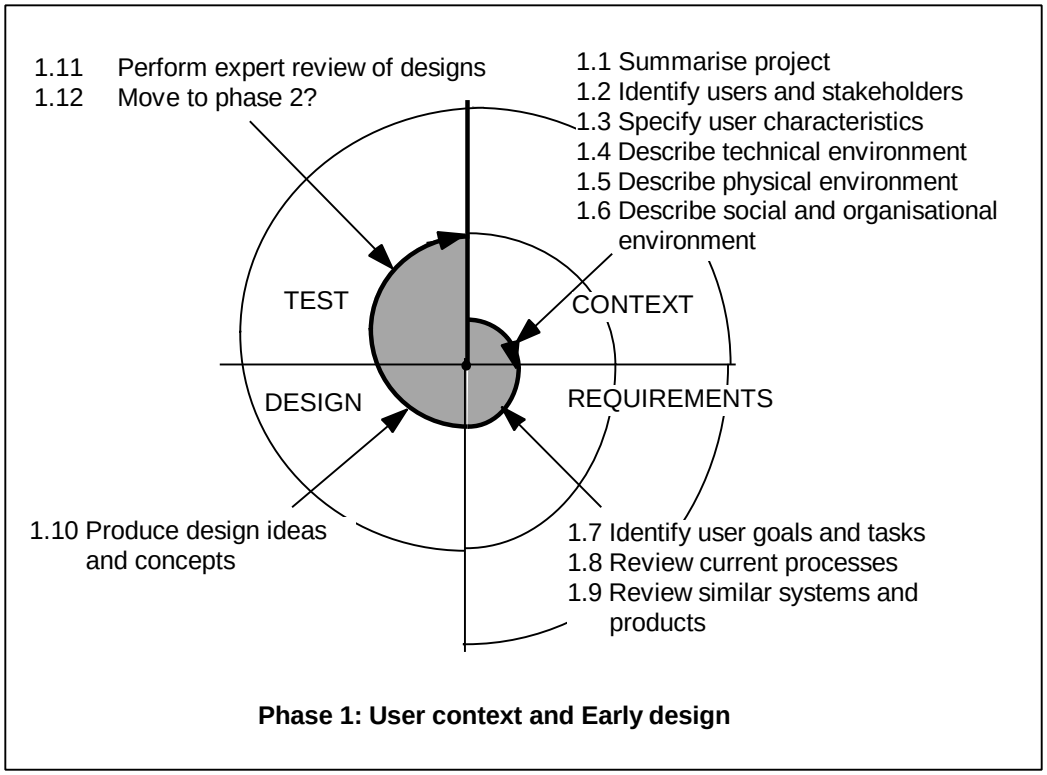

FIGURE 4. PHASE 1 - USER CONTEXT AND EARLY DESIGN

1.1 Summarise project

Objective

The development of a new or existing system will normally take place as a 'project'. It is important for the user requirements analyst to gain a high level understanding of the project and the reason for the system development taking place. It then becomes possible to understand how this will affect the user population.

The information may be obtained by interviewing the project manager. More details may be drawn from a 'system proposal document' or a document giving an initial statement of requirements, and obtaining clarification where needed.

Process

To support the creation of a project summary, **FORM 1.1** may be used.*****

*****Note 1 - Throughout Part B, the recommended Forms are shown (as below) with example data. This is a 'new ergonomic bank machine' which can be offered to banks to provide new facilities and attract new customers who traditionally are wary of using bank machines.

The parts of each form which the requirements analyst will complete are shown in italics.

*****Note 2 - Blank Master Copies of the recommended Forms throughout Part B are given in **Appendix 3**.

FORM 1.1 - PROJECT SUMMARY FROM USERS' VIEWPOINT (EXAMPLE)

1.1 Project Summary	
Questions	**Assumptions**
What is the system or service?	*New ergonomic bank machine*
What functions or services is it intended for the system to provide?	*Traditional services of cash withdrawal, statement, balance, ordering chequebook etc. Possible new services include getting change, requesting a loan, transferring money between accounts etc.*
What are the aims of the project?	• *to provide an increased range of services to bank customers via bank machines.* • *to offer a reliable service with the machines out of operation for less time* • *to offer a more secure service and safe service.*
Who is the system intended for? (Target market)	*Banks and building societies.*
Who will use the system?	*The general public, especially encouraging new users such as the elderly and disabled.*
Why is the system needed?	*To promote the wider use of bank machines in a competitive market.*
Where will the system be used?	*Outside banks and standalone in public locations such as in stations, airports, shopping malls, etc.*
How will the system be used?	*The current method will be used i.e. the user will follow instructions on screen and make inputs via a keypad. However other methods of input (speech, remote handset) or output (speech, Braille screen etc.) may be considered.*
How will the user obtain the system?	*Not applicable.*
How will the user learn to use the system?	*Via short leaflet or on-screen guidance. System should be intuitive enough not to require much learning.*
How will the system be installed?	*System pre-installed on bank machine.*
How will the system be maintained?	*Via 'state-of-health' interface. Bank staff will be prompted to carry out basic maintenance. Engineers will correct major faults.*

1.2 Identify Users and Stakeholders

Objective

This section identifies different kinds of user of the system. These include:

- **User groups** – those who use the system directly 'hands on'. They may be the *primary users* who use the system frequently, or *secondary users* who use it more infrequently such as installers, maintainers.

- **Other stakeholders** – those who influence or are affected by the system, but not the actual users. They include: recipients of information, marketing staff, purchasers, support.

For each groups of users, a short description of their intended role in the system will be recorded, or how they will use the new product.

Process

FORM 1.2, shown below can be used to record the list of user groups and a description of how they will use the system.

Bank customer
Will use the bank machine to access services.

Bank staff
Will be responsible for day-to-day maintenance, e.g. filling machine with notes and paper (for receipts and statements), correcting minor faults and reporting.

To complete the form:

1. Enter the name of the system at the top of the form.
2. Identify distinct user groups who have some interest or stake in the system. Try to be as inclusive as possible and consider all those parties who might be influenced by a system, or will have some influence on how it may be used.
3. Describe the role of each user in the system or how they will use the system.
4. Select the user groups for which user requirements will be specified. For each such group, tick the 'Expand' column.
5. For each subgroup, complete the following forms:

   Section 1.3 User characteristics
   Section 1.4 Technical environment
   Section 1.5 Physical environment
   Section 1.6 Social and organisational environment
   Section 1.7 User goals and tasks

Please refer to RESPECT D4.2, section 3.2 for an introduction on involving users within the design process. This will be of value when users and stakeholders are being identified for recruitment to the design team.

FORM 1.2 - LIST OF USERS AND CURRENT OR EXPECTED ROLE IN SYSTEM (EXAMPLE)

1.2 Users and Stakeholders		
System name: *New bank machine*		
USERS	**ROLE IN SYSTEM OR USE OF SYSTEM**	**EXPAND**
Bank customers	*Will use the bank machine to access services.*	✓
Bank staff	*Will be responsible for day-to-day maintenance, e.g. filling machine with notes and paper (for receipts and statements), correcting minor faults and reporting major faults.*	✓
Machine maintenance staff	*Will perform routine maintenance every six months and will come out to deal with major faults.*	✓
STAKEHOLDERS	**ROLE IN SYSTEM OR USE OF SYSTEM**	**EXPAND**
Bank marketing staff	*Will be concerned with deciding what services to offer on the machine and what advertising to display when the machine is not in use.*	

Each user group ticked may be described in the User context FORMS 1.3 to 1.6
All goals for each user group will be listed in FORM 1.7

General notes

The most obvious group are the end users for whom the particular system is being designed. However even this will sometimes be less than straightforward, as systems can have a range of different end users who have different objectives. It can be useful to identify subgroups of users who will have very different needs for a system. For example the transport needs of a young girl are very different from a woman accompanying her children, and the needs of the elderly and disabled traveller are also likely to be very different from those of the general population. Often assumptions are made for those who will use a system, and discussions as to the likely range of users and their attributes will be a useful design exercise.

In addition there may be indirect users who may be affected by the outcome of a direct user using the technology. The child being accompanied by the adult can be considered in this way, but in addition there may be many other indirect users of transport systems and many people other than travellers may need to consult timetables. For example, taxi drivers are major users of airport information systems, as they often take people to and from airports. Also a particular transport system will be just one stage in a long journey, and these different stages all interact. Thus it may be necessary to consider how well the timetables for different transport system work together. It is also important to try and consider the needs of other parties who will have a "stake" in a system, and what their needs may be. In the transport sector this could include a wide range of actors, including service providers and maintenance engineers.

Use whatever data you have available from analysis, customer contacts, etc., but do not be reluctant to use your imagination where data is missing. Mark such assumptions so that you remember which are assumptions and which are based on data. When you have identified user groups according to the categories listed, so that each category is represented by a set of people that you can think about in a concrete way, proceed to the next stage 'User Characteristics'.

It is difficult for stakeholders, often with little background in information systems, to formulate their requirements precisely and for a system designer to correctly mirror those requirements in the specification documentation. It is therefore imperative that requirements analysis, specification and design are treated in a disciplined way by following rigorously through the RESPECT framework.

1.3 Specify user characteristics

Objective

This section records the context of each user group selected within **Section 1.2** (e.g. end users, customers, maintainers). This data will be used to help to identify critical design features to be fed into the user requirements specification.

The user group characteristics are expressed either for users of the current system or users of the future system.

Relevant characteristics include: age range, gender, culture, education, intelligence, language, physical attributes, frequency of use, discretion to use, experience of system, general IT experience or training.

Process

To help gather this information, **FORM 1.3** should be completed for each relevant user group.

To complete the form, carry out the following steps:

1. Fill in the first (left hand) column of the form completing all relevant user characteristics.

2. Consider each of the characteristics in turn and write down any implications for the design of the system in the second column. These will become provisional user requirements for the system.

3. Assign each potential requirement a reference number as shown in the following form (1.3.1, 1.3.2 etc.). This will allow the information to be traced back to this original table. Where a characteristic has two or more implications for the design, give each a separate number.

Please refer to RESPECT D4.2, section 3.3.1 on user related factors which will help to identify user requirements based upon particular types of user: dedicated, professional, intermediary, technically experienced, novice and occasional. User language is also considered.

Please refer to RESPECT D6.2 ‘Requirements specification and evaluation for user groups with special needs’ for guidelines on user requirements and evaluation methods relating to users with impairments and disabilities (visual, hearing, motor and cognitive), elderly and young users.

FORM 1.3 - USER GROUP CHARACTERISTICS (EXAMPLE)

1.3 User group characteristics		
System name: *New bank machine* **User group:** *General public* ⬅ Form completed for users groups selected in FORM 1.2		
CHARACTERISTICS	**POTENTIAL USER REQUIREMENTS**	**REF.**
Size of user group *Full population of UK plus overseas visitors*		
Age range *18 upwards*	*Given particular consideration to older user groups who may be more reserved about new technology.*	*1.3.1*
Gender *Roughly equal numbers of males and females.*	*Include equal numbers of males and females in user trials*	*1.3.2*
Language and culture *English will be main language. Some areas of country will include up to 30% of population where English is second language.* *Used by tourists, especially from EU*	*Use English language and up to 8 other language options, depending on local area.* *Use simple terminology, diagrams and pictures.*	*1.3.3* *1.3.4*
Educational level/Qualifications *Any level.*	*Design to be usable by people who may have limited reading skills.*	*1.3.5*
Physical limitations/Disabilities *Full range* *Includes people with physical/visual handicaps*	*Ensure that system keyboard and screen are placed at a standard height.* *Use easy input device: if a keyboard, larger keys, secondary means of identification.*	*1.3.6* *1.3.7*
Special skills (e.g. touch typing, use of mouse, spatial awareness) *None*		

⬇ **Transfer to relevant FORMS in Phase 3**
e.g. FORM 3.1 General system characteristics
or FORM 3.4 Technical environment

1.3 User group characteristics (continued)		
CHARACTERISTICS	**POTENTIAL USER REQUIREMENTS**	**REF.**
Experience with similar systems *70% of users will have used bank machines and elsewhere*	*Try to make the system conform with any accepted ad hoc standards for bank machines.*	*1.3.8*
IT Experience *Variable, assume none.*	*Use very supportive dialogues to make user feel comfortable.* *Develop attractive interfaces.*	*1.3.9* *1.3.10*
Knowledge of task *Variable, assume none*	*Use highly supportive interface with clear logical structure.* *Use terms that users will understand.*	*1.3.11* *1.3.12*
Previous training *None*		
Frequency of use *Mostly first-time users or very infrequent. May return to system*	*Use very supportive dialogue - easy to learn and remember.*	*1.3.13*
Motivation to use *May be reluctant to use.*	*Make system appear attractive to use.*	*1.3.14*
Discretion to use *Can users decide whether to use the product?* *Can ignore system or abandon for any reason.*	*Make attractive and easy to use.* *Ensure that results can be achieved quickly.*	*1.3.15* *1.3.16*
Likely concerns *Being robbed or defrauded*	*Provide security feature e.g. alarm button.* *Ensure design allows privacy when using machine.*	*1.3.17* *1.3.18*
Other relevant characteristics *Wish to attract casual enquirers who may be short on time.*	*Make as attractive and simple as possible for new users.*	*1.3.19*

Transfer to relevant FORMS in Phase 3
e.g. FORM 3.1 General system characteristics
or FORM 3.4 Technical environment

1.4 Describe Technical environment

Objective

Information is collected about the future technical characteristics of the system. This includes details of the system hardware, software, and documentation, where this is known beforehand. Other equipment used to carry out related tasks such as a telephone, dictation machine or reference books is also documented.

Process

To assist in capturing details about the technical environment, complete **FORM 1.4** by carrying out the following steps:

1. Fill in name of system and user group at the top of the form.

2. Fill in first (left hand) column of the form completing all relevant technical characteristics.

3. Consider each of the characteristics in turn and write down any user-related Potential User Requirements, for the design of the system, in the second column.

4. Review each of the Potential User Requirements and assign it a reference number. Where a characteristic has two or more user-related implications for the design, give each a separate reference number.

FORM 1.4 - TECHNICAL ENVIRONMENT (EXAMPLE)

1.4	**Technical Environment**	
System name: *New Bank machine* **User group:** *General public*	⬅ Form completed for users groups selected in FORM 1.2	
CHARACTERISTICS	**POTENTIAL USER REQUIREMENTS**	**REF.**
Hardware which user will interact with e.g. desktop PC, printer, kiosk. *Standard bank machine kiosk to be adapted as necessary to provide new services.* *Touch screen may be used if required.* *Speaker and microphone may be used for speech input and output if required.*	*Hardware should be robust .* *Layout of interface elements and keypad should be consistent with existing conventions and standards.*	*1.4.1* *1.4.2*
Software environment in which system will run **e.g. Windows, WWW Browser.** *The system will run within the 'IBM Public Terminal' operating system.*	*IBM Public Terminal operating system will be used.*	*1.4.3*
Software environment to be used to develop system. *The system will be developed with 'Public kiosk builder'*	*Users representatives should be given the opportunity to see existing applications developed with same software.*	*1.4.4*
Other equipment required for use alongside system. *Telephone to contact maintenance if system fails.* ** Note entry would go into a table for Bank Staff users.*	*Bank machine could be adapted to automatically send a message to maintenance if a major failure occurs.*	*1.4.5*
Reference materials required either to perform tasks with system or to learn about or operate system *Guidebook to the new bank machines.*	*Card should give clear step-by-step instructions for main transactions which can be followed when using the machine.*	*1.4.6*

⬇ **Transfer to relevant FORMS in Phase 3**
e.g. FORM 3.1 General system characteristics
or FORM 3.4 Technical environment

1.5 Describe physical environment

Objective

The aim is to capture information about the future physical environment. This includes, for example: workstation layout, workplace design and physical conditions (the visual, thermal, auditory and atmospheric environments, as well as environmental stability.)

Process

To assist in capturing details about the physical environment, **FORM 1.5** should be completed by carrying out the following steps:

1. Fill in name of system and user group at the top of the form.

2. Fill in the first column ('Characteristics') of the form completing all relevant physical environment characteristics.

3. Consider each of the characteristics in turn and write down any user-related implications ('Potential user requirements') for the design of the system in the second column. These will become provisional user requirements for the system.

4. Review each of the implications for design and assign a reference number to it. This will allow the information to be traced back to this original table. Where a characteristic has two or more user-related implications for the design, give each a separate reference number.

Physical characteristics include the following:

Thermal and atmospheric environment

- If the system is to be used in the open, for example, an electronic ticket machine or timetable system for the public, then it will need to be designed for all weathers. This will include specifying the system to resist rain, moisture and temperature extremes. This may have implications such as users wanting to wear gloves and being able to use the system keyboard at the same time.

Auditory environment

- The level of sound that takes place may have an impact on the use of the system and it may be necessary to specify ways of damping down sound. For example in a lorry, the engine sound may hamper the driver's mate from communicating by radio to the control centre.

Vibration or instability

- Vibrations are a common problem for travellers and can hamper activities of both drivers and passengers. It is necessary therefore to determine the level of vibration and to ensure that account is taken of it in designing system controls.

Visual Environment

- The visual environment will also affect people's ability to use a system, and both low and high levels of lighting can impair the user. A public information system, for instance, must be usable both during the day and at night. Sunlight may produce glare on the user's screen. Details of the visual environment must therefore be recorded in order that potential user problems will be considered.

Space and furniture

- The characteristics of the current installation place must be studied so that the user will have enough space to operate the system safely and comfortably.

User posture

- The postures that the user will generally adopt when using the system should be recorded (e.g. standing and looking down at a display, height 1.5m) in case there are any implications for system design.

Location

- Here it is necessary to consider where the system will be located in relation to the workplace, and where the workplace is located. It may be important to consider how close this location is or needs to be to the target areas of influence, resources, fellow work colleagues, customer's and possibly the user's home.

Health and Safety hazards

- Any health and safety conditions must be considered so that safeguards can be built into the user needs analysis activity. In this way, suitable safeguards can be incorporated into the user requirements specification.

Protective clothing and equipment

- Here should be recorded any protective clothing or safety equipment that the user may be wearing, either by preference (e.g. winter clothing) or as a requirement of the job in the workplace. This includes items of clothing or equipment which protects the user from the effects of high or low temperatures.

FORM 1.5 - PHYSICAL ENVIRONMENT (EXAMPLE)

1.5 Physical Environment		
System name: *Bank machine* **User group:** *General public*	⬅ Form completed for user groups selected in FORM 1.2	
CHARACTERISTICS	**POTENTIAL USER REQUIREMENTS**	**REF.**
Thermal and atmospheric environment *UK outdoor weather: conditions.*	*Equipment should work in the following conditions:* *Temperature -10c to +40c* *Humidity 55% - 90%, Rainfall*	*1.5.1*
Auditory environment *UK urban street*	*Any use of auditory feedback or output may be drowned by street noise unless some form of earpiece or volume control is available.*	*1.5.2*
Vibration or instability *Not applicable*		
Visual Environment *Will be used during the day and night. Sunlight may produce glare on screen.*	*Sighting of machine should avoid glare where possible. Screen filters and matt screen surfaces should be tested to see if they reduce potential problems. Needs to be luminescent for use in the dark.*	*1.5.3*
Space and furniture *Bank machine should be easy to reach for a wide range of members of the public.*	*Bank machine should be mounted 1m. above ground, inset into the wall.*	*1.5.4*
User posture *Bank machine will normally be used standing.* *Wheelchair users will be sitting.*	*Bank machine should be reachable by at least 80% of wheelchair users both in terms of height and posture when operating the machine.*	*1.5.5*
Location *Street, public thoroughfares*	*Ensure machine is clearly visible and signposted for people trying to locate it.*	*1.5.6*
Health and Safety hazards *Danger of robbery and mugging of people withdrawing cash.*	*Position bank machine in the open and with extra lighting to maximise safety.*	*1.5.7*
Protective clothing/equipment *Winter clothing would include gloves, muffs etc.*	*Keys should be operable by users wearing gloves.*	*1.5.8*

⬇ **Transfer to FORM 3.8 Physical environment**

1.6 Describe Social and Organisational environment

Objective

In this section, information is captured about the future social and organisational environment. This includes:

- General structure (hours of work, group working, job function, working practices, assistance, interruptions, management structure, communications structure).
- Attitudes/culture (IT policy, organisational aims, industrial relations).
- Job characteristics (job flexibility, performance monitoring and feedback, discretion, valued skills).

Note that this part is different from the specification of organisation and business requirements for the system but may be assisted by results from that analysis.

Process

To assist in capturing details about the organisational environment, **FORM 1.6** should be completed by carrying out the following steps:

1. Fill in name of system and user group at the top of the form.
2. Fill in the first column of the form, completing all relevant social and organisational characteristics.
3. Consider each of the characteristics in turn and write down any user-related implications for the design of the system in the second column. These will become provisional user requirements for the system.
4. Review each of the implications for design and assign a reference number to it. This will allow the information to be traced back to this original table. Where a characteristic has two or more user-related implications for the design, give each a separate reference number.

Examples of the factors to be considered are as follows:

Staff and management structure

- This aspect includes descriptions of organisational structures within the environment in which the system will operate. It should also include descriptions of people's roles so that new procedures associated with the system maintain levels of status and activity satisfaction.

Assistance available

- When a new system is implemented users often require support in learning how to use it and in overcoming problems. The possibility to offer support should be considered. Suitable support mechanisms should then become part of the user needs specification.

Interruptions, stressful conditions

- Interruptions to travellers and periods of high workload need to be recorded as part of the context in order that the system will be designed to cope with, and not contribute to, these conditions.

Communications structure

- The new system must be specified so that communications at least as effective as the current system are maintained.

Privacy

- Where privacy is a key issue, the new system should be specified so that existing privacy conventions are not compromised.

Performance feedback

- Users generally like to receive performance feedback, so if this is part of the current system, it should be maintained within the new system.

Job function

- Here the main roles of the users are listed to show the scope of there current jobs.

Safety and security

- An important user need is to feel safe and secure in performing their tasks. If the system does not give the impression of safety (avoidance of accidents) and security (protection against loss or injury) users will not perform well or will not use the system.

FORM 1.6 - SOCIAL AND ORGANISATIONAL ENVIRONMENT (EXAMPLE)

1.6 Social and Organisational Environment		
System: *Bank machine* ⬅ Form completed for user groups selected in FORM 1.2 **User group:** *General public*		
CHARACTERISTICS	**POTENTIAL USER REQUIREMENTS**	**REF.**
Staff and Management structure *Not relevant.*		
Communications structure *Not relevant*		
IT Policy *All bank branches to have own bank machine and to encourage usage to reduce staff time.*	*Staff should be prepared to give advice to the public on using Bank machines. Counter staff should always be available to handle similar transactions if person does not wish to use a bank machine.*	*1.6.1*
Organisational aims *Not relevant*		
Industrial Relations *Not relevant*		
Performance monitoring *The public will expect reasonably quick and consistent response times.*	*Bank machines should be monitored for response speeds and number of transactions per day.*	*1.6.2*
Performance feedback *Bank staff will need to be able to check manually on machine performance, and current quantity of paper and money.*	*Staff should be able to interrupt queue to perform a quick check if query arises about quality of output.*	*1.6.3*
Group working *User normally alone, sometimes with partner*	*Allow two users to view/access bank machine comfortably.*	*1.6.4*
Assistance required or available *Possibly available from bank staff or others in queue. Required by novice users or if system fails and card is lost.*	*Assistance is not normally available at external bank machine. User needs way of registering problem and to request help soon afterwards.*	*1.6.5*
Interruptions, stressful conditions *Queues may build up during busy periods.*	*Customer needs a way of abandoning transaction if they cannot proceed and feel under pressure.*	*1.6.6*

⇨ Continued on next page

⬇ **Transfer to FORM 3.9 Social and organisational environment**

1.6 Social and Organisational Environment (continued)		
System: *Bank machine* **User group:** *General public*		
CHARACTERISTICS	**USER REQUIREMENTS**	**REF.**
Safety and Security *Danger of theft and mugging particularly from external machines.*	*Bank machine may provide an alarm bell to signal help required if theft takes place.*	*1.6.7*
Privacy *Danger of others seeing financial details of customer or learning PIN number.*	*Bank machine should provide sufficient barriers to prevent others from seeing the transaction.*	*1.6.8*
Job function *Not applicable*		
Hours of work *Not applicable*		
Job flexibility *Not applicable*		
Valued skills *Not applicable*		

Transfer to FORM 3.9 Social and organisational environment

At this stage it may also be necessary to develop the organisational basis for the new system. This will show at a high level how the users will interact with the system and communicate with other people as part of the work process or operating environment. It will also show how information will flow through the system. The organisational design should be developed in collaboration with different user group representatives. Ideally this will be carried out using the methods of:

- **4.6 group discussion** or
- **4.7 interviewing** individuals (described in **Part C**).

1.7 Identify user goals and tasks

Objective

The section considers the following:

- To list the range of user (task) goals that can be identified for each user group.
- To show how these break down into individual tasks that need to be performed by different types of user.

Process

To produce the user goal list and related user tasks, use **FORM 1.7** below.

1. Write down in the left hand column, any goals that are of concern to users. These may be single user group goals or goals that two or more users are concerned with. Try to express each goal in general terms and keep the number down to below 20.
2. At the top of column 2, 3, 4 etc., write down the name of each user group.
3. Now for each user goal, mark with a cross ('X') the columns relating to users that are likely to be involved in helping to achieve the goal. This will show how 2 or more users may work to achieve the same goal.

FORM 1.7 - USER GOALS AND TASKS (EXAMPLE)

1.7	**User Goals and Tasks**			
System: *New bank machine*				
	Customers	*Bank staff*	*Maintenance*	
G1 *Access required service quickly and safely*	X			
G2 *Replenish money*		X		
G3 *Replenish paper*		X		
G4 *Report fault with bank machine*	X	X		
G5 *Repair fault*		X	X	

⬇ **Goals are expanded in 'Review current process', in FORM 1.8.**

1.8 Review current processes

Objective

The aim of this section is:

- To review current processes for achieving user goals and to document problems that may arise within that process.

- Highlight possible ideas for addressing the problems that may arise.

Process

For each goal (e.g. G1, G2 etc.) complete **FORM 1.8**, listing the basic steps in achieving that goal. Task steps might be identified by means of **group discussions** (see section 4.6) or **interviews** (see section 4.7). It may also be necessary to **observe** tasks (section 4.8) currently being performed to identify the individual steps.

To complete the form, carry out the following steps:

1. Break the goal down into a series of task steps.

2. Review each task step, and think about possible problems, events or task variations that might affect the basic flow of the task. Write down any such problems or unusual events in the second column. These form the basis of **scenarios** that will be used to test the feasibility of the system design.

   To generate such problems consider, for example any inputs or task dependencies. For a bank machine, they would be the bank card and knowledge of the PIN. This could lead to the problem of the user forgetting the PIN after inserting the card. Similarly another aspect is the danger of theft, so the problem to be noted of the user being robbed, or feeling suspicious about the person standing behind them in the queue. Not all items need be problems — they may be opportunities for design ideas. Thus a characteristic of bank machine usage is that the user withdraws the same amount of money on 90% of occasions, for which a quick interaction path could be designed.

3. Now consider possible user requirements that could overcome the problems or variations listed. Write these in **column 3**. These are then taken forward into the development of the new design concept within Phase 2.

   *For tasks that need to be analysed in more detail, a **Task Analysis** should be carried out. (see section 4.15 in Part C, for further description). This includes diagramming techniques **Task Flow Diagrams** and **Task Decomposition**.*

Refer also to RESPECT D4.2, section 3.3.2, on task related factors which will help to identify user requirements based-upon particular types of task, either: data entry,

querying a database, reading or browsing, lengthy and complex tasks, monitoring and safety critical tasks, and task interruptions.

FORM 1.8 - CURRENT PROCESS (EXAMPLE)

1.8 Current Process			
User group or groups concerned: *Bank customer* **Goal:** *G1 Access required service quickly and safely (e.g. Withdraw cash)* ← Form completed for all user goals listed in FORM 1.7			
TASK STEP	**PROBLEM/TASK VARIATION**	**POTENTIAL USER REQUIREMENT**	**REF.**
1. Line up to use bank machine.			
2. Insert card.	*Card inserted wrong way around.*	*Notch on card.* *Picture on machine as guidance*	*1.8.1* *1.8.2*
3. Enter PIN.	*User forgets PIN.* *Machine abandoned.* *Machine breaks down.* *Next person in queue takes too much interest in transaction.*	*Allow thumbprint.* *Quick reset button.* *Provide more privacy with surround on machine*	*1.8.3* *1.8.4* *1.8.5*
4. Select 'withdraw cash' service.			
5. Select or enter required amount.	*Amount required greater than current limit.* *Machine runs out of money.* *User decides not to proceed with transaction.* *User selects same amount of money on most occasions.*	*Display amount available.* *Provide cancel button.* *Provide short cuts to commonly required options.*	*1.8.6* *1.8.7* *1.8.8*
6. Choose whether a receipt is required.	*Machine runs out of paper.*		
7. Take card.	*Card not returned.*		
8. Take money and receipt (if chosen) .	*No money and/or no receipt returned.* *Incorrect amount of money returned.* *User threatened by passer-by and money stolen.*	*Provide button to register problem at given time.* *(See above)* *Camera on machine to video transactions*	*1.8.9* *1.8.10*
↓ **All information used to generate task scenarios and propose new processes in Section 2.4**		↓ **Transfer to relevant forms especially 3.5 Functions/Features**	

1.9 Review similar systems and products

Objective

The aim of this section is:

- To record information about similar, possibly competing, systems that may be included in the design of the new system. This will normally be attractive features that should be considered for inclusion. However problems with other systems that should be avoided in the new system may also be considered.

Process

Complete **TABLE 1.9** below as follows:

1. List each of the user goals in column 1.
2. For each user goal, consider the functions of features to be included or excluded from the current system.
3. Write the name of the system that the function or feature relates to in column 2.
4. If the function or feature is to be included, write this in column 3. If it is to be excluded, write this in column 4.

FORM 1.9 - FUNCTIONS AND FEATURES OF SIMILAR SYSTEMS (EXAMPLE)

1.9 Functions and features of similar systems				
System: *New bank machine*				
	PRODUCT NAME	**GOOD FEATURE TO INCLUDE**	**POOR FEATURE TO EXCLUDE**	**REF.**
GENERAL IDEAS				
GOAL SPECIFIC				
G1 *Access required service quickly and safely*	*IBM Cash point* *Sirocco Corp.*	*Non reflective screen*	 *Small buttons*	*1.9.1* *1.9.2*
G2 *Replenish money*	*AT&T Bank machine*	*Low warning before machine empty*		*1.9.3*
G3 *Replenish paper*	*AT&T Bank machine*	*Low warning before machine empty*		*1.9.4*
G4 *Report fault with bank machine*				
G5 *Repair fault*	*Sperry Univac*	*Step-by-step diagnostic routine*		*1.9.5*

Transfer to FORM 3.5 Functions and Features and to FORM 3.4 Technical environment

1.10 Produce design ideas and concepts

Objective

The aim of this section is to develop one or more ideas upon which the new design would be based. Each idea may be regarded as a system concept.

It is desirable to develop several concepts and to compare them. The most feasible concept is then taken forward as part of the user requirements specification. It is also likely that the best parts of different concepts will be used to create as a single, agreed basis for the design.

Process

The following methods may be used to generate system concepts. These include:

- a **brainstorm** (section 4.1) session where design team members and users think and present ideas in an unconstrained manner.
- a **parallel design** (section 4.10) session where different groups, given the same design brief, come up with their own concept. Afterwards the different concepts are compared and the most feasible one is taken forward.

To allow users to visualise and assess a system concept, it may be represented as:

- a set of written **scenarios** (section 4.12) of how the new system might be used
- a **storyboard** (section 4.13) presenting a sequence of drawings showing the system in action
- a **paper prototype** (section 4.9) which users may manipulate

Descriptions and guidance on applying all the above techniques are given in **Part C (section 4)** of this document.

A list of the suggested ideas and concepts should be produced as an index. **FORM 1.10** provides an example list:

FORM 1.10 - DESIGN IDEAS AND CONCEPTS
(EXAMPLE SHOWING FORM PARTIALLY COMPLETED)

1.10	**Design ideas and concepts**		
System: *New bank machine*			
	IDEAS AND CONCEPTS	**COMMENTS**	**Take Forward? /Ref.**
GENERAL IDEAS			
	Speech synthesis for guidance.		
GOAL SPECIFIC			
G1 Access required service quickly and safely	*Question and Answer mode for beginners.*		
G2 Replenish money	*Simple reload drawer while machine still running.*		
G3 Replenish paper	*Simple reload drawer while machine still running.*		
G4 Report fault with bank machine	*Allow user to notify bank if their card gets stuck in the machine, by pressing special button.*		
G5 Repair fault	*Allow bank staff to repair some faults to save visit from maintenance staff.*		

Transfer those taken forward to Form 3.5 Functions and Features

Designing user interfaces is a complex and highly creative process that blends intuition, experience, and careful consideration of numerous technical issues. Designers should begin with a thorough task analysis for the user community. Explicit recording of task objects and actions based on task analysis can lead to the useful construction of metaphors or system images. There are several methods for envisioning design as described in Chapter 22 of Preece et al (1994). These include the use of sketching, scenarios and storyboards.

Once the general design concept is produced, the computer objects that users need to perform their tasks and the actions they will perform with them will be identified. Next designers

create consistent and meaningful syntactic forms for input and display. Extensive testing and iterative refinement are then performed early on to validate the design.

There are several methods for representing the logical structure of a user interface during the design process such as flow charts, dataflow diagrams, state transition diagrams, and structure charts (see Hartson and Hix, 1989, Hartson et al, 1990, and Sutcliffe, 1991). Selection of an appropriate user interface style or combination of styles is also an important part of user interface design. The style must be appropriate for the users, the work they are doing, the system and the environment. Preece (1994), chapter 13 offers an up-to-date and useful general overview of the different interaction styles. Shneiderman (1987) discusses in depth menu dialogues, command languages, and was one of the first to offer practical guidelines on the design of direct manipulation interfaces.

User-interface structures, for particular user goals or task steps, may also be developed to illustrate the ideas listed in **FORM 1.10** (as shown below). Figure 5 (Example 1) shows a global interface structure for bank machine access (goal 1), while Figure 6 (Example 2) expands on the tasks of inserting a card and entering a PIN.

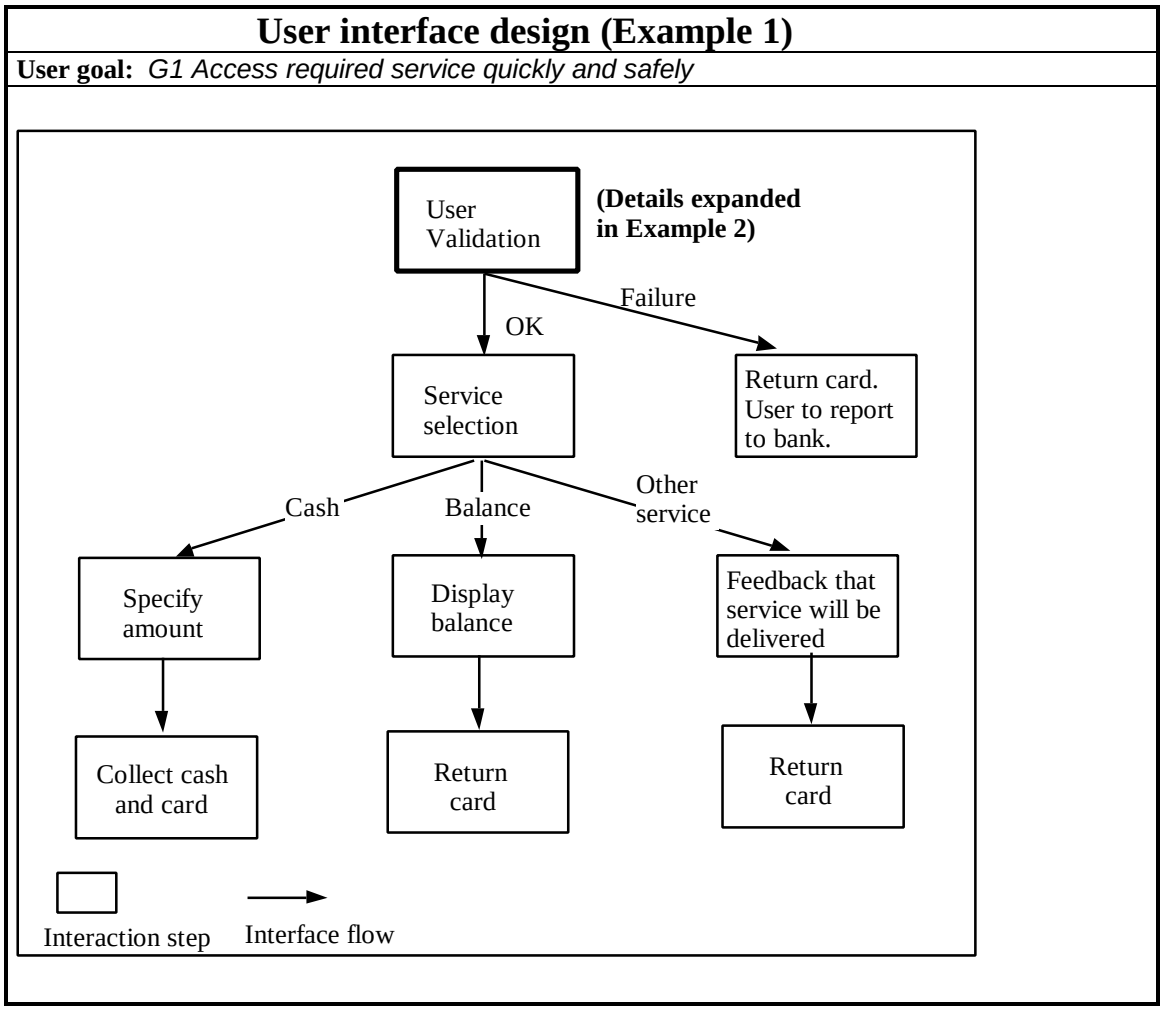

FIGURE 5. GLOBAL USER-INTERFACE STRUCTURE

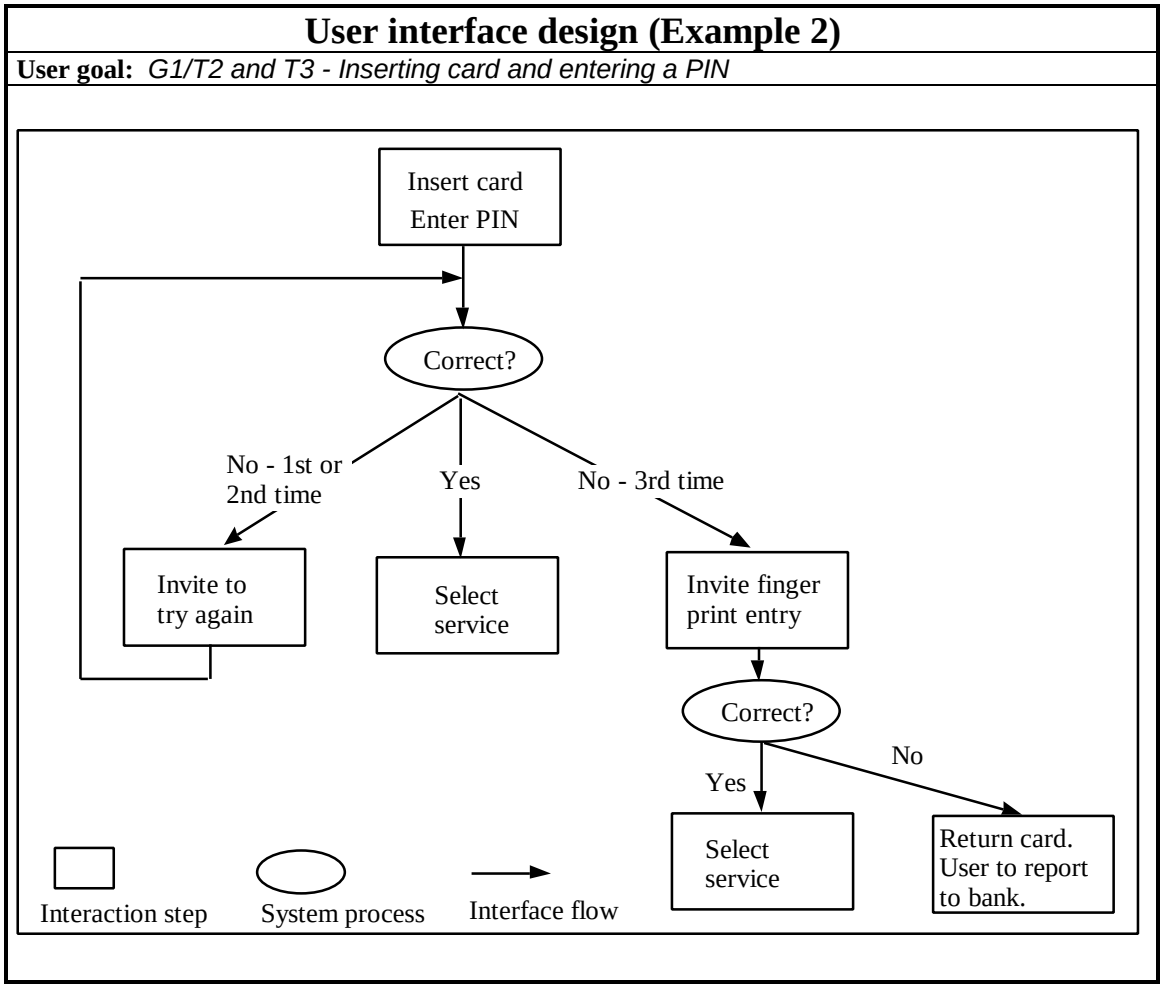

FIGURE 6. EXPANSION OF USER-INTERFACE COMPONENT

The proposed organisational design can be expressed with a diagram, using simple symbols, as shown below in the Figure below:

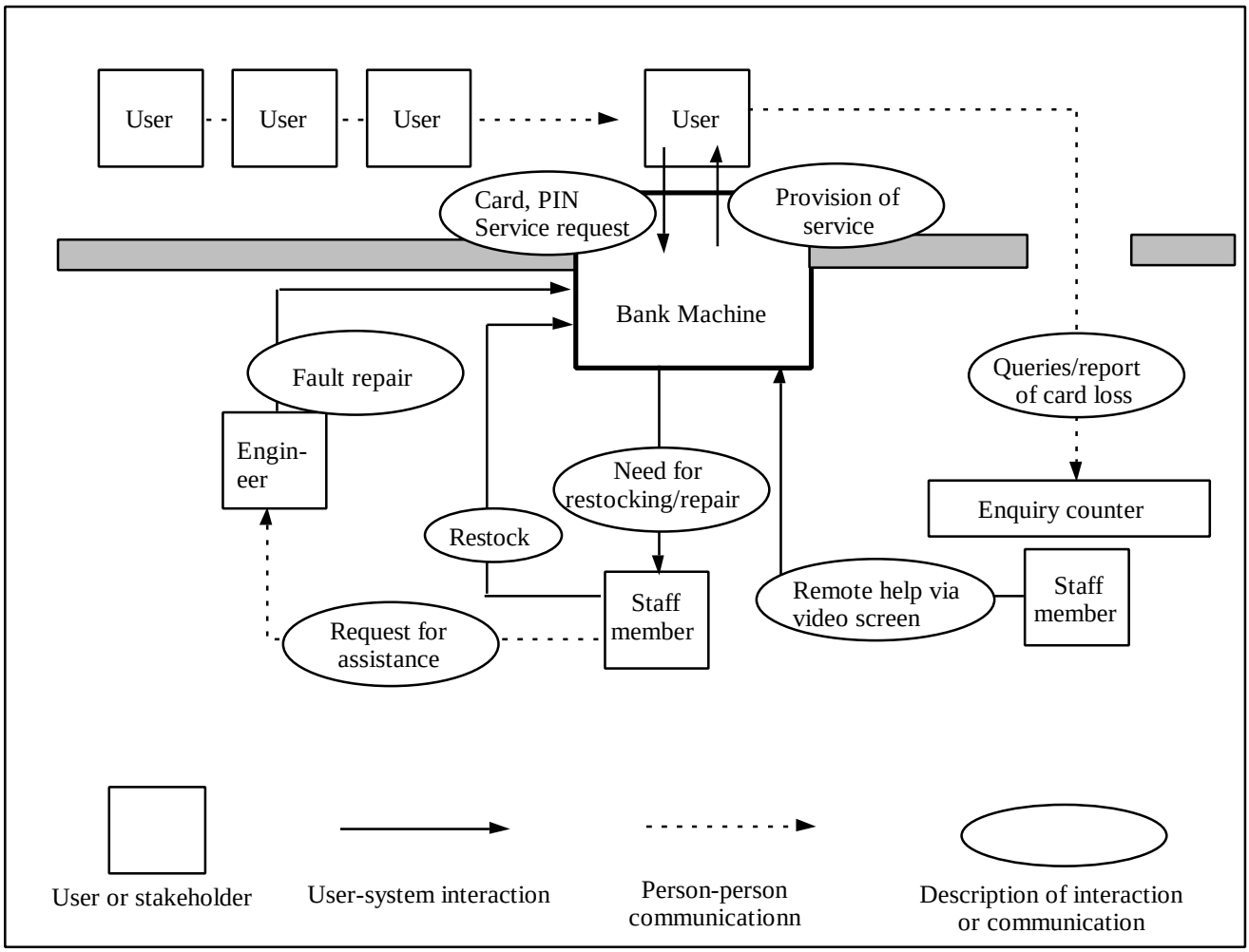

FIGURE 7. ORGANISATIONAL PROCESS DIAGRAM

The organisational design should be supplemented with a written description to explain the process.

It is possible to simulate the organisational design by setting up a working environment, such as a simulated office with people acting out different user roles. Such simulations can then can then be assessed either:

- by performing an **expert review** (section 1.11) to walk through the system concept without an operational system in place, or
- by testing with a **prototype system** (section 2.6).

1.11 Perform expert review of designs

Objective

The aim of this section is to review all the ideas proposed for the new system and to evaluate them by means of an expert review. Those taken forward will feed into the development of a system prototype (section 2.3)

Process

1. Consider the list of ideas produced on **FORM 1.10**. Consider the feasibility of each design and write suitable comments in the relevant column (column 3).

2. Consider each idea again and either put a tick or cross against it (in column 4) to indicate whether the idea will be taken forward into the design or not. For those ideas taken forward, assign a reference number to each one.

FORM 1.10 - REVIEW OF DESIGN IDEAS AND CONCEPTS
(EXAMPLE SHOWING FORM FULLY COMPLETED)

1.10	**Design ideas and concepts**		
System: *New bank machine*			
	IDEAS AND CONCEPTS	**COMMENTS**	**Take Forward? /Ref.**
GENERAL IDEAS			
	Speech synthesis for guidance.	*Expensive but may be useful for those with impaired vision.*	✓ *1.11.1*
GOAL SPECIFIC			
G1 Access required service quickly and safely	*Question and Answer mode for beginners.*	*Useful idea but may slow down the interaction process*	✓ *1.11.2*
G2 Replenish money	*Simple reload drawer while machine still running.*	*Not recommended as may cause electrical faults.*	✗
G3 Replenish paper	*Simple reload drawer while machine still running.*	*Not recommended as may cause electrical faults.*	✗
G4 Report fault with bank machine	*Allow user to notify bank if their card gets stuck in the machine, by pressing special button.*	*Good idea as long as button only active for when card is actually lost in the machine*	✓ *1.11.3*
G5 Repair fault	*Allow bank staff to repair some faults to save visit from maintenance staff.*	*Worth considering. Danger of overloading staff.*	✓ *1.11.4*

Transfer those taken forward to FORM 3.5 Functions and Features

1.12 Move to Phase 2 ?

Having developed a range of ideas for the new system, an assessment is made as to whether the design ideas and concepts form a sufficiently good basis for further development as a prototype.
If so, then the process continues with Phase 2.
If not, then return to 1.10 to consider new ideas that may be included in the system.

Phase 2

Prototype and User test

PHASE 2 of the user requirements specification process consists of the second iteration around the user-centred design loop as shown below.

- **CONTEXT:** The first part of phase 2 is to identify task scenarios that will be used as a way of testing the system design. At least one scenario is produced for each user goal. Each scenario represents a user goal set within a particular context. Usability goals to be achieved for each scenario are also produced.

- **REQUIREMENTS:** For each scenario, a set of steps representing an interactive process is developed. At the same time, a list of potential functions of features to support that process is also documented.

- **DESIGN:** Here an interactive prototype of the system is produced, based on the list of functions and features defined. This will be used to test the system concept against the scenarios with potential users.

- **TEST:** Here the prototype is used to carry out the different scenarios with users. Any problems are then documented. A review is carried out of the tasks that each user will carry out for all user goals. The acceptability of each group tasks for that user group is considered. If both the prototype is satisfactory and the user task groupings are acceptable, the process may move on to the documentation of user requirements in Phase 3.

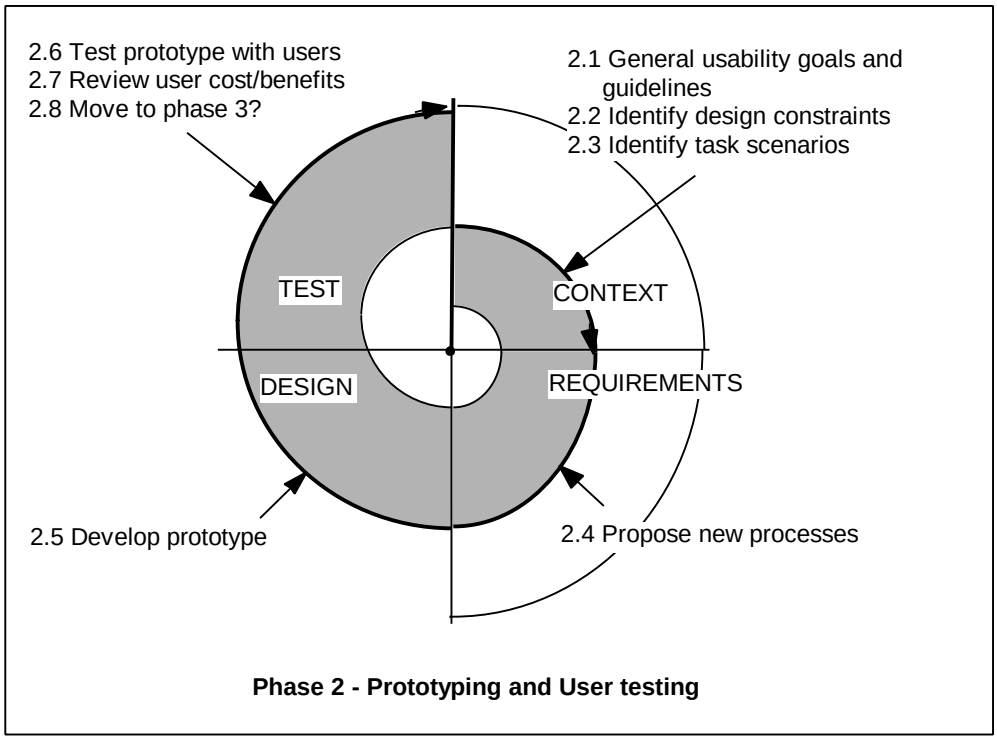

FIGURE 8. PHASE 2 - PROTOTYPING AND USER TESTING

2.1 General usability goals and guidelines

Objective

The aims of this section are to:

- Specify general usability goals for the system and,
- Review guidelines or styles guides to which the user interface should conform, and possibly to produce a short checklist.

Process

General usability goals

1. Review the following general usability goals shown in **FORM 2.1**, write in the second column any specific details about how that goal will apply to the current system.

2. Identify those that are particularly relevant to the system and mark these appropriately (e.g. with a tick, ✓).

The information collected in Stage 1 can be used to highlight the kinds of usability goals that may be important.

Thus for example if the users are the general public who are to use, say, an information system on a walk up and use basis, then the system must be simple and helpful to cater for the capabilities of this user population. It must also be efficient if people are to be able to achieve their task goal within the short time they are likely to have available.

In contrast the user of an air traffic control system is likely to face the problem of excess mental workload in controlling a region of airspace and so critical usability goals are likely to be: reduction in mental workload and avoidance of error.

FORM 2.1 - GENERAL USABILITY GOALS (EXAMPLE)

2.1	**General Usability Goals**	
System: Bank machine **User:** General public		
USABILITY GOAL	USER REQUIREMENT WITH RESPECT TO GOAL	KEY GOALS
Effectiveness Quality or quantity of output/task completion.	*It is important that the user be able to complete tasks accurately.*	✓
Efficiency Time to perform task, time compared with an expert.	*Users will expect to use the bank machine quickly and will become impatient if response times are slow.*	✓
Satisfaction Perceived satisfaction or enjoyment in using the system.	*The satisfaction from using the system will derive from completing a task successfully and quickly.*	
Learnability Ability to use the system help or manuals to perform the task.	*Short instruction leaflets may be provided. However it is not expected that system will rely on on-line help.*	
Intuitiveness Ability to perform the tasks with limited introduction.	*New users will be discouraged from using the system if it is not intuitive.*	✓
Helpfulness/supportiveness Ability to overcome problems that arise.	*The system should give some guidance if the users get stuck.*	
Controllability Perceived feeling of being in control/tracking performance etc.	*Users should feel confident that they can operate the system and take suitable action if something unexpected happens.*	
Avoiding excessive mental load Perceived mental effort, or physical indicators.	*The system should not require the user to remember a long PIN number.*	
Avoiding excessive physical load Heart rate, respiratory measurement.	*People with motor impairment, and in wheelchairs should be able to operate the system comfortably.*	
Safety To be able to operate the system safely	*The system must be sighted in a location in which users feel safe and should be well lit.*	✓

While all the above goals may have some relevance to a given system, it will be necessary to prioritise the goals, select those of most relevance and specify them more precisely as operational

goals. The achievement of these goals is then a matter of subjective judgement by users or human factors experts.

Guidelines

There are many sets of guidelines that may be employed to assist in the design of a usable system. In Appendix 1, a series of general user interface guidelines are presented. Those involved in the development of the design concept and prototype should review them to make themselves aware, or to remind themselves of the main principles to bear in mind.

It is recommended to use a highlighter pen to highlight any phrases that are particularly relevant to the current application. It may also be helpful to produce a short checklist of items that should be remembered when developing the system design.

It is also necessary to review any internal design guides that need to be followed or de facto interface standards (e.g. Windows 95, Windows NT, OSF Motif). Later in section 3.10, a list of standards that the design should follow is also produced.

2.2 Identify design constraints

Objective

In this section, any design constraints or areas of flexibility that will apply to the new system are listed. These may include statements such as that 'it must be handheld', 'it must not use more than 12 keys', 'its maximum storage capacity is 100 A4 pages'.

It will be important to clarify these constraints before design starts and it is later found that proposed design concepts cannot be implemented.

Some constraints may in fact define flexibility, in order to open-up possibilities for the design team. For example, the number of keys may be fixed by the layout or may be flexible.

Process

1. Review the project documentation so far prepared and try to identify possible constraints that will affect the user. These may be in terms of hardware, likely response times, memory, or technology already being used in the user environment.
2. List these constraints in the table below:

FORM 2.2 - DESIGN CONSTRAINTS (EXAMPLE)

2.2 Design Constraints	
System: *Bank machine*	REF.
System will be based on standard bank machine terminal.	*2.2.1*
Screen size will be standard 10 inches, with a resolution of 640x480	*2.2.2*
There will be 8 screen buttons for soft keys, 4 down each side of the display	*2.2.3*
There will be a keypad of no more than 16 keys. Configuration is flexible.	*2.2.4*
Colour of keys is flexible	*2.2.5*

Transfer to FORM 3.4 Technical environment

2.3 Identify task scenarios

Objective

Here a number of task scenarios which represent common or important task situations are listed. These will be used to test the success of the prototype. Usability criteria are also established to help judge the success of the system in relation to the tasks. At least one scenario is required for each user goal.

The strength of scenario development is that it allows design team members and users to consider a range of possible situations and problems and to pose them in the form 'what would happen if...?'. This can test the system against such situations.

A prototype is then developed and tested with these scenarios using a **walkthrough** (section 4.18) or set of **controlled tests** (section 4.2) in order to check that the interface provides a usable system, and helps to develop specific usability goals.

Process

A general approach for defining task scenarios is to review the user goals defined in section 1.7. Aim to have at least one scenario per goal. The scenarios should be based on common tasks, difficult tasks, and/or critical tasks. The aim is to test different features of the system design as fully as possible. They should also include characteristics of the user's context such as using a bank machine in the dark or users wearing gloves.

It is also important that the task scenarios reflect the likely changes resulting from the introduction of new systems and facilities. If possible concentrate on the following kinds of task:

- tasks done currently which could be rendered obsolete by the introduction of information systems (for example, remote fault diagnosis might render unnecessary routine maintenance inspection);

- tasks generated by the use of new systems (for example, the need to make backups of information held on the system);

- tasks made significantly easier or more complex by the changes (e.g., the recall and searching of stored information might be easier, whereas issues of security and access might be more complex).

The scenario may be generated by discussion between design team members and users. The following set of steps can assist the process:-

1. For each user goal, review the task steps for current processes and any potential problems listed in **FORM 1.8**.

2. Review also the forms containing the context information for the users who contribute to each goal including **FORM 1.3** User characteristics, **FORM 1.4** Technical environment, **FORM 1.5** Physical environment, and **FORM 1.6** Social and organisational environment.

3. For each goal, produce one or more scenarios in **FORM 2.3** below reflecting where possible important aspects of context or likely problem characteristics. To do this, write a reference number for the scenario in column 1, including the original goal from which it is derived (e.g. S1(3) - 'first scenario, third goal). Then in column 2 write a short statement to represent the scenario. This should reflect the starting conditions and problems the user might face.

   Note that it is possible to cover more than one user goal in the same scenario, as shown in FORM 2.3, with scenarios S5 and S6 both covering user goals G2 and G3.

4. Finally in column 3, write down in descriptive form what would be an acceptable outcome in terms of user performance. If appropriate, a measurable performance goal may be stated. This will be used to help assess the user's performance when they try to use the system based on the given scenario.

FORM 2.3 - TASK SCENARIOS (EXAMPLE)

2.3	**Task Scenarios**	
System: *New bank machine* **User group:** *General public* ⬇ **Based on FORM 1.7 User goals and FORM 1.8 Current processes.**		
REF.	SCENARIO	PERFORMANCE AIM
S1(1)	*The user inserts the card and enters the PIN. They request to withdraw cash. They select £50 and request a receipt. They collect the cash, card and receipt.*	*90% of existing bank machine users should be able to carry out this task within one minute.* *70% of first time users should also be able to carry out this task within one minute.*
S2(1)	*The user inserts the card and enters the PIN. They request £200, when the limit is £50.*	*Users should be able comfortably to realise what has happened when they exceed their limit and be able successfully to obtain £50.*
S3(1)	*The user inserts the card and enters the PIN. They request £100, then decide not to proceed with the transaction.*	*Users should be able successfully to exit the interaction and retrieve their card.*
S4(1)	*The user inserts the card and enters the PIN. They feel threatened by the person in the queue behind them. What should they do?*	*Users should be able to take some action which allows them to exit rapidly from the transaction and possibly raise an alarm.*
S5 (2/3)	*The machine runs out of money or paper during bank working hours. By some means it is reported to bank staff to replenish.*	*The bank machine should report the fault to bank staff who should be able to replenish the money or paper within 5 minutes.*
S6 (2/3)	*The machine runs out of money or paper outside bank working hours.*	*The machine should handle this situation in a helpful way to the bank customer.*
S7(4)	*The user inserts the card without looking at it. It is not accepted by the machine, so the user reorients and re-inserts it.* *The user forgets the PIN, and tries several attempts.*	*The machine should handle this situation in a helpful way to the bank customer.*
S8(4)	*The user inserts the card and enters the PIN. They request £100. They receive a receipt but an incorrect amount of money.*	*The machine should handle this situation in a helpful way to the bank customer.* *The system should also allow the bank staff to rectify the situation easily.*

2.4 Propose new processes

Objective

The aim of this section is:

- To review the current processes documented in section 1.8, and the scenarios described in the previous section 2.3 and to suggest revised processes for achieving user goals.

- For each scenario, a set of interaction steps are listed to demonstrate how the system should be used. A list of functions and features which support these steps are listed.

Process

For each scenario (e.g. S1, S2 etc.) complete **FORM 2.4**, listing the basic steps in achieving the goal for that scenario. System functions and features should also be listed for each part of the scenario.

To complete the form, carry out the following steps:

1. Copy the scenario description and performance aim into the top two sections of the form.

2. Review each scenario and list the steps that the users should carry out in order to achieve the scenario in column 1.

3. Consider each step and write down the functions or features that need to be included within the system to support those steps.

Refer also to RESPECT D4.2, section 3.3.2, on task related factors which will help to identify user requirements based-upon particular types of task, either: data entry, querying a database, reading or browsing, lengthy and complex tasks, monitoring and safety critical tasks, and task interruptions.

FORM 2.4- PROPOSE NEW PROCESSES (EXAMPLE)

2.4	**New processes**		
User groups concerned: *Bank customer* **Goal:** *G1 Access required service quickly and safely (e.g. Withdraw cash)*			
Scenario *S1(1)* ⬅ Form completed for all task scenarios listed in FORM 2.3 *The user inserts the card and enters the PIN. They request to withdraw cash. They select £50 and request a receipt. They collect the cash, card and receipt.*			
Performance Aim *90% of existing bank machine users should be able to carry out this task within one minute.* *70% of first time users should also be able to carry out this task within one minute.*			
TASK STEP	**POSSIBLE FUNCTIONS OR FEATURES**	**INCLUDE**	**REF.**
1. Line up to use bank machine.			
2. Insert card.	*Develop reader that will read card whichever way it is inserted.* *Notch on card.* *Picture on machine as guidance*		*2.4.1* *2.4.2* *2.4.3*
3. Enter PIN.	*Allow user to enter PIN or thumbprint.*		*2.4.4*
4. Select 'withdraw cash' service.	*System displays maximum amount that can be withdrawn.* *System offers options: 'Withdraw £20, £50, £100' on first menu.*		*2.4.5* *2.4.6*
5. Select or enter required amount.			
6 Choose whether a receipt is required.	*Allow user to select :* *'Cash with receipt' or 'Cash without receipt'*		*2.4.7*
7. Take card.			
8. Take money and receipt (if chosen) .			

⬇ **Transfer to relevant forms especially 3.5 Functions/Features**

2.5 Develop prototype

Objective

In order to help users visualise the possible system and to clarify user requirements, a rapid prototype or simulation of the system should be developed. This will demonstrate system concepts or specific features.

The aim of this section is to draw out from the Design Concept the main features of the system and the functions it should provide for development into an interactive prototype.

Process

The user interface can be prototyped in different ways, for example,

- as a **prototype or simulation** of the system concepts produced in software.
- as a simulation of the system, the interface being controlled by a person, acting as the system and responding to user input. This is known as a **'Wizard of Oz'** simulation.
- as a **video simulation** showing the concept behind the system.

There are a number of key issues, however which need to be considered (Maguire, 1996):

- **Keep within design limitations**

It is important to create a prototype which stays within the likely limits of the design context. For example, the prototype should be developed to run on the hardware that most users will have and to exhibit similar response times. This will avoid raising user expectations which will lead to disappointment when the real system is implemented.

- **Take account of likely data volume and structure**

A prototype system may appear to present a simple interface based on the example data that is incorporated into it. However in the real situation, pick lists may be longer, data records may contain more text, and there may be many more of them, which may affect the usability of the system in use. Similarly, a process control system prototype may appear manageable since the range of displayed messages or warnings is limited. However unforeseen problems may occur producing messages from the full range when the complete system is running.

- **Be aware of user representatives becoming too technically aware**

It is recommended that users are closely involved in the development of system prototypes and their inputs will help match the system to user needs. However as the process of development continues, and users learn more about the technology behind the system, and the concepts and jargon of software design, there is a tendency for them to lose sight of the problems that non-technical users may face. Thus users not involved in system development should be brought in to provide feedback on the developing prototype.

- **Over Reliance on Prototyping**

It is possible that the design team may place too much reliance upon prototyping activities. By assuming that the prototype will capture the essence of the whole system, it can lead to poor specification. If the prototype is seen as the specification itself, then designers may fail to consider the more hidden aspects of the system with the result that they will be implemented poorly. It is important then that internal aspects such as data structures and communications links are considered in terms of their possible effect on user requirements.

Please refer to Appendix 1 which offers general guidelines on the design of user interfaces.

2.6 Test prototype with users

Objective

The prototype is tested with users and observations recorded by evaluators. Also performance scores and subjective ratings are recorded which may be used as a basis for a user requirements test plan.

Process - Present to Users and Gather Feedback

The prototype can be presented to users as part of a discussion session. By 'walking through' specific system tasks (e.g. the task scenarios), the user can provide feedback on the design approach leading to changes being made where appropriate.

More formally, the system is demonstrated to users by talking through each of the system tasks. The **walkthrough** may be conducted using the concept description whether it is represented on paper or as a prototype. Guidelines for conducting a **walkthrough** are given in **Part C (section 4.18)**.

The basic approach is to for the evaluator and the user(s) to select a range of suitable **task scenarios** (from those listed in section 2.3). They then discuss how each task would be carried out, with the new system, in a step-by-step fashion.

The task walkthrough will serve to validate the system concepts and identify parts that require change. For a particular task, the different interactions required between the user and the system, or with other people, will also become clearer and specified in more detail. Further details such as task frequency, duration and criticality will also emerge leading to improved system navigation, warnings, confirmation dialogues, etc. The task walkthrough results may be recorded using a form similar to that shown below (**FORM 2.6**).

The following steps provide a guide for running **walkthrough** sessions.

1. Decide what issues or task scenarios should be covered by the walkthrough.
2. Set up a recording mechanism to gather user comments. This may be for one person to show the system and ask questions, with another person taking notes or recording the discussion on tape for transcription later.
3. Select appropriate users to take part in the walkthrough, trying to cover the range of user groups within the target population. Other stakeholders with an interest in how the system will operate such as supervisors and maintenance staff may also wish to be included within the walkthrough. Similarly stakeholders interested in the system generally or are affected by its outputs (e.g. managers and customers) may wish to view and comment on the system but at a higher level of detail.
4. Pilot the walkthrough to work out how much time is needed for each session.
5. Ensure recording facilities are available and working properly.
6. Conduct the walkthrough sessions, making sure that each session covers the issues identified beforehand.
7. Analyse the information obtained grouping them appropriately. Rather than group them by task, it may be preferable to group them according to the nature of the issue e.g. task

match, screen layout, user support etc. Try to determine how many users made the same comment and assign frequency values to them. Consider the importance of the problems identified as a result of the comments and list them in order of priority.

FORM 2.6- TASK WALKTHROUGH FEEDBACK (EXAMPLE)

2.6	**Task walkthrough feedback**		
System: *Bank machine* **User group:** *User group* **⬇ Transferred from FORM 2.3 Task scenarios.**			
TASK SCENARIO	**COMMENTS**	**MODIFIED OR NEW SYSTEM REQUIREMENT**	**REF.**
S7(4) *The user inserts the card without looking at it. It is not accepted by the machine, so the user reorients it and reinserts it.*	*Picture on machine helped user to re-orientate the card.*		
The user forgets the PIN, and tries several attempts. The machine keeps the card.	*Machine keeping card seems unsatisfactory. User prefers card to be returned to allow them to test it at the bank.*	*If user forgets PIN, system returns card for user to take into bank.* *If card not shown to bank within 5 days, it is cancelled and letter sent to customer with new card.*	*2.6.1*
S8(4) *The user inserts the card and enters their PIN. They request £100. They receive a receipt but an incorrect amount of money.*	*Users want a way to notify the bank that a fault has occurred immediately so the bank can check back whether enough money has been located.*	*If an incorrect amount of money has been dispensed, the user presses a special button to allow them to report the fault by video camera at the machine.*	*2.6.2*
		⬇ Copy to relevant Stage 3 Forms.	

2.7 Review user cost/benefits

Objective

An important aspect in weighing up alternative system concept options will be the costs and benefits that users perceive, as well as each option's effectiveness in meeting the business goals. This activity provides a basis for eliminating certain concept options and selecting the most suitable.

The information collected are the costs and benefits of the system concept from each user group's view. These will be drawn from a range of viewpoints including task characteristics, job content, security, organisational procedures and personal development. It will thus highlight general implications of the new tasks on the users' jobs and the human aspects of the organisation.

Process

In order to capture user cost/benefits in a structured way, the following method may be used. Users judge whether the benefits to be gained by using particular products are outweighed by the costs experienced (usually implicit such as effort or delay). Factors that are considered include ease of use, ease of learning, training, etc. It is worthwhile to try to be explicit about the costs and the benefits for particular user groups so attention can be paid to these during design. The method is most useful for systems which affect whole work processes rather than single-user, single-task products.

The method enables a realistic study of the costs and benefits of the new system across a range of user groups early in the development cycle, allowing new options to be considered.

The resources required are fairly small. The process requires input from people with knowledge of different user types in existing work process.

The stages in performing the user cost/benefit analysis are as follows:

Procedure

1. Each user group/stakeholder is identified (or drawn from **FORM 1.2**). For each group **FORM 2.7** is completed. There are two parts to the form as shown on the following pages. One part covers individual tasks or roles of the different users, while the other looks at characteristics of the job in general. Either or both parts of the form may be completed as appropriate.

2. Review the organisational process diagram (Section 1.10, Figure 7).

3. List the benefits for each user group in the 'benefit' column for each aspect of the system that would be regarded as a benefit, and rate its importance from +1 to +5.

4. List the costs for each user group in the 'cost' column for each aspect of the system that would be regarded as a cost, and rate its importance from -1 to -5.

5. Where costs appear to outweigh the benefits, try to think of a change to the system that would help to redress the balance. This may involve changing the system functions, the way they are implemented, or the organisational design in which the user is operating. Write this down in the sixth column: 'Modified or New User Requirement' column. An example of **FORM 2.7** is given below.

   Note: The ratings are intended to highlight the importance of the costs and benefits of individual factors. The scores for different factors should not however be added together as the scores for different factors may not be of an equivalent scale.

6. Summarise the costs and benefits at the bottom of the form, and try to assess how acceptable, to the user group, the whole system concept would be.

7. If more than one user group is being considered, complete **FORM 2.7** for each.

8. If necessary, identify how a more equitable spread of costs and benefits can be achieved for all user groups. Refer back to the organisational process diagram (section 1.10) and modify it to capture the changes of how each user group relates to each other. There is, of course, no guarantee that the requirements of different user groups will be compatible with each other.

FORM 2.7 - USER COST/BENEFITS (EXAMPLE)

2.7 User Cost/Benefits

User group: *Bank staff.*

ISSUE	BENEFITS	RATING +1 TO +5	COSTS	RATING –1 TO –5	MODIFIED OR NEW USER REQUIREMENT	REF.
Main roles or tasks:						
1. Day to day maintenance of bank machine						
Receive some form of warning when bank machine needs to be refilled with money and paper.	*Able to respond to faults before bank customers complain*	*+3*	*Interruption of warnings and having to keep checking bank machines.*	*-3*	*Rota of staff to check on level of paper and money in machine.*	*2.7.1*
Refill as required.			*Time consuming task*		*Develop faster refill mechanism.*	*2.7.2*
2. Dealing with faults						
Receive warning if machine develops a fault.	*Increased responsibility as able to sort out some problems locally.*	*+2*	*Time and effort*	*-2*		
Decide what type of fault has arisen. If minor fault correct oneself.	*Some broadening of skills*		*May be critic-ised if call out maintenance for minor problem*		*Develop diary of past faults to guide staff on correcting faults and when to call out an engineer*	*2.7.3*
If more major, contact maintenance to come out and repair	*Social and professional contact with another group.*					

Transfer to FORM 3.5 System Functions and features and FORM 3.4 Technical environment

2.7 User Cost/Benefits

User group: Help desk support engineer.

ISSUE	BENEFITS	RATING +1 TO +5	COSTS	RATING −1 TO −5	MODIFIED OR NEW USER REQUIREMENT	REF.
Job Security						
Job Content 1. Task variety 2. Effort required 3. New skills/skills lost 4. Work pace/deadlines 5. Workload 6. Satisfaction	*Increased range of tasks* *Fault correction.*	*+3* *+5* *—* *—* *—* *+1*		*-1* *-1* *0* *—* *—* *0*		
Organisational procedures 1. Discretion and autonomy 2. Standardisation and formality 3. Power and influence 4. Privacy 5. Communications 6. Status						
Personnel policies 1. Basic pay 2. Other rewards 3. Career prospects 4. Industrial relations						

Summary: *Generally an improvement to working procedures.*

Transfer to FORM 3.9 Social and Organisational Requirements

2.8 Move to phase 3 ?

The results of the prototype test and the task acceptability review are considered. If the prototype appears to be successful, the design can be used as a basis for the user requirements specification which is carried out in Phase 3.

Phase 3

User requirements documentation

PHASE 3 of the user requirements specification process consist of the third iteration around the user-centred design loop as shown below.

- **REQUIREMENTS:** This phase include stating requirements for the system in general including the general characteristics of the system and the organisational structure on which it is based. User goals and tasks, the technical environment and functions and features are also described. The user interface design approach and user support requirements are documented. The physical and social environment that will be created is described as well as a plan for testing the implemented system.

- **DESIGN:** To support the design phase, an implementation plan is produced. This will include what actions will be taken to make users aware of the new system, and what training is to be provided. If system phasing is required, it will include a description of the migration path from the user's point of view. Also it will include a list of internal guides or external standards that should be referred to during implementation.

- **TEST:** A plan will also be developed to specify how feedback on system usage, usability and acceptability will be collected while the system is in use.

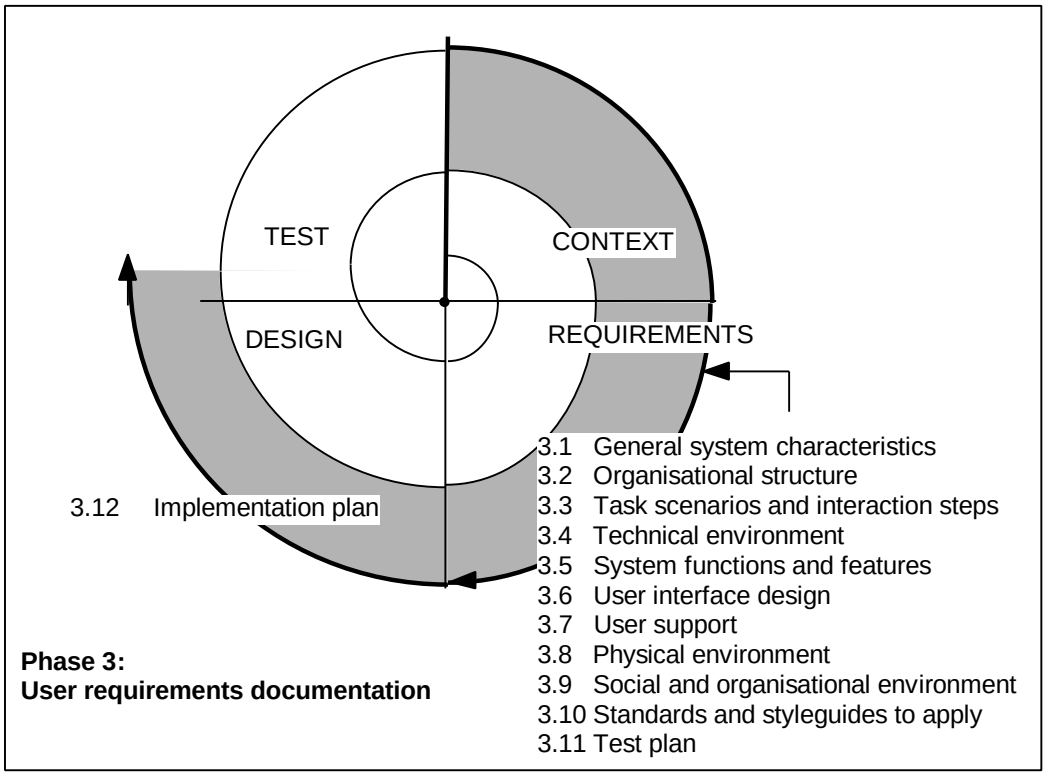

FIGURE 9. PHASE 3 - USER REQUIREMENTS DOCUMENTATION

The previous two stages will have produced a series of **potential user requirements** that should be considered as a basis for agreed user requirements which will form the basis of the user requirements specification.

The design team and user representatives should go through each of the requirements specified in Phases 1 and 2, copy each potential user requirement across to one of the following sections in **Phase 3**.

Once all these potential requirements have been copied into the following sections, they should be reviewed by the design team and users and the following stages carried out:

- Any duplicate requirements in any particular session should be removed.
- Any other requirements that do not seem needed or are superseded by other requirements should also be excluded.
- Any new requirements which arise from the review of each section below should be added.

Once the requirements have been carefully reviewed in this way, they should be written up as a Draft User Requirements Specification document. This may be structured using the headings presented in Stage 3 below. The document must then be reviewed by representatives of the development team with users in order to check or **validate the user requirements**. For any requirements that users are unsure about or which they think are missing, it will be necessary to go back (or trace) the relevant sections of the Framework and to clarify or add a requirement, as necessary.

The sections for compiling the user requirements, initially, are presented below.

Requirements prioritisation

It is important to prioritise the user requirements listed in the following tables. A column labelled 'PRI' within each of the following tables should be used for this. This prioritisation should be achieved by assembling a rating team comprising end users, marketing and industry personnel, managers, and technical designers. Once the user requirements have been assembled, the rating team will consider each requirement one at a time and assign a priority from 1 to 5 to each requirement, where 1 is 'critical' or 'vitally important' and 5 is 'unimportant'. If a rating as low as 5 is actually assigned to a requirement, consideration should then be given as to whether it can be deleted to simplify the specification.

Some items, e.g. the hardware to be used, are given as part of the system rather than as a requirement. These items are labelled 'CON.', to indicate that they are constraints on the design. Other items are simply statements of flexibility in the design e.g. the labelling and colour of keys requirement. These items are labelled 'Flex'.

For a more details about prioritising user requirements, see the QFD process (Fehin, 1997).

Requirements achievement

It is also important to monitor progress in achieving the user requirements as the design process continues. This should be done at meetings of the design team. For each requirement, the progress in achieving it should be estimated. Then either 1, 2 or 3 asterisks

are written into the column marked 'ACH'. One asterisk * represents some progress, two asterisks ** equals considerable progress, and three asterisks *** indicates that the requirement fully achieved. If no progress has been made, then the cell is left blank.

It is possible to estimate the current level of progress in achieving the user requirements by means of the following calculation:

$$\frac{\text{Sum of } (\text{ Each priority assignment, 1 to 5 } \mathbf{X} \text{ Progress level, 1 to 3 })}{(\text{Total of the priority assignments } \mathbf{X} \text{ 3})}$$

This value will be a proportion which can then be converted into a percentage by multiplying it by 100. Thus if each requirement is fully met (progress level 3), the current achievement level will be 100%. However in practice, not all requirements will be fully met, and the design team will be looking to achieve the highest percentage as possible. In setting some level of acceptability it may, for instance, be decided that all requirements with a rating of 3 or more must be fully met.

3.1 General system characteristics

Objective

This section lists the system characteristics (not functions) that have arisen during the analysis and concept stages. They may describe the appearance of the system, any performance requirements, customisation or flexibility features.

Process

1. Review all the potential user requirements identified in Phases 1 and 2, particularly in **FORMS 1.3 User characteristics, 1.7 User goals and tasks,** and **1.10 Design ideas and concepts**. Copy those that relate to system characteristics into **FORM 3.1** below.

2. Remove any requirements that duplicate others or do not seem relevant.

3. Add any new requirements which arise from the review of this section.

Please refer to RESPECT D4.2, section 3.4 which briefly describes the establishment of an in-house usability guide. This will be a natural extension to the process of specifying general and desirable characteristics of the system.

FORM 3.1 - GENERAL SYSTEM CHARACTERISTICS (EXAMPLE)

3.1 General System Characteristics			
System: *New Bank machine* ⬇ **Transfer from FORM 1.3 User Characteristics, 2.3 Task scenarios and FORM 1.10 Design Concept characteristics**	PRI. 1-5	ACH. * ** ***	REF.
Given particular consideration to older user groups who may be more reserved about new technology.	*2*	*	*1.3.1*
Use simple terminology, diagrams and pictures.	*3*	*	*1.3.4*
Design to be usable by people who may have limited reading skills.	*4*		*1.3.5*
Try to make the system conform with any accepted ad hoc standards for bank machines.	*3*	*	*1.3.8*
Use very supportive dialogues to make user feel comfortable.	*2*	**	*1.3.9*
Develop attractive interfaces which are also easy and enjoyable to use.	*1*	***	*1.3.10* *1.3.14* *1.3.15*
Ensure that results can be achieved quickly.	*2*	*	*1.3.16*
Make as attractive and simple as possible for new users.	*2*	*	*1.3.19*
Users representatives should be given the opportunity to see existing applications developed with same software.	*4*		*1.4.4*
Speech synthesis for guidance - Expensive but may be useful for those with impaired vision.	*4*		*1.11.1*
System response times should normally be <2 seconds, maximum 10 seconds.	*3*	*	
System should support the performance of tasks in variable order.	*3*	**	
Provide a good level of privacy for the user to avoid encouraging theft.	*2*	***	

3.2 Organisational structure

This section describes the intended organisational structure implied by the new system.

Drawing upon the design ideas of section 1.10, the new organisational structure for bank machine operation is as follows:

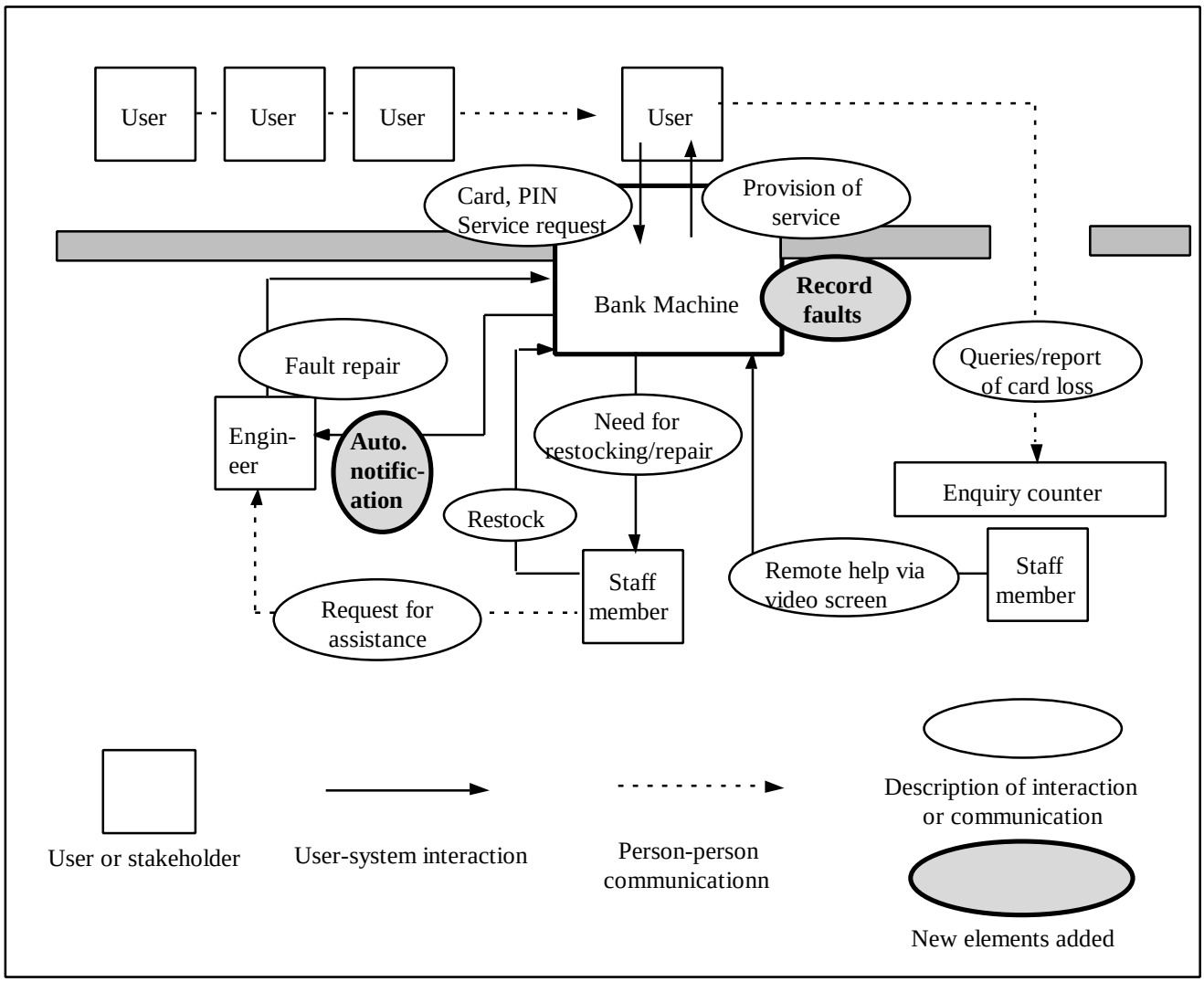

FIGURE 10. NEW SYSTEM ORGANISATIONAL STRUCTURE

3.3 Task scenarios and interaction steps

Objective

This section describes the interaction steps required to carry out key tasks defined by the task scenarios.

Process

1. List each Task Scenario in a copy of **FORM 3.3** below.
2. In the left hand column list the task steps to be performed.
3. In the right hand column write, for each task step, the interaction required.

FORM 3.3 - TASK SCENARIOS AND INTERACTION STEPS (EXAMPLE)

3.3	**Task Scenarios and Interaction Steps**
System : Bank machine **Scenario** : S2(1) The user inserts the card and enters the PIN. They request £200, when the limit is £50. ⬇ **Transfer from FORM 2.3 Task scenarios and FORM 2.4 New processes**	PRI: 2 ACH: *
USER	SYSTEM
User identifies themselves by inserting card. *User types PIN or presses finger on fingerprint pad.* *User selects 'withdraw cash'.* *User selects or enters amount up to the maximum.* *User selects 'exit from system'.*	*System reads card, whichever way card inserted and displays message prompt .* *If recognised, system displays options.* *System displays maximum amount that can be withdrawn and possible amounts.* *System responds that cash is ready to be dispensed and displays menu of other services.* *System returns card and dispenses cash.*

Screen printouts from the prototype system will help to illustrate these sequences.

3.4 Technical environment

Objective

This section specifies:

- the general characteristics of the software interface of the system, e.g. Windows 95 interface, or monochrome text-based display.
- the hardware interface of the system, e.g. touch screen input with voice output, hands-free telephone, and
- equipment that will be used alongside the system (e.g. hands free telephone, camera for video conferencing, printer, scanner, fax machine, etc.).

A different hardware interface may be provided to support each user group. At the first pass through the process, alternatives should be listed to record all the hardware and methods of interaction that may be used or considered during the design process.

Process

1. Review all the potential user requirements identified in Stages 1 and 2, particularly in **FORM 1.4 Technical environment**, and **FORM 2.2 Design constraints**. Copy those that relate to the software, hardware or other equipment that the user will interact with, into **FORM 3.4** below.
2. Remove any requirements that duplicate others or do not seem relevant.
3. Add any new requirements which arise from the review of this section.

FORM 3.4 - TECHNICAL ENVIRONMENT REQUIREMENTS (EXAMPLE)

3.4 Technical environment requirements			
System: *Public interface to bank machine* ⬇ **Transfer from FORM 1.4 Technical environment**	PRI.	ACH.	REF.
Main hardware required			
System will be based on standard bank machine terminal. - AT&T terminal V1360	*Con*	—	*2.2.1*
Ensure that system keyboard and screen are placed at a standard height.	*2*	***	*1.3.6*
Hardware should be robust .	*2*	***	*1.4.1*
Layout of interface elements and keypad should be consistent with existing conventions and standards.	*3*	**	*1.4.2*
Software environment			
Full colour and graphical screens to be built with 'kiosk builder'	*Con*	—	
IBM Public Terminal operating system will be used.	*Con*	—	*1.4.3*
Output hardware			
Screen size will be standard 10 inches, with a resolution of 640x480	*Con*	—	*2.2.2* *1.9.1*
Screen should be non reflective	*2*	*	
Input hardware			
Use easy input device: if a keyboard, larger keys, secondary means of identification.	*3*		*1.3.7*
Avoid having buttons too small (minimum 2.5 cm square)	*3*	*	*1.9.2*
There will be 8 screen buttons for soft keys, 4 down each side of the display	*Con*	—	*2.2.3*
There will be a keypad of no more than 16 keys.	*Con*	—	*2.2.4*
Configuration of keypad is flexible.	*Flex*	—	
Colour of keys is flexible	*Flex*	—	*2.2.5*
Keys should be operable by users wearing gloves.	*4*		
Standards to be applied			
ISO DIS 9241-14 Menu dialogues	*4*	*	
ISO DIS 9241-7 Display requirements	*4*	*	
ISO DIS 9241-4 Keyboard	*4*	**	
Communication/network capabilities			
Ethernet connection to area computer.	*Con*	—	
Other equipment			
Headphones available for visually impaired user to plug in to machine for speech instructions.	*4*	***	
Other characteristics			
System response times should be <2 seconds, and never more than 10 seconds.	*3*	*	
Card should give clear step-by-step instructions for main transactions which can be followed when using the machine.	*2*	**	*1.4.6*

3.5 System functions and features

Objective

This section lists the system functions and features that have arisen during the analysis and concept stages and by considering new tasks.

Process

1. Review all the potential user requirements identified in Stages 1 and 2, particularly in:
   - **FORM 1.8 Review current processes**
   - **FORM 1.9 Review other systems and products**
   - **FORM 1.10 Design ideas and concepts**
   - **FORM 2.4 Propose new processes**

   Copy those that relate to **system functions** and into **FORM 3.5.1** below, and those that relate to **system features** and into **FORM 3.5.2**.
2. Remove any requirements that duplicate others or do not seem relevant.
3. Add any new requirements which arise from the review of this section.

To perform a check of the system functions and features, a **functionality matrix (section 4.5)** can be used. This is essentially a matrix which matches proposed system functions across all user tasks. It provides a check as to whether any additional functions are required to support some tasks or whether any functions should be removed or amalgamated because they support the same tasks. The matrix can also act as documentation to demonstrate the need for all the functions included within the system.

FORM 3.5.1 - SYSTEM FUNCTIONS (EXAMPLE)

3.5.1 System Functions			
System: *New bank machine* **Transfer from FORM 1.8 Review current processes** **FORM 1.9 Review other systems and products** **FORM 1.10 Design ideas and concepts** **and FORM 2.4 Propose new processes**	PRI. 1-5	ACH. * ** ***	REF.
Provide security feature e.g. alarm button.	*3*	*	*1.3.17*
Bank machine could be adapted to automatically send a message to maintenance if a major failure occurs.	*4*	**	*1.4.5*

If user forgets PIN, system returns card for user to take into bank. If card not shown to bank within 5 days, it is cancelled and letter sent to customer with new card.	*4*		*2.6.1*
Allow PIN or thumbprint for verification.	*4*		*1.8.3 2.4.4*
If machine abandoned, provide quick reset button	*4*	**	*1.8.4*
Low warning before machine runs out of money	*2*	***	*1.9.3*
Low warning before machine runs out of paper	*3*	***	*1.9.4*
Allow user to notify bank if card gets stuck in the machine, by pressing special button - Good idea as long as button only active for when card is actually lost in the machine	*3*		*1.11.3*
System displays maximum amount that can be withdrawn.	*2*	***	*2.4.5*

FORM 3.5.2 - SYSTEM FEATURES (EXAMPLE)

3.5.2 System Features			
System: *New bank machine* **Transfer from FORM 1.8 Review current processes** **FORM 1.9 Review other systems and products** **FORM 1.10 Design ideas and concept** **and FORM 2.4 Propose new processes**	PRI.	ACH.	REF.
Use English language and up to 8 other language options, depending on local area.	*4*	*	*1.3.3*
Use highly supportive interface with clear logical structure and which are easy to learn and remember	*2*	**	*1.3.11* *1.3.13*
Use terms that the user will understand	*2*	***	*1.3.12*
Ensure design allows privacy when using machine.	*2*	*	*1.3.18*
To assist user in orienting card properly provide notch on card and *and picture on machine as guidance*	*3* *3*	*** ***	*1.8.1* *1.8.2* *2.4.2*
Display amount available that can be withdrawn from the account.	*2*	***	*1.8.6*
Provide cancel button if user decides not proceed with transaction.	*4*		*1.8.7*
Provide short cuts to commonly required options.	*3*	*	*1.8.8*
If user does not receive expected money or no receipt returned, provide button to register problem at given time. Allow them to report the fault by video camera.	*3*	*	*1.8.9* *2.6.2*
Provide camera on bank machine to record possible threatening behaviour to user by passers-by	*4*		*1.8.10*
Question and Answer mode for beginners - Useful idea but may slow down the interaction process	*4*	*	*1.11.2*
Develop reader that will read card whichever way it is inserted.	*3*	**	*2.4.1*
System offers options: 'Withdraw £20, £50, £100' on first menu.	*3*	***	*2.4.6*
Allow user to select : *'Cash with receipt' or 'Cash without receipt'*	*2*	***	*2.4.7*
Machine gives out of service notice before next card inserted.	*2*	**	
Exit button provided.	*3*	*	
Fast path available giving standard amount of money	*4*		
User asked if happy to continue withdrawing cash without receipt.	*3*	**	
Help button which user can press to register problem. Reference number displayed which can be quoted inside bank.	*3*		
Develop faster refill mechanism.	*2*	**	*2.7.2*

3.6 User interface design

Objective

This section specifies details of the user interface that the system will offer to support each user group. This may include:

- an outline of the user interface structure, based on the system concepts
- example screens.

It should also record further details of the software that will be used to create each user-system interface (e.g. development package to be used, standards containing screen layout templates, etc.).

Process

1. Review all the potential user requirements identified in Stages 1 and 2, particularly in **FORM 1.6 Technical characteristics and 2.2 Design constraints.** Copy those that relate to the software interface into **FORM 3.6** below.

2. Remove any requirements that duplicate others or do not seem relevant.

3. Add any new requirements which arise from the review of this section.

Please refer to RESPECT D4.2, section 2.2 which describes formal characteristics of user interface specifications. Section 2.3 describes a set of principles for user interface specifications based on the concepts of task adequacy, self-descriptiveness, controllability, conformity to user expectations, error tolerance, flexibility and learnability. Section 2.4 presents an outline for documenting the user interface.

FORM 3.6 - USER INTERFACE DESIGN (EXAMPLE 1)

3.6 User Interface Design			
Interface name: *1. Customer interface to bank machine* ⬇ **Transfer from FORM 1.4 Technical environment** **Transfer from FORM 1.10 Design ideas and concepts**	PRI.	ACH.	REF.
Purpose *This is the interface by which members of the public will obtain services from the bank machine.*			
Features *Cash limit display.* *Indication of date of balance*	*3* *3*	** **	*1.8.6*
Components of interface *Main screen structure shown in document: 'Kiosk style guide v2'.*	*Con*	—	
Standards to be used *ISO DIS 9241-14 Menu dialogues*	*4*	*	
Example screens *Screen templates shown in document: 'Proposed user Interface Style guide', section 1*	*Con*	—	
Tools to be used for building interface *Borland C++ and Screen Builder*	*Con*	—	

FORM 3.6 - USER INTERFACE DESIGN (EXAMPLE 2)

3.6 User Interface Design			
Interface name: *2. Bank staff interface to bank machine* ⬇ **Transfer from FORM 1.4 Technical environment** **Transfer from FORM 1.10 Design ideas and concepts**	PRI.	ACH.	REF.
Purpose *This is the interface by which bank staff will maintain the machine and refill it with materials.*			
Methods of interaction *Full keyboard, function keys and monochrome screen.*	*Con*	—	*1.8.6*
Components of interface *Main screen structure shown in document: 'Proposed user Interface Styleguide', section 2*	*Con*	—	
Standards to be used *ISO DIS 9241-14 Menu dialogues.*	*4*	*	
Example screens *Screen templates shown in document: 'Kiosk style guide v2''.*	*Con*	—	
Tools to be used for building interface *Borland C++ and Screen Builder.*	*Con*	—	

3.7 User Support

Objective

Specification of the support that will be needed for each user group, including training, documentation, help facilities, local experts, telephone hot lines etc. The methods by which these support services are provided may also be specified.

Process

1. Review all the potential user requirements identified in Stages 1 and 2, particularly in **FORM 1.3 User characteristics**. Copy those that relate to User support requirements into **FORM 3.7** below.

2. Remove any requirements that duplicate others or do not seem relevant.

3. Add any new requirements which arise from the review of this section.

FORM 3.7 - USER SUPPORT REQUIRED (EXAMPLE)

3.7	**User Support**			
User Group: *Bank staff* ⬇ **Transfer from FORM 1.3 User characteristics**		PRI.	ACH.	REF.
User training	*A 3 day training course will be provided on the basic facilities within the system.* *This will be topped up by one hours training per week for the following two months.*	*1* *2*	** *	
User document-ation	*Two manuals required:* *- Introductory guide (less than 20 pages)* *- Reference manual.*	 *2* *3*	 *** *	
Help facilities	*The system will provide an on-line help system that describes how to perform each of the main tasks that users need to perform.* *The system functions will also be presented in alphabetical order with synonyms. Hypertext links will be provided between different help sections.*	*3* *4*	* **	
Local expert	*A local expert will be given 3 weeks training and it will form part of his or her job to answer questions from other staff.*	*2*	***	
Telephone hot lines	*A telephone enquiry line will be provided to answer questions that cannot easily be solved by reference to help or the local expert.*	*2*	*	
Other support				

3.8 Physical environment

Objective

This section lists the physical requirements that have arisen during the analysis and concept stages and by considering new tasks.

Process

1. Review all the potential user requirements identified in Stages 1 and 2, particularly in **FORM 1.5 Physical environment**. Copy those that relate to physical requirements into **FORM 3.8** below.

2. Remove any requirements that duplicate others or do not seem relevant.

3. Add any new requirements which arise from the review of this section.

FORM 3.8 - PHYSICAL ENVIRONMENT (EXAMPLE)

3.8 Physical Environment			
System: *Bank machine* ⬇ **Transfer from FORM 1.5 Physical environment**	PRI.	ACH.	REF.
Equipment should work in the following conditions: *Temperature -10c to +40c* *Humidity 55% - 90%, Rainfall*	 *2* *2*	 ** **	*1.5.1*
Any use of auditory feedback or output may be drowned by street noise . Provide some form of earpiece or volume control is available.	*4*		*1.5.2*
Sighting of machine should be optimum to avoid glare where possible. Screen filters and matt screen surfaces should be tested to see if they reduce potential problems. Needs to be luminescent for use in the dark.	*3*	*	*1.5.3*
Bank machine should be mounted 1m. above ground, inset in wall.	*3*		*1.5.4*
Bank machine should be reachable by at least 80% of wheelchair users both in terms of height and posture when operating the machine.	*3*	*	*1.5.5*
Ensure machine is clearly visible and signposted for people trying to locate it.	*3*	*	*1.5.6*
Position bank machine in the open and with extra lighting to maximise safety.	*3*	*	*1.5.7*
Keys should be operable by users wearing gloves.	*4*		*1.5.8*
Provide more privacy with surround on machine	*2*	***	*1.8.5*

3.9 Social and Organisational environment

Objective

This section lists the organisational requirements that have arisen during the analysis and concept stages and by considering new tasks.
Note that this part is different from the specification of organisation and business requirements for the system but may be aided by results from that analysis.

Process

1. Review all the potential user requirements identified in Stages 1 and 2, particularly in **FORM 1.6 Social and Organisational environment**. Copy those that relate to social and organisational requirements into **FORM 3.9.** below.
2. Remove any requirements that duplicate others or do not seem relevant.
3. Add any new requirements which arise from the review of this section.

FORM 3.9 - SOCIAL AND ORGANISATIONAL ENVIRONMENT

3.9 Social and Organisational Environment			
System : *Bank machine* ⬇ **Transfer from FORM 1.6 Social and Organisational environment**	PRI.	ACH.	REF.
Staff should be prepared to give advice to the public on using bank machines. Counter staff should always be available to handle similar transactions if person does not wish to use a bank machine.	*2*	*	*1.6.1*
Bank machines should be monitored for response speeds and number of transactions per day.	*3*	*	*1.6.2*
Staff should be able to interrupt queue to perform a quick check if query arises about quality of output.	*3*	**	*1.6.3*
Allow two users to view/access bank machine comfortably.	*4*		*1.6.4*
Assistance is not normally available at external bank machine. User needs way of registering problem and to request help soon afterwards.	*3*	*	*1.6.5*
Customer needs a way of abandoning transaction if they cannot proceed and feel under pressure.	*4*		*1.6.6*
Bank machine may provide an alarm bell to signal help required if theft takes place.	*3*	**	*1.6.7*
Bank machine should provide sufficient barriers to prevent others from seeing the transaction.	*2*	***	*1.6.8*
Rota of staff to check on level of paper and money in machine.	*2*	**	*2.7.1*
Allow bank staff to repair some faults to save visit from maintenance staff - Worth considering. Danger of overloading staff.	*2*	**	*1.11.4*
Develop diary of past faults to guide staff on correcting faults and when to call out an engineer.	*3*	*	*2.7.3*

3.10 Standards and styleguides to apply

Objective

In this section, identify relevant Human Factors standards to which the new system, the support facilities and working conditions must conform e.g. ISO 9241 parts 11, 14 and 15, or 'must use OSF (Open Systems Foundation) Motif'. In addition, internal standards or styleguides e.g. for user interface design, should also be identified. Conformance to both external and internal standards will become additional user requirements.

Process

To assist in selecting appropriate standards, carry out the following steps:

1. Read through the descriptions of different Human Factors standards that may be applied to system design in the following section. Identify those which are relevant to the system being developed and record them in **FORM 3.10**.

2. Use the additional blank section of the form to record any **internal standards or style guides** the standards that should apply to the system, or any **external standards not listed below**. You also need to decide whether the standard is to be used as guidance only or whether adherence to it will be part of the testing process.

This section lists the relevant standards to which the new system, the support facilities and the working conditions must conform. Conformance to them will become additional user requirements.

Process

Review the Human factors standards described in part D, Appendix 2. Copy those that seem most relevant into **FORM 3.10** as shown below.

FORM 3.10 - STANDARDS AND STYLEGUIDES TO APPLY (EXAMPLE)

3.10 Standards and styleguides to Apply				
System: *New bank machine*				
STANDARD	**USER REQUIREMENT (METHOD OF APPLICATION)**	**PRI.**	**ACH.**	**REF.**
2. ISO 9241-1: 1993, Ergonomic requirements for office work with visual display terminals (VDTs) - General Introduction.	*To be read by design team members.*	*2*	***	
4. ISO /DIS 9241-11 : Guidance on Usability	*To be read by design team members.*	*4*	*	
13. ISO DIS 9241-5 Workstation layout and postural requirements	*Check during design process that requirements are met.*	*4*	*	
15. ISO DIS 9241-7 Display requirements with reflections	*Check during design process that requirements are met.*	*4*	*	
21. ISO/DIS 9241-14: Menu dialogues	*To be used as general guidance.*	*4*	***	
26. ISO/IEC DIS 11581-1 : Icon symbols and functions - Part 1: Icons - general	*Ensure that all standard symbols related to application are complied with.*	*4*	*	
30. ETSI ETR 116 Human Factors guidelines for ISDN terminal design.	*To be read by design team members.*	*4*	*	
40. NCR Style guide on bank system design.	*All user interfaces must conform to this standard.*	*2*	***	

3.11 Test plan

Objective

This section provides guidance in testing the feasibility of the system concept and to generate possible changes to it. The assessment may also help in selecting the most suitable design concept from a number of competing concepts.

Process

Controlled experiments can be used to test the usability of a prototype or of a simulation of the future system concept. Representative users are asked to perform tasks with the prototype to help check whether the system concept will be an acceptable basis for the new system. Specific elements of the system design may be tested as required; for example input devices and icon sets might be examined.

The results will also help in generating usability goals for the future system when it is developed fully.

For more information about **controlled testing** see Part C, section 4.2.

The basic method is as follows:

1. Define the sample of users (number in sample and their characteristics) that will perform the tasks. To do this refer to the **list of users** (section 1.2) and the **user characteristics** (section 1.3).

2. Define the list of tasks that they will carry out, referring to the **task scenarios** developed in section 2.3.

3. Define the conditions that will be set up for running the trials. To identify relevant conditions, refer to characteristics of the **physical environment** (section 1.5), **organisational environment** (section 1.6) and **technical environment** (section 1.4).

4. Define suitable measures to assess each task performed with the system e.g. task achievement, task time, satisfaction or ease of use ratings, and how data will be gathered to take the measurements.

5. Define how much introduction will be given to the users before the trials and how help will be offered during the trials. Record these in a table similar to **FORM 3.11.1** below. When an operational version of the system is ready, the user trials will be run using it.

6. Run the user trials capturing performance measures, rating scale scores, and user comments about the system for each task. These results will be recorded in a table similar to **FORM 3.11.2** below.

7. Following the user test, the results will be reviewed and the system design will be modified to address any problems identified. The results will also be used as a baseline for testing the system when it is implemented.

FORM 3.11.1 - USABILITY TEST PLAN (EXAMPLE)

3.11.1	**Usability test plan**
User sample and characteristics	*Include equal numbers of males and females in user trials 1.3.2*
Method of recruitment	*Advertise for bank machine users through local newspapers.*
Tasks or Scenarios ⬆ **Based on FORM 1.7 user goals or FORM 2.3 Task scenarios**	*S1(1) Access bank machine and withdraw £50 in cash and collect receipt.* *S2(1) Access bank machine and withdraw £200 when limit £50* *S3(1) Access bank machine, request £100 then stop transaction* *S8(4) Access bank machine and request £100. Report that incorrect amount dispensed.*
Measures/Method of data capture	*Task achievement. - Observation* *Time to complete task. - Observation/stop watch* *Comparison with expert user - Video analysis* *Comments pre and post system. - Interview* *Ratings of ease /difficulty of task. - Survey form and prompt cards*
Test equipment	*Panasonic 9000 video recorder.* *Panasonic video camera.* *Notepads and pens.* *Stopwatch.*
Environment ⬆ **based on FORMS:** **1.4 Technical environment** **1.5 Physical environment** **1.6 Social and organisational environment**	*Simulation of bank foyer.* *Bank machine set-up which user interacts with.* *Desk manned by member of bank staff (experimenter) who will provide support as required.*
Introduction to subject	*Brief introduction to the new bank machine.* *Five minutes of guided practice exploring the basic facilities.*
Assistance to be given	*No help to be given unless the user is unable to proceed.*

FORM 3.11.2 - USABILITY TEST RESULTS (EXAMPLE)

3.11.2 Usability test results

Results based on user test with of 20 bank customers.

Tasks	Task achievement	Mean Time and Standard deviation	Comparison with expert task time	Subjective rating of difficulty (1=easy 7=difficult)	Comments
S1(1) Withdraw £50	*80%*	*Mean: 4 mins SD: 25secs*	*75%*	*2.2*	*All users carried out task fairly easily. .*
S2(1) Withdraw amount> cash limit	*90%*	*Mean: 3 mins SD: 45 secs*	*60%*	*4.6*	*Some users not sure of level of cash limit .*
S3(1) Access bank machine then halt transaction	*60%*	*Mean: 2 mins 30 sec SD: 15 secs*	*75%*	*3.2*	*most users able to exit easily. Some users expected quit key to be red not yellow.*
S8(4) Access bank machine and report incorrect money dispensed.	*80%*	*Mean: 8 mins SD: 15 secs*	*75%*	*6*	*Most users unsure how to report this problem through the machine. Most reported problem to member of bank staff.*

3.12 Implementation plan

The user requirements specification is complemented by the **implementation plan**. This defines how the user requirements that have been specified will be carried forward into the system design process. The plan can also be used to demonstrate to customers how user requirements have been dealt with after the system has been developed.

If product phasing is required, a statement of the migration path from the user's view is also produced. This will specify:

- The system configuration needed to match end user needs.
- The organisational changes necessary for users to gain maximum benefit.
- The planning stages to implement the system.

The section headings recommended for the user requirements plan and the type of information it should contain are shown in the table below (**FORM 3.12**).

FORM 3.12 - USER REQUIREMENTS IMPLEMENTATION PLAN (EXAMPLE)

3.12	**User requirements implementation plan**
System: *New bank system*	
1. Design rationale *The system will be developed iteratively, starting with identification of the basic concepts for the interface. A number of simulations will be identified as a basis for the overall concept.* *The simulations that seem to have the most potential will be developed as interactive prototypes for user discussion and walkthrough. Finally the approach which receives the most user support will be taken forward and a full prototype will be created as a basis for the full system.* *Formal user trials will be carried out on the prototype to test if it meets the usability and acceptability goals set for it.*	
2. Form of prototype *Specify what form the system prototype will take: for example, paper, operational.*	
3. Usability test method *Specify the test(s) and test method(s) to be employed with the prototype(s) to examine usability, e.g. expert walk-through, diagnostics, user trials, etc. Where user trials are proposed, specify the subjects to be employed (e.g. number, ages, genders etc.). Add the results of usability testing.*	
4. Acceptability test method *Specify the test(s) and method(s) by which the product will be tested for acceptability e.g. user cost/benefits assessments, user trials, etc. Where field trials are proposed, specify the field site. Add the results of acceptability testing.*	
5. Scope for iteration *Specify the opportunities and constraints for re-testing usability and acceptability.*	
6. User audit *Specify how feedback on product usage, usability and acceptability will be collected. Add results of user audits.*	
7. System phasing *Specify how the system will be introduced. Describe the order in which different modules will be introduced.* *Also describe how users will be made aware of and introduced to the new system and in general terms what training will be provided.*	

Part C

4. User Requirements Methods

This section provides descriptions of each of the methods that may be used to support relevant stages of the user requirements specification framework. The material draws principally from the RESPECT Methods deliverable D3.1 (Kirakowski and Vereker, 1996).

Each method contains the following headings:

- **What Is The Method, And When Can It Be Used?**
  This provides a summary of what happens when the method is applied, and an indication of when it can be used.
- **Typical Application Areas**
  This describes the kinds of systems (e.g. a software package) or application areas (e.g. office systems) in which the method might be applied.
- **Benefits and Limitations**
  First the benefits are listed and then the limitations. When a method is employed it must be ensured that each limitation is either not relevant or has been catered for.
- **What you need**
  A description is given of the resources required to carry out the method in terms of people and resources such as equipment, rooms, etc.
- **Process**
  The key activities that should be carried are listed in the order they are to be performed.
- **Practical guidelines**
  These give practical guidance for applying the method, based on experience and special considerations for applying them to different categories of user (e.g. people with impairments, the young).
- **Further Information**
  This section provides references to sources for further information.

Methods covered include:

4.1 Brainstorm
4.2 Controlled testing
4.3 Diary keeping
4.4 Focus groups
4.5 Functionality matrix
4.6 Group discussion
4.7 Interviews
4.8 Observation
4.9 Paper prototyping
4.10 Parallel design
4.11 Rapid prototyping
4.12 Scenario building
4.13 Storyboarding
4.14 Survey
4.15 Task analysis
4.16 Task allocation
4.17 Video prototyping
4.18 Walkthrough
4.19 Wizard of Oz prototyping

Relating the Methods to the User Requirements Framework

The following table provides guidance on the selection of methods for use in the main stages of the User Requirements Framework (Part B). Selection is based on a number of factors. These include:

- The phase in the process of specifying user requirements
- The time and effort required to apply the method
- The expertise or skills required
- The equipment or facilities required
- The particular strengths of the method.

Note also that the RESPECT deliverable D6.2 'Requirements specification and evaluation for user groups with special needs', describes how to relate some of the above methods to users with impairments and disabilities (visual, hearing, motor and cognitive), as well as elderly and young users. The methods covered include: group discussions, interviews, observation, surveys, controlled testing.

Characteristics / *Methods*	Applicable to Framework stage: 1. User context & Early design	2. Proto-type and User test	3 User Req Document-ation	Time and Effort required.	Expertise or Skills required.	Equipment Facilities required.	Particular Strengths
1. Brainstorm		√		Low	Group motivater.		Generating ideas
2. Controlled testing		√		Medium	Subject handling. Planning.	Quiet area or lab., VCR and camera.	Identifying interaction problems
3. Diary keeping	√			Medium			Captures day-to-day usage.
4. Focus group	√	√		Low	Group chairing.		Discuss topic in-depth
5. Functionality matrix		√	√	Low	System knowledge	Spreadsheet software	Refining list of functions.
6. Group discussion	√	√	√	Low	Group chairing.		Airing issues.
7. Interview	√			Low	Neutral non-leading		Individual opinions in-depth.
8. Observation	√			Medium	Event recording.		Seeing real situation
9. Paper prototyping		√		Medium	Ability to present and capture ideas.		Quick way to test out ideas for system.
10. Parallel design		√		High	Group man -agement		Provides design ideas.
11. Rapid prototyping		√		Medium	Openness. Software development	Rapid development software.	Tests outs interactivity of system.
12. Scenario building		√		Medium	Group chairing.		Way of testing concept.
13. Storyboarding		√		Medium	Group chairing. Drawing.		User appreciati'n of concept.
14. Survey	√			High	Survey design and analysis.	Data handling package.	Mass collection of opinion.
15. Task analysis	√			Medium	Interview-ing.		Understand-ing current work in depth.
16. Task allocation		√		Medium	Interview-ing.		Establishing basis for satisfying jobs.
17. Video prototyping		√		High	Editing. Interview-ing		Way of presenting concept realistically
18. Walkthrough		√		Medium	Group chairing.		careful check of system design.
19. Wizard of Oz		√		High	Technical. Testing.	Simulation equipment linked together.	Way to testing advanced concepts.

TABLE 1. COMPARISON OF USER REQUIREMENTS METHODS

4.1 Brainstorming

What Is The Method, And When Can It Be Used?

Brainstorming is one of several group methods, probably the oldest and best known. The idea is to bring together a set of experts to inspire each other in the creative, idea generation phase of the problem solving process. Brainstorming is used to generate new ideas by freeing the mind to accept any idea that is suggested, thus allowing freedom for creativity. The method has been widely used in design. The result of a brainstorming session is hopefully a set of good ideas, and a general feel for the solution area. 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 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.
It does not take long to obtain useful data and the session need not take more than one hour.

Limitations

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

What you need

The human resources are the most important for succeeding with this method. The more creative people with a variety of experiences in the field, the better the result. In the range of 5 - 12 people may participate.

Process

1. Decide on the objectives of the brainstorm and the participants required to take part in it.
2. When contacting the participants explain clearly what topics are to be considered and the meeting format. Also 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 discussion leader should be active in leading the discussion, and summing up the results at the end of each topic. It is important to distinguish

between what is the consensus of the group, and what is the opinion of different participants.

Practical guidelines

See the guidance provided in section 4.6 for **group discussion**.

Additional rules are also provided by Cross (1989):

- Do not allow any criticism of ideas during the session.
- Encourage a large quantity of ideas.
- Welcome seemingly irrelevant ideas.
- Keep all ideas short and precise.
- Try to combine and improve on the ideas of others.

Further information

Osborn (1963), Jones (1980).

4.2 Controlled testing

What Is The Method, And When Can It Be Used?

Controlled experiments can be used to test the usability of discrete elements of a prototype or simulation of a future system. Representative users are asked to perform tasks with the prototype to help clarify the details of a user requirements specification. The elements should be capable of being tested separately from the product as a whole; for example input devices and icon sets might be examined.

Typical Application Areas

Useful for testing user behaviour in response to a prototype.

Analysis Methods

User testing often results in a wealth of information. This must be systematically analysed to clarify user requirements. It is important to establish the events that will be identified and analysed before embarking on controlled testing.

Benefits

User trials will indicate how users will react to the real system when built.

Provides experimental evidence to show the problems that users might envisage with the future system.

Enables the design team to compare existing products as a way of considering future options.

Limitations

If the trials are too controlled, they may not give a realistic assessment of how users will perform with the system.
Can be an expensive method, with many days needed to set up the trials, test with users and analyse results. Its inputs to the design process may be too slow for the intended timescales.

What you need

The method of controlled testing depends on recruiting a set of suitable subjects as users of the system. It also requires equipment to run the system and record evaluation data. This may include a video camera, video recorder, microphones, etc. Also it will be necessary to set up the tests in a suitable environment such as a laboratory or quiet area.

Process

To prepare for controlled testing, the following items are required:

1. The description of the test tasks and scenarios.
2. A simple test procedure with written instructions.
3. A description of usability goals to be considered and criteria for assessing their achievement (see section 3.9).
4. A predefined format to identify problems.
5. A debriefing interview guide.
6. A procedure to rank problems.
7. Develop or select data recording tools to apply during the test session (e.g. observation sheets) and afterwards (e.g. questionnaire and interview schedules).
8. Distribution of testing roles within the design team (e.g. overseeing the session, running the tests with the user, observation, controlling recording equipment etc.).
9. Estimate of the number of subjects required (possibly firmed up after the pilot test session).

Once trials are run, data is analysed, and problem severity is prioritised in an implications report.

Practical guidelines

- Conduct the tests in an informal and friendly atmosphere.
- Allow enough time between sessions for overruns.
- Make arrangements for telephone calls to be taken by someone else rather than interrupting the session.
- Make it clear that it is the system being tested.
- Make it clear beforehand how much subjects will be paid for the whole session. Avoid flexible payment based on time spent in the session.
- Avoid prompting the user too much during the session.
- If the user does get completely stuck, do not force them to continue but help them or move on to the next task.

Further information

Lindgaard (1994), Maguire (1996), Nielsen (1993).

Refer to RESPECT deliverable D6.2 for further information on performing controlled tests involving users with impairments and disabilities, as well as elderly and young users.

4.3 Diary keeping

What Is The Method, And When Can It Be Used?

Activity diaries require the informant to record activities they are engaged in throughout a normal day. Diaries may vary from open-ended, where the informant writes in their own words, to highly structured tick-box forms, where the respondent gives simple multiple choice or yes/no answers to questions. The required materials range from paper and pencil techniques, to video tape diaries and on-line input forms administered by computer.

Typical Application Areas

Useful to capture user behaviour over a period of time.

Benefits

Allows data to be captured about every day tasks, without researcher intrusion.

Limitations

Users may forget to complete their diary or fail to complete it properly if insufficient instruction is given.

What you need

Production of diaries and visits to help maintain and bolster user efforts to complete their diaries. Computer-administered formats and video-cassette recording for more sophisticated data capture.

Process and guidelines

1. Decide on whether the diaries are to be free form allowing the person to express themselves freely, or structured with fixed response formats which will be easier to analyse.
2. If a structured method is used, then a careful selection of questions and response categories must be produced.
3. Decide how often the respondents should complete the diary e.g. hourly, daily, weekly. This will be determined by the nature of the data that needs to be captured

and the tasks being carried out. Alternatively the user may need to complete the diary immediately after a particular event has occurred.
4. Produce copies of the diary and clear instructions for completing them.
5. Provide a means (e.g. a telephone number) whereby the respondent can check on the diary keeping procedure.

Further information

Poulson et al (1996).

4.4 Focus group

What Is The Method, And When Can It Be Used?

A focus group brings together a cross-section of stakeholders in a discussion group format. Views are elicited by a facilitator on relevant topics. Meetings can be taped for later analysis. Useful early in requirements specification. Helps to identify issues which may need to be tackled and provides a multi-faceted perspective on them.

Typical Application Areas

Useful to consider particular questions of user need or design options.

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.

What you need

Meeting facilities and audio/video recording facilities if a record of the session is desired.

Process

1. 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.

2. 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'.

3. 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.

4. 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.

5. 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.

Practical guidelines

See guidelines provided for group discussions.

- Since the group is focusing on a set of concepts, make sure that the discussion stay on the topics of the meeting.
- If possible explain the concepts to be explored using slide shows, storyboards or other vehicles for embodying aspects of the system or product.
- Provide several alternatives to emphasise the point that there is more than one possible solution and to stimulate discussion about common themes, gaps and problems.

Further information

Blomberg, Giacomi, Mosher and Swenton-Hall (1993)
Macaulay (1996), Preece (1994), Poulson et al (1996), Caplan (1990).

4.5 Functionality matrix

What Is The Method, And When Can It Be Used?

This process specifies the system functions that each user will require for the different tasks that they perform. The most critical task functions are identified so that more time can be paid to them during usability testing later in the design process. The form below (**FORM 4**) shows

the structure for a functionality matrix. It is important that input from different user groups is obtained in order to complete the matrix fully.

Typical Application Areas

This method is useful for systems where the number of possible functions is high (e.g. in a generic software package) and where the range of tasks that the user will perform is well specified. In these situations, the functionality matrix can be used to trade-off different functions, or to add and remove functions depending on their value for supporting specific tasks. It is also useful for multi-user systems to ensure that the tasks of each user type are supported.

Functionality Matrix **System:**						
	Functions					
Users and Tasks	**F1**	**F2**	**F3**	**F4**	**F5**	Key: ● = Critical to task ○ = Occasional use
User A						Comments
Task A	●					
Task B			○			
Task C					○	
User B						
Task A		●				
Task B			○			
Task C				○		
Function selection						

FIGURE 11. STRUCTURE FOR FUNCTIONALITY MATRIX

Benefits

- It can be tailored to suit varying design processes and in-house styles.
- It allows different user types to be considered together in a single process.
- Superfluous functions are identified.
- It represents a reference in subsequent product lifecycle stages and may be updated in the light of prototyping.

Limitations

- The prime focus is on functions and features rather than interface appearance.
- It can be cumbersome for large numbers of functions.

What you need

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

Process

1. Identify user groups (or take from FORM 3) and enter into matrix rows.
2. Identify tasks per user group and enter into matrix rows.
3. List potential functions and features and enter into matrix columns.
4. Identify functions which are critical to task.
5. Identify functions which are only occasional used.
6. Add new functions or features as required to support gaps in tasks.
7. Remove functions which are not required.
8. Develop prototypes to help create more detailed user requirements specification.

Practical guidelines

- Start off with a simple high level version of the matrix, to get an overview of the main tasks and related groups of functions. This will help to define the scope of the matrix and help to keep it within manageable proportions.
- Use a spreadsheet package to manage large functionality matrices. A print out from this spreadsheet can then be used for group discussion.

Further information

Catterall (1990).

4.6 Group Discussion

What Is The Method, And When Can It Be Used?

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.

Typical Application Areas

Useful for obtaining opinions efficiently from a range of people.

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.

What you need

Requires preparation on the part of the group chair to make sure that the meeting focuses on the issues at hand.

Process

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. Also 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 discussion leader should be active in formulating the themes for the discussion, and summing up the results at the end of each topic. It is important to distinguish between what is the consensus of the group, and what is the opinion of different participants.

Practical guidelines

- Create a good atmosphere.
- 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.
- Prevent destructive behaviour on the part of specific participants.
- 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.

Further information

Maculae (1996), Poulson et al (1996).

Refer to RESPECT deliverable D6.2 for information on running discussion groups involving users with impairments and disabilities, as well as elderly and young users.

4.7 Interviews

What Is The Method, And When Can It Be Used?

Commonplace technique where domain experts or less experienced users 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

Useful for obtaining in-depth data about a particular role or set of tasks. Also useful to obtain detailed feedback on a design option.

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 possess from their experience of social conversation. It is characterised by 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 popular, well known and widely accepted and are useful for investigating events which occur infrequently.

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.

What you need

Requires considerable preparation on the part of the interviewer.

Process

In an elicitation context, the semi-structured interview is generally most fruitful.

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. Also referred to as the relaxing phase, where summaries may be given as to what has taken place. Subsequent actions are noted, and future planning is made.

Practical guidelines

- If the interview is to be conducted in a structured manner, the questions should be constructed and tested in the same way as for a survey.
- Before the interviews, 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.
- Conduct the interview in a friendly but businesslike way.
- Be consistent in how you pose the questions between interviews.
- Allow the respondent time to elaborate their answer before moving on to the next stage.
- Avoid leading the respondent.
- Be prepared to diverge form the standard questions and probe further if an interesting line of discussion develops (in line with the aims of the interview).
- 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.
- For hearing impaired users, allow them to complete a survey form equivalent to the interview.

Further information

Preece (1994), Macaulay (1996).

Refer to RESPECT deliverable D6.2 for information on performing interviews involving users with impairments and disabilities, as well as elderly and young users.

4.8 Observation

What Is The Method, And When Can It Be Used?

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 recorder.

Typical Application Areas

Useful early in specification for obtaining qualitative data. It is useful for studying currently executed tasks and processes.

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

Observation can be obtrusive and subjects may alter their behaviour due to the presence of an observer. Co-operation of users is vital, 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.

What you need

Time for analysis of observation results.
Indirect observation requires access to audio visual recording and playback equipment.

Process

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.

Practical guidelines

- Make sure that those being observed are aware of the reason for your study (i.e. to assist them) and do not see you in negative terms. This is particularly important for mentally impaired and blind users who may be disturbed by a passive presence that they are not sure about.
- Run a pilot observation session to get a feel for what to expect and to test out any observation sheets. This will also help to judge how long the observation session needs to be. If the session involves informal activities with members of the public, they may take the opportunity to have conversations with the observer. Make sure that there is enough time to allow these interactions to take place.
- Try to be as unobtrusive as possible.
- Note down any events that you do not understand and try to clarify them with the user as soon as the session is completed.
- Try to be aware of the range of influences that are affecting the user.
- If possible photograph the users work area or the area of operation as this will act as a reminder of the environmental context.
- After your observations, write down your first impressions before the analysis stage later on.

Further information

Preece (1994).

Refer to RESPECT deliverable D6.2 for information on running observation sessions involving users with impairments and disabilities, as well as elderly and young users.

4.9 Paper prototyping

What Is The Method, And When Can It Be Used?

This method features the use of simple materials and equipment to create a paper-based simulation of an interface or system with the aim of exploring user requirements. (Later in the lifecycle, paper prototypes provide a valuable and cost-effective means to evaluate design options. Interface elements such as menus, windows, dialogues and icons are created using paper, card, acetate, and 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 in front of the 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 recording can be made. Costs

may also be incurred when recruiting users and allocating time to manage each evaluation session.

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 a range of user requirements and general system concepts can be explored easily and quickly.

Benefits

- Communication and collaboration between designers and users is encouraged.
- Paper prototypes are quick to build and refine.
- 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 exploration of fine detail.
- Due to the use of paper and a human operator, this form of prototype cannot 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.

What you need

Only simple materials are required to create the elements of the prototype. These include paper, acetate, pens and adhesives. 'Post-its' may also be used to represent interface elements such as system messages, input forms and dialogue boxes. A video camera may also required to record the paper-based interactions. The user session may employ two evaluators, one to manipulate the paper interface elements, acting as the system, and the other to control the session.

Process

The following procedure may be adopted for implementing this method:

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 pieces 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 user works 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 during the session for later scrutiny.
10. Conduct post-session interviews with the user, drawing upon pre-set questions and issues raised during the prototype evaluation.
11. Debrief and thank the user.
12. Analyse information obtained, summarise observations and user evaluations. Consider the themes and severity of the problems identified.
13. Summarise implications for user requirements and feed back to the design team. Video recordings can support this.
14. Where necessary, refine the paper prototype and repeat the above process.

Practical guidelines

- Work through the paper-based interactions as fully as possible and try to cover the different paths that users may wish to follow.
- Have spare paper, Post-its, etc. available to simulate new paths of interactions that the user would expect to make.
- Practice the possible interactions with a pilot user to make the interactive process as slick as possible.

Further information

Nielsen (1991), Rettig (1994).

4.10 Parallel design

What Is The Method, And When Can It Be Used?

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 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 take 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 a very cheap way of exploring the range of possible system concepts and selecting the probable optimum.

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.
- Requires a lot of time over a short period for the design work to be carried out.
- Time is also needed to compare parallel design outputs properly so that the benefits of each approach are obtained.

What you need

The method requires design team members to be available concurrently in order to carry out design work in parallel. Briefing notes are also needed to make sure that the designers are given the same information are start the design from the same starting point.

Process

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 designs 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.

Practical guidelines

- Make sure that all the design teams are given the same information before starting the design activity.
- Decide beforehand how much time to allocate to the design work. However if one team completes their work first be flexible in allowing others to come to their conclusions.
- Allow design teams to use whatever media they prefer in order to present their designs.

Further information

Nielsen (1993)

4.11 Rapid prototyping (software or hardware based)

What Is The Method, And When Can It Be Used?

This method is concerned with developing different proposed concepts through software or hardware prototypes, and evaluating them. In general the process is termed 'rapid' 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. Later on in the lifecycle, it can also be used to specify details of the user interface to be included in the future system.

A major difficult in system design is getting the concept itself across to the users. Whereas a rapid prototype can demonstrate the system concept well, paper prototyping methods may be less effective.

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 process and graphical interface elements for defining the software's logical structure and 'look-and-feel'. 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.

Many tools exist for producing rapid prototypes ranging from a sequence of Microsoft PowerPoint screens, to script based programming systems such as HyperCard, Toolbook and Visual Basic which can help to create a software prototype. Thus 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. A member of staff is also required to direct users if the prototype is to be evaluated formally.

What you need

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

Benefits

- Gives users (especially the general public) a tangible demonstration of what the system is about.
- 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.

Process

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.

Practical guidelines

- Avoid spending too long on the development of the first prototype as uses may require substantial changes to it. Similarly try not to put too much effort into particular features (e.g. animations) which may not be required.
- Avoid making the prototype too polished as this may force users to accept it as finished.
- Avoid putting in features that will raise the users expectations but which are unlikely to be achieved with the real system (e.g. too fast response times, too sophisticated graphics).

Further information

Maguire (1996), Preece (1994), Wilson, J. and Rosenberg, D. (1988).

4.12 Scenario building

What Is The Method, And When Can It Be Used?

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 in the early phases of a development cycle to generate end user requirements and usability aims. 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 may also be discussed with users to establish how they would like or not like to interact with the system (in general terms).

Benefits

- It 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 not appropriate for considering the details of interface design and layout.

What you need

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.

Process

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.

In terms of output, the method encourages a deeper understanding of user requirements and can be specifically used to plan subsequent user-based evaluations.

Practical guidelines

- Try to generate scenarios to cover a wide range of situations, not just the most common ones or those of most interest to the design team.
- Try to include problem situations, that will test the system concept not just straightforward scenarios.
- Work through the scenarios fully and judge the system on that basis rather than trying to change the system half way through.

Further information

Clark (1991), Nielsen (1991).

4.13 Storyboarding

What Is The Method, And When Can It Be Used?

Storyboards (also termed "Presentation Scenarios" by Nielsen) are sequences of images which demonstrate the relationship between individual events (e.g. screen outputs) 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 possible structures, functionality and navigation options available. The storyboard can be shown to colleagues in a design team as well as potential users, allowing others to visualise the composition and scope of possible interfaces and offer critical feedback.

Few technical resources are required to create a storyboard. Simple drawing tools (both computer and non computer-based) are sufficient. Storyboards provide a platform for exploring user requirements 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 requirements.

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.

Limitations

- Storyboards can lack the interactive quality of other prototyping methods although interactive storyboarding systems are available. Madsen & Aiken (1993), for instance, report the use of an interactive HyperCard based system.
- Because of their simplicity, storyboards do not support the evaluation of fine design detail.
- Storyboards do not accurately convey system response times.

What you need

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.

Process

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 will indicate 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 requirements 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.

Practical guidelines

- Produce high level drawings to represent the system. Details on drawings may distract the users or side-track the discussion.
- It may not be necessary to produce many drawings covering every stage of the interaction. Using a single drawing as a basis it is possible to discuss the events taking place before and afterwards.
- Try to include higher level organisational aspects in the drawings (e.g. interactions between people) rather than just showing user interactions with the system interface. This will give the users a fuller picture of the system concept and allow them to foresee socio-technical mismatches.

Further information

Madsen and Aiken (1993), Nielsen (1991) and Preece (1994).

4.14Survey

What Is The Method, And When Can It Be Used?

A survey involves administering a set of written questions to a large sample population of users. Surveys can help determine information or customers, 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

Useful for obtaining quantitative data from users about existing tasks or the current system.

Benefits

Quick and relatively inexpensive to administer but not to design. Results can be subjected to statistical analysis.
Provides ‘hard’ data to supplement more subjective, qualitative information such as unstructured opinions.

Limitations

Survey design is not straightforward; many think they can do it but few laymen do it well - experienced guidance is needed. It may be hard to follow up on interesting comments as it is often not desirable or possible to keep records of respondents.

What you need

Depends very much on the complexity of the survey and the number of respondents needed.

Process

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

2. 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.

3. 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.

Practical guidelines

- Explain the aim of the survey at the beginning of the form. If it is a postal survey include a polite covering letter explaining its purpose. If the respondents can see the reason for completing the survey, they are more likely to do so.
- If the survey is to be returned by post, make it easy to return by providing a pre-addressed envelope (preferably stamped).
- Avoid too many open-ended questions as this will increase the analysis effort required.
- Make sure that multiple choice answers are mutually exclusive.
- Make sure that the question wording is clear and concise.
- Avoid double questions in a single question.
- Avoid questions involving negatives.
- Avoid complex branching structures.
- Avoid asking questions that make users feel uncomfortable or offend. For example it may be better to ask for selection of an age band rather than exact age.
- For visually impaired users provide the survey either in large print form, Braille, or allow the respondent to provide their responses via a face to face interview or over the phone.
- Provide mentally impaired users the option to respond to the survey via an interview.
- Motor impaired users may have trouble filling in the survey. The provision of large completion boxes will assist them. Again they may prefer to be interviewed.
- Young users may be able to answer the questions but not be able to read. They may therefore also require the survey to be conducted as an interview.
- For young users able to read, make sure that the questions are simple and understandable.

Further information

Preece (1994).

Refer to RESPECT deliverable D6.2 for information on performing surveys with users who have impairments and disabilities, as well as elderly and young users.

4.15 Task Analysis

What Is The Method, And When Can It Be Used?

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 task. A detailed task analysis can be conducted to understand the current system and the information flows within it. These information flows are important to the maintenance of the existing system and must be incorporated or substituted in any new system. Failure to allocate sufficient resources to this activity increases the potential for costly problems arising in later phases of development. Task analysis makes it possible to design and allocate tasks appropriately within the new system. The functions to be included within the system and the user interface can then be accurately specified.

Typical Application Areas

Suitable and recommended for most situations.

Benefits

Provides knowledge of the tasks that the user wishes to perform. Thus it is a reference against which the value of the system functions and features can be tested.

Limitations

Formal task analysis can be time consuming and produce much data requiring considerable effort to analyse.

What you need

It is important to gain access to real users to discuss their current or possible future tasks as well as user representatives.

Process

The process in this section is divided into two phases:

1. High level task decomposition, where major tasks are broken down into sub-tasks. This step provides a good overview of the tasks being analysed.
2. Task flow diagramming, where specific tasks are divided into the basic task steps.

Both of these phases are described in the following sections.

Practical guidelines

- Produce a 'map' of relevant users and try to understand initially their main tasks or roles.
- Identify individual people who will be able to provide correct information about the tasks that are performed. Timetable meetings to make sure that all these users can be included in the analysis.
- If necessary plan observation sessions to get a richer picture of the tasks or task problems.
- Gather as much information as possible and then try to structure it soon afterwards while it is still fresh in the memory.
- During the structuring process go back to any users to clarify your understanding.
- Try to supplement textual descriptions of tasks with diagrams such as task decompositions or task flow diagrams.

Task decomposition

The aim of 'high level task decomposition' is to decompose the high level tasks and break them down into their constituent subtasks and operations. This will show an overall structure of the main user tasks. At a lower level it may be desirable to show the task flows, decision processes and even screen layouts (see task flow analysis, below)

The process of task decomposition is best represented as a structure chart (similar to that used in Hierarchical Task Analysis). This shows the sequencing of activities by ordering them from left to right. In order to break down a task, the question should be asked 'how is this task done?'. If a sub-task is identified at a lower level, it is possible to build up the structure by asking 'why is this done?'. This approach can be summarised as shown below.

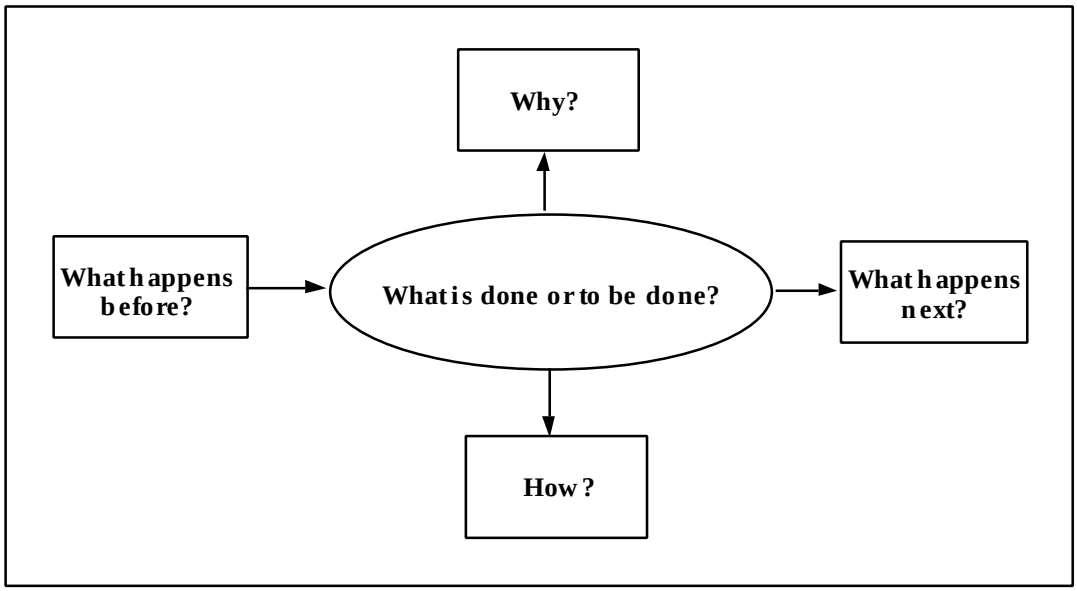

FIGURE 12. PROCESS OF TASK DECOMPOSITION

The task decomposition can be carried out using the following stages:

1. Identify the task to be analysed from the Task list (**section 1.4**).
2. Break this down into between 4 and 8 subtasks. These subtasks should be specified in terms of objectives and, between them, should cover the whole area of interest.
3. Draw the subtasks as a layered diagram ensuring that it is complete.
4. Decide upon the level of detail into which to decompose. Making a conscious decision at this stage will ensure that all the subtask decompositions are treated consistently. It may be decided that the decomposition should continue until flows are more easily represented as a task flow diagram.
5. Continue the decomposition process, ensuring that the decompositions and numbering are consistent. It is usually helpful to produce a written account as well as the decomposition diagram.
6. Present the analysis to someone else who has not been involved in the decomposition but who knows the tasks well enough to check for consistency.

The following structure chart represents a hierarchical decomposition of a function into its components. It shows the sequencing of activities by ordering them from left to right. Activities that may be repeated a number of times (iteration) are indicated by a small asterisk in the box. When one of a number of activities may be chosen (selection) a small circle is included in the box. A horizontal line in a box can be used to indicate no action, for example where the user can choose to take an action or not. Where appropriate a sub-task or group of sub-tasks can be represented as a flow diagram. The circles labelled 'TF1' and 'TF2' show the links to task flow diagrams.

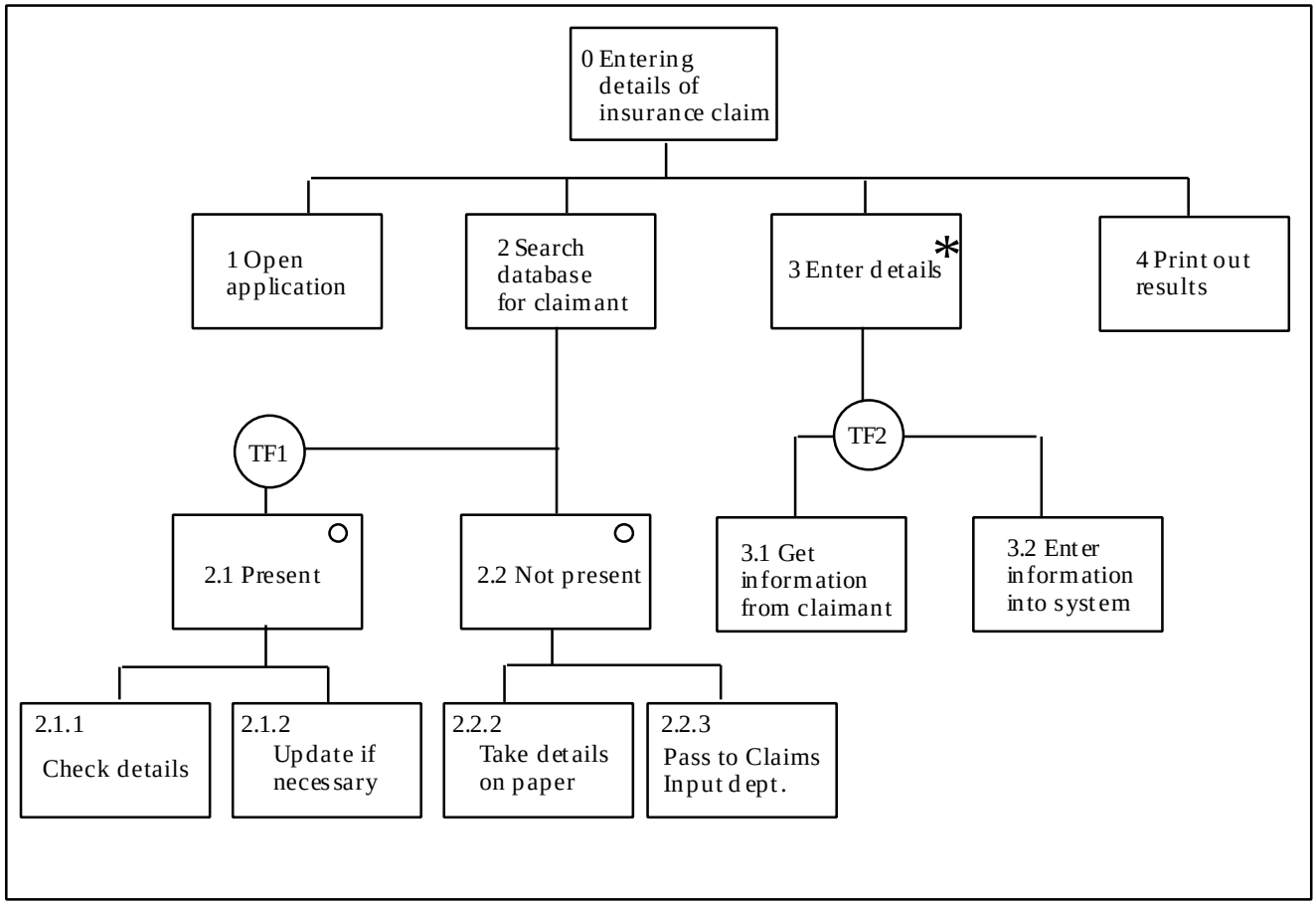

FIGURE 13. TASK DECOMPOSITION DIAGRAM

As previously stated, it is important that the requirements definition process does not simply record existing operations but also assesses the likely changes resulting from the introduction of new systems and facilities. The implications of likely changes should be recorded on the task decomposition diagrams themselves. This can be done by the use of comments at the bottom of the diagram with reference to the task decomposition box numbers (e.g. 2.1.1 or 2.1.2).

Task flow diagrams

Task flow analysis will document the details of specific tasks. It can include details of interactions between the user and the current system, or other individuals, and any problems related to them. Copies of screens from the current system may also be taken to provide details of interactive tasks. Task flows will not only show the specific details of current work processes but may also highlight areas where task processes are poorly understood, are carried out differently by different staff, or are inconsistent with the higher level task structure.
An example task flow chart is shown below. 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.

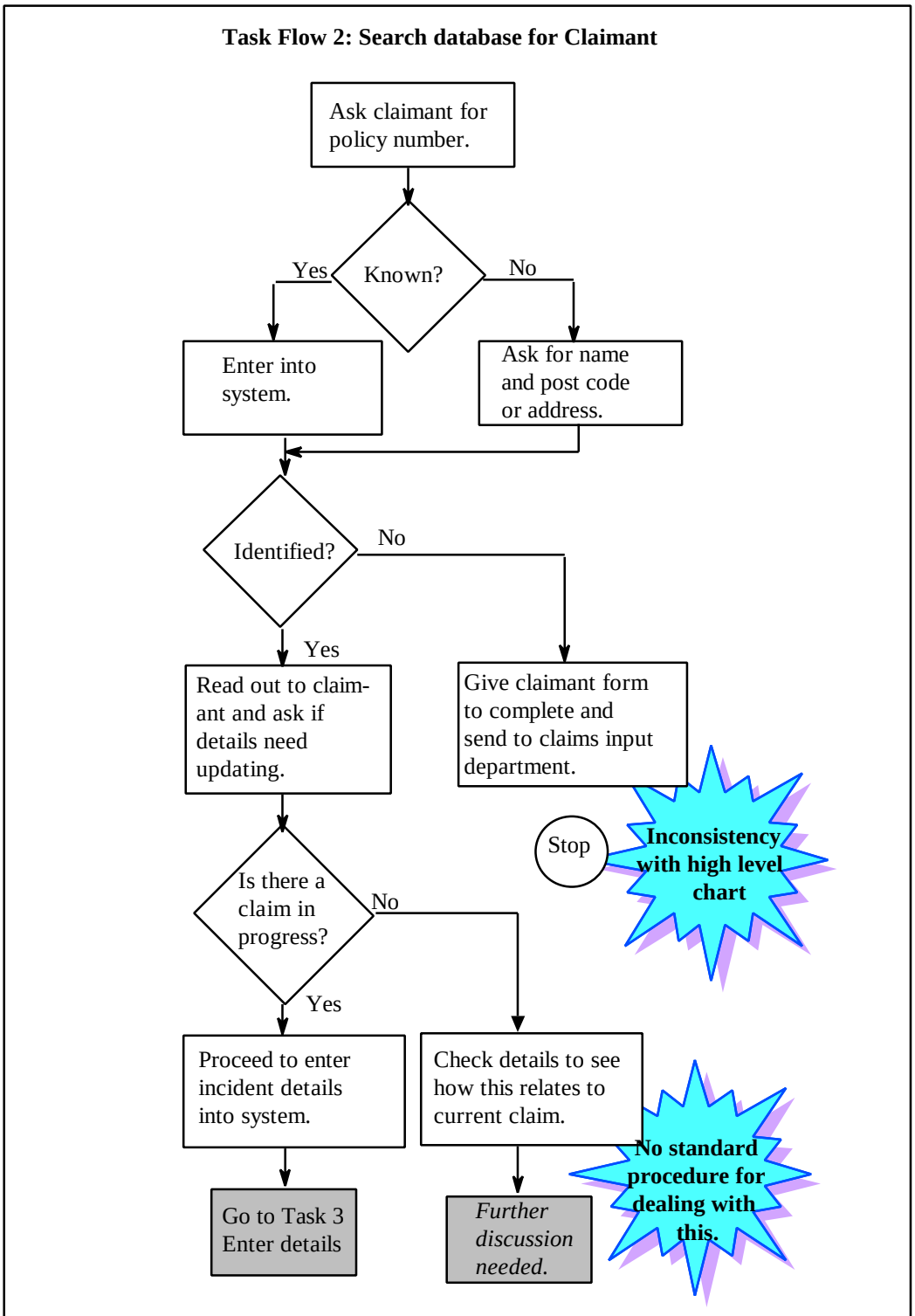

FIGURE 14. TASK FLOW CHART

Further information

Preece (1994), Shepherd (1985, 1989).

4.16 Task Allocation

What Is The Method, And When Can It Be Used?

A successful system depends on the effective allocation of tasks between the system and the users. Different task allocation options may need to be considered before specifying a clear system boundary. A range of options are established to identify the optimal division of labour, to provide job satisfaction and efficient operation of the whole work process. The approach is most useful for systems which affect whole work processes rather than single user, single task products.

An important activity which integrates the system function identified with the organisational design (section 3.2) is the allocation of function. The objective of task allocation is to specify which functions should be carried out by the system and which by the users. This technique can also be used to identify elements of the user interface.

Task allocation decisions determine the extent to which a given job, task, function or responsibility is to be automated or assigned to a human. The decisions are based on many factors, such as relative capabilities and limitations of human versus technology in terms of reliability, speed, accuracy, strength, flexibility of response, cost and the importance of successful or timely accomplishment of tasks.

The goal of the design team is to allocate functions to the human, to the system and to the human-computer system in order to achieve effective, efficient and satisfying results. It is frequently a decision which designers make unconsciously rather than by deliberation, yet it establishes the framework within which relevant jobs and tasks will be done by users. Designers may be tempted to identify which functions the technology is capable of performing and then simply allocate the remaining functions to users, relying on their flexibility to make the system work. This does not make best use of users abilities and skills and can lead to unsatisfactory job design. Generally, it is better to involve users or their representatives in this decision and ensure that the resulting human functions form a meaningful set of tasks.

Benefits

- It 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

- It requires some concept of the new system for users to contribute to the process and generate new options.

What you need

The analysts will need good understanding of existing job roles.

Process

To apply task allocation charts, the following steps are performed:

1. Each task or work process is identified.
2. Identify those for which there is some possibility of different task allocations.
2. For each task, a flow diagram is drawn up to show the existing split of tasks between users and computers, and the interactions between them.
3. Try to generate at least two charts (i.e. options for allocating tasks) to show task allocation between user(s) and computer for the new system.
4. Comment on the implications for job satisfaction and efficiency for each chart and annotate it accordingly.
5. Select the chart which is most acceptable to users, or generate new charts if necessary.

Practical guidelines

- Consider all the tasks and identify those for which there is some possibility of different task allocations.
- Try to develop more than one task allocation option to stimulate discussion within the design team or between users.
- Do not spend too long trying to produce neat diagrams. Rough drawings of possible allocations are adequate.

Reference

Ip, Damodaran, Olphert and Maguire (1990).

FORM - TASK ALLOCATION (EXAMPLE)

Task Allocation Chart (Example 1)

Task: *Bank machine error report*
Version: *2*

System: Bank machine	User: Member of staff	User: Engineer	System requirement
State: Error warning displayed – If mechanical fault, send call automatically to engineer ➡ – If money or receipt paper has run out display signal. ➡	➡ **END** Refill with money or receipt paper. **END**	⬇ Receives signal. Rings bank to arrange visit to fix fault. ⬅	
Advantages: Immediate call to engineer. Speeds up process of fault reporting.	**Advantages:**	**Advantages:**	**Advantages:**
Disadvantages:	**Disadvantages:** No chance to investigate problem first. May feel controlled by engineer.	**Disadvantages:** Feel controlled by machine.	**Disadvantages:**

Task Allocation Chart (Example 2)

Task: *Bank machine error report*
Version: *2*

System: Bank machine	User: Member of staff	User: Engineer	System requirement
State: Error warning displayed ➡	⬇ Check if money or receipt paper running out. If so refill. **END** If engineering fault, call engineer. ➡	Respond to call. Agree time to visit. **END**	
Advantages:	**Advantages:** Has control over the error process.	**Advantages:**	**Advantages:**
Disadvantages:	**Disadvantages:** May not see signal quickly so service may be out of operation for some time	**Disadvantages:**	**Disadvantages:**

4.17 Video prototyping

What Is The Method, And When Can It Be Used?

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.

Application areas

There is 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.

What you need

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.

Process

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.

Practical guidelines

- Do not produce too much detail in the prototype as this can distract the users or side-track the discussion.
- Presentation packages can be used to simulate sequences of screens quickly and easily.
- If possible show the user on the video, as well as just system interactions.

Further information

Young and Greenlea (1992), Vertelney, L (1989).

4.18 Walkthroughs

What Is The Method, And When Can It Be Used?

A walkthrough is a process of going step by step through a system design getting reactions from relevant staff, typically 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

Benefits

- Obtains reactions to a design in an informal manner.
- Flexible means of obtaining reactions, allowing the users' discussion to range over issues not originally considered.

Limitations

- Requires some form of prototype to show and 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.

What you need

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.

Process

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, people's comments for transcription later on.
3. Select appropriate users to take part in the walkthrough, trying to cover the range of users within the target population.
4. Pilot the walkthrough to work out how much time is needed for each session.
5. Ensure recording facilities are available and functioning.
6. Conduct the walkthrough sessions, making sure that all sessions cover the issues identified beforehand.
7. Analyse information obtained by issues raised and comments made. Try to determine how many users made the same comment. Consider the themes and severity of the problems identified.

Practical guidelines

- Give an introduction to the participants in the walkthrough, explaining the aims of the session.
- Describe beforehand the range of tasks being covered.
- Allow enough time to discuss different aspects in depth.
- Explain that the participants should not be afraid to criticise the system.
- Encourage all participants to express their opinions.
- Consider audio recording the session and transcribing the information afterwards. This will allow the walkthrough to proceed more steadily.

Further information

Maulsby, Greenberg and Mander (1993), Nielsen (1993).

4.19 Wizard of Oz prototyping

What Is The Method, And When Can It Be Used?

Wizard of Oz is a technique used to present advanced concepts of interactions to users. In essence an expert (the wizard), possibly located behind a screen, processes input from a user and emulates system output. The aim is to demonstrate computer capabilities which cannot be done by the computer, for technical reasons or lack of resources. For example, the expert may

be someone behind the screen typing in responses to speech inputs from the user and thus pretending that the machine can understand this speech - just like the 'Wizard' in the eponymous film.

It is highly applicable to “intelligent interfaces” which feature agents, advisors and/or natural language processing.

Benefits

- This method allows user requirements and usability 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 method requires a higher commitment of resources than other approaches to prototyping such as those that rely on simple paper-based materials.

What you need

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.

Process

The general procedure for implementing this method 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.
4. Develop the prototype itself.
5. Select appropriate users to test the prototype, trying to cover the range of users within the target population.
6. Prepare realistic task scenarios for the evaluation.
7. Pilot the evaluation procedure and ensure the Wizard is well practised in playing the role of the computer.

8. Ensure recording facilities are available and functioning.
9. Conduct each session. The facilitator instructs the user to work through the allocated tasks interacting and responding to the system as appropriate.
10. Conduct post-session interviews with the users, drawing upon questions and issues raised during the use of the prototype.
11. Debrief and thank the user.
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. Video recordings can support this.
14. Where necessary refine the prototype and repeat the above process.

Practical guidelines

- Explain to the users beforehand the Wizard of Oz idea and that the system is being operated by another person. This will avoid users later feeling they are being tricked by the evaluators.
- Practice beforehand to make sure that the Wizard can keep up with the user. If necessary slow the user down.
- Keep the session short to avoid tiring the Wizard too much.

Further information

Maulsby, Greenberg and Mander (1993), Nielsen (1993).

Part D

References

Bevan N. and Macleod M. (1994) Usability measurement in context. Behaviour and Information Technology, **13**, 132-145.

Bevan N., Bowden R., Corcoran R., Curson I., Macleod M., Maissel J., Rengger R., Thomas C., Dillon A., Maguire M., Sweeney M. (1996) *Usability Context Analysis - a practical guide*, v4.02, NPL Usability Services, National Physical Laboratory, Queens Road, Teddington, Middlesex, TW11 0LW.

Bevan N and Azuma M (1997) Quality in use: Incorporating human factors into the software engineering lifecycle. In: Proceedings of the Third IEEE International Software Engineering Standards Symposium and Forum (ISESS'97), p169-179.

Bevan N (1997) Quality and usability: a new framework. In: Achieving software product quality, van Veenendaal, E, and McMullan, J (eds) Tutein Nolthenius, Netherlands

Bevan N (1998) Achieving Quality in Use through User Centred Requirements Engineering. RESPECT working paper (in preparation).

Beyer, H. and Holtzblatt, K. (1997) *Contextual design - Defining customer centred-systems*, SanFrancisco, CA: Morgan Kaufmann series in Interactice Technologies.

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

Caplan S. (1990) Using focus group methodology for ergonomic design. *Ergonomics*, 33.5, 527-533.

Catterall B. (1990) The HUFIT functionality matrix. *Human-Computer Interaction - INTERACT'90 conference proceedings,*, 27-31 August, (Diaper D., Cockton G., Gilmore D. and Shackel B., eds). Amsterdam: North-Holland, pp377-382, ISBN 0-444-88817-9.

Clark L. (1991) The use of scenarios by user interface designers. *HCI'91 conference proceedings,*, 20-23 August, (Diaper D. and Hammond, N., eds). Cambridge: Cambridge University Press, pp103-115, ISBN 0-521-416949-9.

Cross N. (1989) *Engineering design methods*, pp37-38. Chichester, Wiley. ISBN 0-471-92215-3

Daly-Jones, O, Thomas, C, Bevan, N. (1997) *Handbook of user centred design.*. EC Telematics Applications Programme, Project IE 2016 INUSE, NPL Usability Services, National Physical Laboratory, Queens Road, Teddington, Middlesex, TW11 0LW, UK.

Damodaran, L. (1996) User involvement in the systems design process - a practical guide for users, behaviour and Information technology, 15, 6, 262-377.

Eason K. D. (1988) *Information Technology and Organisational Change* . London: Taylor and Francis. ISBN 0-85066-388-1.

Fehin, P. (1997) Using QFD as a framework for a user-driven participatory design process, In: Achieving software product quality, Erik van Veenendall and Julie McMullan (eds), UTN Publishers, Den Bosch, The Netherlands, 69-82.

Fowler C. (1991) Usability Evaluation - Usability in the product lifecycle, *Usability Now! Newsletter*, Issue 3, Spring 1991, pp6-7. HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, LE11 1RG, UK.

Hartson, H. R. and Hix, D. (1989) Towards empirically derived methodologies and tools for HCI development. International Journal of Man-Machine Studies, 31, 477-494.

Hartson, H. R., Soichi, A. C., and Hix, D. (1990) The UAN: A user-oriented representation for direct manipulation interface designs. ACM Transactions on Information Systems, 8, 3, 181-203.

Heim, J., Endestad, T., Sketjne, J. H., Maguire, M. C., and Vereker, N. (1997) Requirements specification and evaluation for users groups with special needs, EC Telematics Applications Programme, Project TE 2010 RESPECT, WP6 Deliverable D6.2.

Beyer, H. and Holtzblatt, K. (1998) Contextual Design — Defining Customer-Centered Systems. San Francisco: Morgan Kaufmann.

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. *Human-Computer Interaction - INTERACT'90 conference proceedings*, 27-31 August, (Diaper D., Cockton G., Gilmore D. and Shackel B., eds). Amsterdam: North-Holland, pp 289-294, ISBN 0-444-88817-9.

ISO (1997a) *ISO 9241 - Ergonomics requirements for office work with visual display terminals (VDTs)*, 17parts. International Standards Organisation.

ISO (1997b) *ISO DIS 13407 User-Centred Design*, ISO Draft Standard. International Standards Organisation.

Jones, J.C. (1980) *Design methods: seeds of human futures.* Chichester, Wiley. ISBN 0-471-44790-0.

Kirakowski J. (1993) *SUMI Questionnaire*, HFRG, Department of Applied Psychology, University College Cork, Ireland.

Kirakowski J. and Vereker N. (1996) *Methods for user-oriented requirements specification*, EC Telematics Applications Programme, Project TE 2010 RESPECT, WP3 Deliverable D3.1, Human Factors Research Group, University College Cork, Department of Applied Psychology, Ireland, April.

Kirwan, B. and Ainsworth L.K. (eds) (1992) *A guide to task analysis* . London: Taylor and Francis, ISBN 0-748-400583.

Lindgaard G. (1994) *Usability testing and system evaluation - a guide for designing useful computer systems*. London, Chapman and Hall Computing Series, ISBN 0-412-46100-5.

Macaulay L.A. (1996) *Requirements Engineering;*. Berlin: Springer Verlag Series on Applied Computing, ISBN 3-540-76006-7.

Madsen K. and Aiken P. (1993) Experiences using co-operative interactive storyboard prototyping. *Communications of the ACM*, June, 36.4, 57-64.

Maguire M. (1996) *Prototyping and evaluation guide*. HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, LE11 1RG, UK. October.

Maguire M. (1997) *User Interface Design Guide*. EC Telematics Applications Programme, Project IE 2016 INUSE. HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, LE11 1RG, UK.

Maulsby D., Greenberg, S. and Mander R. (1993) Prototyping an intelligent agent through Wizard of OZ; *Human-Computer Interaction - INTERCHI'93 conference proceedings*, Sponsored by ACM SIGCHI and IFIP TC.13, 24-29 April; Addison-Wesley, pp277-284; ISBN 0-201-58884-6.

Nielsen J. (1991) Paper versus computer implementations as mock up scenarios for heuristic evaluation. *Human-Computer Interaction - INTERACT'90 conference proceedings,*, 27-31 August, (Diaper D., Cockton G., Gilmore D. and Shackel B., eds). Amsterdam: North-Holland, pp315-320, ISBN 0-444-88817-9.

Nielsen J. (1993) *Usability Engineering*, London: Academic Press (AP Professional), ISBN 0-12-518405-0.

Osborn A. F. (1963) *Applied Imagination*, New York: Schribeners and Sons.

Poulson D., Ashby M., and Richardson S. (eds) (1996) *USERfit A practical handbook on user-centred design for assistive technology*, Handbook produced within the European Commission TIDE programme USER project. HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, LE11 1RG.

Preece J., Rogers Y., Sharp H., Benyon D., Holland S. and Carey T. (1994) *Human-Computer Interaction*. Reading MA: Addison-Wesley. ISBN 0-201-62769-8.

Rettig M. (1994) *Prototyping for tiny fingers*. Communications of the ACM, April, 37.4, 21-27.

Shackel B. (1991) Usability - context, framework, definition, design & evaluation. In *Human Factors for Informatics Usability*, (Shackel B. and Richardson S. J., eds.); pp21-37, Cambridge: Cambridge University Press. ISBN 0-521-36570-8.

Shepherd A. (1985) Hierarchical task analysis and training decisions, *Programmed Learning and Educational Technology*, 22, 162-176.

Shepherd A. (1989) Analysis and training in information technology tasks, In *Task Analysis for Human-Computer Interaction*, (Diaper, D. ed), pp15-55; Chichester: Ellis Horwood. ISBN 0-745-80721-6.

Shneiderman, B. (1987) Designing the User Interface: Strategies for Effective Human-Computer Interaction, Reading, MA: Addison-Wesley. See also second edition, 1992.

Sommerville, I. and Sawyer, P. (1997) Requirements Engineering – A Good Practice Guide. Chichester, UK: Wiley.

Suttcliffe, A. G. (1991) Integrating methods of human-computer interface design with structured systems development. International Journal of Man-Machine Studies, 34, 631-655.

Taylor B. (1990) The HUFIT Planning, Analysis and Specification Toolset, *Human-Computer Interaction - INTERACT'90 conference proceedings,*, 27-31 August, (Diaper D., Cockton G., Gilmore D. and Shackel B., eds). Amsterdam: North-Holland, pp371-376, ISBN 0-444-88817-9.

Vertelney L (1989) *Using video to prototype user interfaces.* ACM SIGCHI Bulletin, 21.2, 57-61.

Vossen, P. H. and Maguire, M. C. (1998) *Guide to mapping requirements to user interface specifications*, EC Telematics Applications Programme, Project TE 2010 RESPECT, WP4 Deliverable D4.2, May 1998.

Wilson J. and Rosenberg D. (1988) Rapid prototyping for user interface design. In *Handbook of Human-Computer Interaction*, (Helander M. ed.) pp859-875; Amsterdam, North-Holland, ISBN 0-444-70536-8.

Young E. and Greenlea R. (1992) Participatory video prototyping. *CHI'92 conference proceedings.* (Poster exhibit).

Appendix 1 - User Interface Guidelines

The following set of general guidelines may be used for the development of the user interface to the proposed system. Before starting the design, it is recommended that this section be reviewed and key phases highlighted. This will form a simple checklist which will act as a reminder of the main general aims of the design.

Simplicity
It is easy to overestimate peoples' ability to use systems. Aim to keep the basic interface simple offering facilities which are of clear value to potential users. Provide the most important facilities on separate keys. Those facilities that are infrequently used or are seen as 'value-added' should be shielded from the unsophisticated user for access via a menu or key combination.

Flexibility
Due to the inherent complexity of the human interaction with many systems, a structure needs to be imposed upon the user dialogue in order to guide the user through it. However, such structures often seem to place barriers on the person's usage and make the device seem inflexible and unusable. Therefore specify access facilities that take into account the kinds of paths that people will wish to follow. Make it very convenient to change modes so that the user should not need to close down one activity in order to start another. Clearly interface design could become very complex, so limits have to be placed on such movements.

Consistency
The user should feel that they are in control and that the system is responding to his or her actions, not vice versa. Care should be taken to ensure that the user does not feel paced by the system, a common problem with systems making excessive use of time-outs. Users should have control over the amount of information they receive at different points of the interaction.

Control
Sequences of actions should generate the expected response, identical terminology and abbreviations should be used throughout, prompts should always appear in the same place. However total consistency cannot always be achieved. For example it may be made easier to recall a stored number on a telephone by simply pressing one key to recall its memory location, compared with the user having to enter a leading zero when storing in memory.

Shortcuts
Frequent use brings with it the desire to reduce the number of interaction steps and to speed up the interaction process. Macro facilities, special key combinations and 'fast path' keys and links are useful additions to the system, but only if their presence does not interfere in any way with the dialogue as presented to a novice user.

Screen layout
The screen layout should give an appearance of clarity with white space used to separate items. The items on the screen should be grouped together appropriately and sequence in a logical order. Human short-term memory limits are generally taken to be 'seven plus or minus two' chunks of information. Displays should be kept simple and users should not be required to 'carry-over' information from one display to another.

Prompts, labels, messages and feedback
Every operator action should elicit some system feedback. At its simplest this might be a click to confirm a key press. Providing no feedback can lead the user to believe that no action has taken place. It is very important to give feedback at the end of a sequence or operations to give the user the satisfaction of reaching task closure. Also provide prompts to guide the user through the interaction sequence. The user should never be left in the state of not knowing what to do next. Messages should be constructive and give guidance for using the system in a courteous way. All messages should be part of the system design and available in the user manual.

Error handling
The user should not be able to damage the equipment or make serious error. Destructive commands such as deleting a directory or erasing all memories should be structured such that the user is made to confirm his action. Inapplicable commands should leave the system state unchanged. Ideally, any action should be 'undoable' or reversible, so that a user does not fear learning by experimentation, though this is often difficult to implement.

Efficiency
The user will generally have the need to carry out tasks as quickly as possible. Bear in mind, when producing a design, what time and effort the user will be prepared to put into achieving their goals with the system.

Help facilities
Help facilities should be easily accessible. The user should not have to spent a long time reading the help in order to use the system or to overcome problems. Therefore help text should be quick and easy to read. Where possible provide context sensitive help so that the help relates to the current point within the system. Help should also be task oriented so that it explains actions in terms of the tasks the user may wish to perform. Allow the system to be used while help remains displayed and avoid obscuring crucial parts of the screen with the help window.

Appendix 2 - Human Factors Standards

Description of standards

Standards related to human-centred design fall into two categories:
- **process-oriented**: these specify procedures and processes to be followed.
- **product-oriented**: these specify required attributes of the user interface.

Some product oriented standards specify the requirements in terms of performance rather than product attributes. These standards describe the users, tasks, and context of use and assess usability in terms of user performance and satisfaction to be achieved.

Process oriented

1. ISO 13407/DIS (1997) Human-centred design processes for interactive systems

This standard provides guidance on human-centred design activities throughout the life cycle of interactive computer-based systems. It is a tool for those managing design processes and provides guidance on sources of information and standards relevant to the human-centred approach. It describes human-centred design as a multi-disciplinary activity, which incorporates human factors and ergonomics knowledge and techniques with the objective of enhancing effectiveness and efficiency, improving human working conditions, and counteracting possible adverse effects of use on human health, safety and performance.

2. ISO 6385 : 1981, Ergonomic principles in the design of work systems

ISO 6385 sets out the ergonomic principles which should be applied to the design of work systems. ISO 13407 is based on these principles and the description of the aims and objectives of ergonomics which are contained in ISO 6385

3. ISO 9241-1: 1993 Ergonomic requirements for office work with visual display terminals (VDTs) - General Introduction

This part introduces the multi-part standard ISO 9241 for the ergonomic requirements for the use of visual display terminals for office tasks and explains some of the basic underlying principles. It provides some guidance on how to use the standard and describes how conformance to parts of ISO 9241 should be reported.

4. ISO 9241-2 : 1993 : Guidance on task requirements

This part deals with the design of tasks and jobs involving work with visual display terminals. It provides guidance on how task requirements may be identified and specified within individual organisations and how task requirements can be incorporated into the system design and implementation process.

5. ISO /DIS 9241-11 : Guidance on Usability

This part provides the definition of usability which is used in ISO 13407:

Usability: the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

ISO/DIS 9241-11 explains how to identify the information which it is necessary to take into account when specifying or evaluating usability in terms of measures of user performance and satisfaction. Guidance is given on how to describe the context of use of the product (hardware, software or service) and the required measures of usability in an explicit way. It includes an explanation of how the

usability of a product can be specified and evaluated as part of a quality system, for example one which conforms to ISO 9001.

It also explains how measures of user performance and satisfaction can be used to measure how any component of a work system affects the quality of the whole work system in use.

6. ISO 10075-1: 1994 Ergonomic principles related to mental work-load - General terms and definitions

This part of ISO 10075 explains the terminology and provides definitions in the area of mental workload

7. ISO/IEC CD 14598-1: Information Technology - Evaluation of Software Products - General guide

The concept of quality in use has been used in ISO/IEC 14598-1 to distinguish between quality as an inherent characteristic of a software product and the quality which is achieved when a software product is used under stated conditions, that is, a specified context of use. This definition of quality in use is very similar to the definition of usability in ISO/DIS 9241-11. The use of the term quality in use therefore implies that it is necessary to take account of human-centred issues in evaluating software products.

Quality in use: the extent to which an entity satisfies stated and implied needs when used under stated conditions.

Standards of this type can be used to support the following activities:

- specification of overall quality and usability requirements and evaluation against these requirements (ISO/DIS 9241-11 and ISO/IEC CD 14598-1)
- incorporation of usability into a quality system (ISO/DIS 9241-11)

Product oriented standards

In the product-oriented view, usability is seen as one relatively independent contribution to software quality, and is defined in this way in:
ISO/IEC 9126 : 1991 : Information technology - Software product evaluation - Quality characteristics and guidelines for their use:

'a set of attributes of software which bear on the effort needed for use and on the individual assessment of such use by a stated or implied set of users'.

Usable products can be designed by incorporating product features and attributes known to benefit users in particular contexts of use. ISO 9241 provides requirements and recommendations relating to the attributes of the hardware, software and environment which contribute to usability, and the ergonomic principles underlying them. Parts 3 to 9 contain hardware design requirements and guidance which can have implications for software. Parts 10 to 17 of ISO 9241 and other standards deal specifically with attributes of the software:

11. ISO 9241-3: 1993 Visual display requirements

This part specifies the ergonomics requirements for display screens which ensure that they can be read comfortably, safely and efficiently to perform office tasks. Although it deals specifically with displays used in offices, it is appropriate to specify it for most applications which require general purpose displays to be used in an office-like environment.

12. ISO DIS 9241-4 Keyboard requirements

This part specifies the ergonomics design characteristics of an alphanumeric keyboard which may be used comfortably, safely and efficiently to perform office tasks. Keyboard layouts are dealt with separately in various parts of ISO/IEC 9995: 1994 Information Processing - Keyboard Layouts for Text and Office Systems

13. ISO DIS 9241-5 Workstation layout and postural requirements

This part specifies the ergonomics requirements for a Visual Display Terminal workplace which will allow the user to adopt a comfortable and efficient posture.

14. ISO DIS 9241-6 Environmental requirements

This part specifies the ergonomics requirements for the Visual Display Terminal working environment which will provide the user with comfortable, safe and productive working conditions.

15. ISO DIS 9241-7 Display requirements with reflections

This part specifies methods of measurement of glare and reflections from the surface of display screens, including those with surface treatments. It is aimed at display manufacturers who wish to ensure that anti-reflection treatments do not detract from image quality.

16. ISO DIS 9241-8 Requirements for displayed colours

This part specifies the requirements for multi-colour displays which are largely in addition to the monochrome requirements in Part 3.

17. ISO DIS 9241-9 Requirements for non-keyboard input devices

This part specifies the ergonomics requirements for non-keyboard input devices which may be used in conjunction with a visual display terminal. It covers such devices as the mouse, trackerball and other pointing devices. It also includes a performance test. It does not address voice input.

18. ISO 9241-10: 1996 Dialogue principles

This part deals with general ergonomic principles which apply to the design of dialogues between humans and information systems: suitability for the task, suitability for learning, suitability for individualisation, conformity with user expectations, self descriptiveness, controllability, and error tolerance

19. ISO/CD 9241-12: Presentation of information

This part contains specific recommendations for presenting and representing information on visual displays. It includes guidance on ways of representing complex information using alphanumeric and graphical/symbolic codes, screen layout, and design as well as the use of windows.

20. ISO/DIS 9241-13: User guidance

This part provides recommendations for the design and evaluation of user guidance attributes of software user interfaces including Prompts, Feedback, Status, On-line Help and Error Management.

21. ISO/DIS 9241-14: Menu dialogues

This part provides recommendations for the ergonomic design of menus used in user-computer dialogues. The recommendations cover menu structure, navigation, option selection and execution, and menu presentation (by various techniques including windowing, panels, buttons, fields, etc.). Part 14 is intended to be used by both designers and evaluators of menus (however, its focus is primarily towards the designer).

22. ISO/DIS 9241-15: Command language dialogues

This part provides recommendations for the ergonomic design of command languages used in user-computer dialogues. The recommendations cover command language structure and syntax, command

representations, input and output considerations, and feedback and help. Part 15 is intended to be used by both designers and evaluators of command dialogues, but the focus is primarily towards the designer.

23. ISO/DIS 9241-16: Direct manipulation dialogues

This part provides recommendations for the ergonomic design of direct manipulation dialogues, and includes the manipulation of objects, and the design of metaphors, objects and attributes. It covers those aspects of 'Graphical User Interfaces' which are directly manipulated, and not covered by other parts of ISO 9241. Part 16 is intended to be used by both designers and evaluators of command dialogues, but the focus is primarily towards the designer.

24. ISO/DIS 9241-17: Form-filling dialogues

This part provides recommendations for the ergonomic design of form filling dialogues. The recommendations cover form structure and output considerations, input considerations, and form navigation. Part 17 is intended to be used by both designers and evaluators of form-filling dialogues, but the focus is primarily towards the designer.

25. ISO/IEC 10741-1 Dialogue interaction - Cursor control for text editing

This International Standard specifies how the cursor should move on the screen in response to the use of cursor control keys.

26. ISO/IEC DIS 11581-1 : Icon symbols and functions - Part 1: Icons - general

This part contains a framework for the development and design of icons, including general requirements and recommendations applicable to all icons.

27. ISO/IEC DIS 11581-2 : Icon symbols and functions - Part 2: Object icons

This part contains requirements and recommendations for icons that represent functions by association with an object, and that can be moved and opened. It also contains specifications for the function and appearance of 20 icons.

These standards can be used in the following ways:

- To specify details of the appearance and behaviour of the user interface.
- To provide detailed guidance on the design of user interfaces.
- To provide criteria for the evaluation of user interfaces.

However the attributes which a product requires for usability depend on the nature of the user, task and environment. A product has no intrinsic usability, only a capability to be used in a particular context. ISO/DIS 9241-11 can be used to help understand the context in which particular attributes can be required.

Appendix 3 - Blank forms to support User Requirements specification

These appendices contain blank forms for use in carrying out the stages of requirements specification. They may be photocopied as required for completion. The forms included are as follows:-

STAGE 1: USER CONTEXT AND EARLY DESIGN

FORM 1.1 - PROJECT SUMMARY

FORM 1.2 - USERS AND STAKEHOLDERS

FORM 1.3 - USER GROUP CHARACTERISTIC S

FORM 1.4 - TECHNICAL ENVIRONMENT

FORM 1.5 - PHYSICAL ENVIRONMENT

FORM 1.6 - SOCIAL AND ORGANISATIONAL ENVIRONMENT

FORM 1.7 - USER GOALS AND TASKS

FORM 1.8 - CURRENT PROCESS

FORM 1.9 - FUNCTIONS AND FEATURES OF SIMILAR SYSTEMS

FORM 1.10 - DESIGN IDEAS AND CONCEPTS

STAGE 2: PROTOTYPING AND USER TEST

FORM 2.1 - GENERAL USABILITY GOALS

FORM 2.2 - DESIGN CONSTRAINTS

FORM 2.3 - TASK SCENARIOS

FORM 2.4- PROPOSE NEW PROCESSES

FORM 2.6- TASK WALKTHROUGH FEEDBACK

FORM 2.7 - USER COST-BENEFITS

STAGE 3: USER REQUIREMENTS DOCUMENTATION

FORM 3.1 - GENERAL SYSTEM CHARACTERISTICS

FORM 3.3 - TASK SCENARIOS AND INTERACTION STEPS

FORM 3.4 - TECHNICAL ENVIRONMENT REQUIREMENTS

FORM 3.5.1 - SYSTEM FUNCTIONS

FORM 3.5.2 - SYSTEM FEATURES

FORM 3.6 - USER INTERFACE DESIGN

FORM 3.7 - USER SUPPORT REQUIRED

FORM 3.8 - PHYSICAL REQUIREMENTS

FORM 3.9 - SOCIAL AND ORGANISATIONAL ENVIRONMENT

FORM 3.10 - STANDARDS TO APPLY

FORM 3.11.1 - USABILITY TEST PLAN

FORM 3.11.2 - USABILITY TEST RESULTS

FORM 3.12 - USER REQUIREMENTS IMPLEMENTATION PLAN

Form 1.1 - Project Summary

1.1	Project Summary
Questions	**Assumptions**
What is the system or service?	
What functions or services is it intended for the system to provide?	
What are the aims of the project?	
Who is the system intended for? (Target market)	
Who will use the system?	
Why is the system needed?	
Where will the system be used?	
How will the system be used?	
How will the user obtain the system?	
How will the user learn to use the system?	
How will the system be installed?	
How will the system be maintained?	

Form 1.2 - Users and Stakeholders

1.2 Users and Stakeholders		
System name:		
DIRECT USERS	**ROLE IN SYSTEM OR USE OF SYSTEM**	**EXPAND**
INDIRECT USERS	TASK GOALS	

Each user group ticked may be described in the User context FORMS 1.3 to 1.6
All goals for each user group will be listed in FORM 1.7

Form 1.3 - User group characteristics

(Form completed for each user/stakeholder group listed in FORM 1.2)

1.3 User group characteristics		
System name: **User group:** ⬅ Form completed for users groups selected in FORM 1.2		
CHARACTERISTICS	**POTENTIAL USER REQUIREMENTS**	**REF.**
Size of user group		
Age range		
Gender		
Language and culture		
Educational level/Qualifications		
Physical limitations/Disabilities		
Special skills (e.g. touch typing, use of mouse, spatial awareness)		

⇨ Continued on next page

⬇ **Transfer to relevant FORMS in Phase 3**
e.g. FORM 3.1 General system characteristics
or FORM 3.4 Technical environment

1.3 User group characteristics (continued)		
CHARACTERISTICS	POTENTIAL USER REQUIREMENTS	REF.
Experience with similar systems		
IT Experience		
Knowledge of task		
Previous training		
Frequency of use		
Motivation to use		
Discretion to use		
Likely concerns		
Other relevant characteristics		

Transfer to relevant FORMS in Phase 3
e.g. FORM 3.1 General system characteristics
or FORM 3.4 Technical environment

Form 1.4 - Technical environment

1.4 Technical Environment		
System name: **User group:**	⬅ Form completed for users groups selected in FORM 1.2	
CHARACTERISTICS	POTENTIAL USER REQUIREMENTS	REF.
Hardware which user will interact with e.g. desktop PC, printer, kiosk.		
Software environment in which system will run **e.g. Windows, WWW Browser.**		
Software environment to be used to develop system.		
Other equipment required for use alongside system.		
Reference materials required either to perform tasks with system or to learn about or operate system		

⬇ **Transfer to relevant FORMS in Phase 3**
e.g. FORM 3.1 General system characteristics
or FORM 3.4 Technical environment

Form 1.5 - Physical Environment

1.5 Physical Environment		
System: **User group:**		
CHARACTERISTICS	**POTENTIAL USER REQUIREMENTS**	**REF.**
Thermal and atmospheric environment		
Auditory Environment		
Vibration or instability		
Visual Environment		
Space and furniture		
User posture		
Location		
Health and Safety hazards		
Protective clothing and equipment		

Transfer to FORM 3.8 Physical environment

Form 1.6 - Social and Organisational Environment

1.6 Social and Organisational Environment		
System: **User group:**		
CHARACTERISTICS	POTENTIAL USER REQUIREMENTS	REF.
Staff and Management structure		
Communications structure		
IT Policy		
Organisational aims		
Industrial Relations		
Performance monitoring		
Performance feedback		
Group working		
Assistance required or available		
Interruptions, stressful conditions		

Transfer to FORM 3.9 Social and organisational environment

1.6 Social and Organisational Environment (continued)		
System: **User group:**		
CHARACTERISTICS	**USER REQUIREMENTS**	**REF.**
Safety and Security		
Privacy		
Job function		
Hours of work		
Job flexibility		
Valued skills		

Transfer to FORM 3.9 Social and organisational environment

Form 1.7 - User Goals and Tasks

1.7	User Goals and Tasks			
System:				

Goals are expanded in 'Review current process', in FORM 1.8.

Form 1.8 - Current Process

1.8	Current process		
User group: **Sub-Goal:**			
TASK STEP	PROBLEM/TASK VARIATION	USER REQUIREMENT	REF.

All information used to generate task scenarios and propose new processes in Section 2.4

Transfer to relevant forms especially 3.5 Functions/Features

Form 1.9 - Functions and features of similar systems

1.9 Functions and features of similar systems				
System:				
	PRODUCT NAME	**GOOD FEATURE TO INCLUDE**	**POOR FEATURE TO EXCLUDE**	**REF.**
GENERAL IDEAS				
GOAL SPECIFIC				

Transfer to 3.5 Functions and Features

Form 1.10 - Design Ideas and Concepts

1.10	Design ideas and concepts		
System: *New bank machine*			
	IDEAS AND CONCEPTS	**COMMENTS**	**Take Forward? /Ref.**
GENERAL IDEAS			
GOAL SPECIFIC			

Transfer those taken forward to Form 3.5 Functions and Features

Form 2.1 - General Usability Goals

2.1 General Usability Goals		
System: **User:**		
USABILITY GOAL	USER REQUIREMENT WITH RESPECT TO GOAL	KEY GOALS
Effectiveness Quality or quantity of output/task completion.		
Efficiency Time to perform task, time compared with an expert.		
Satisfaction Perceived satisfaction or enjoyment in using the system.		
Learnability Ability to use the system help or manuals to perform the task.		
Intuitiveness Ability to perform the tasks with limited introduction.		
Helpfulness/supportiveness Ability to overcome problems that arise.		
Controllability Perceived feeling of being in control/tracking performance etc.		
Avoiding excessive mental load Perceived mental effort, or physical indicators.		
Avoiding excessive physical load Heart rate, respiratory measurement.		
Safety To be able to operate the system safely.		

Form 2.2 - Design Constraints

2.2 Design Constraints	
System:	REF.

Form 2.3 - Task Scenarios

2.3 Task Scenarios		
System: **User group:** **⬇ Based on FORM 1.7, User goals and 1.8, Current processes.**		
REF.	SCENARIO	PERFORMANCE AIM

Form 2.4 - Propose New processes

2.4	New processes		
User groups concerned: **Goal:**			
Scenario	⬅ Form completed for all user goals listed in FORM 2.3		
Performance Aim			
TASK STEP	**POSSIBLE FUNCTIONS OR FEATURES**	**INCLUDE**	**REF.**

⬇ **Transfer to relevant forms especially 3.5 Functions/Features**

Form 2.6- Task Walkthrough Feedback

2.6	**Task walkthrough feedback**		
System: **User group:** ⬇ **Transferred from FORM 2.3 Task scenarios.**			
TASK SCENARIO	**COMMENTS**	**MODIFIED OR NEW SYSTEM REQUIREMENT**	**REF.**
		⬇ **Copy to relevant Stage 3 Forms.**	

Form 2.7 - User cost-benefits

2.7	User Cost/Benefits					
User group:						
ISSUE	BENEFITS	RATING +1 TO +5	COSTS	RATING -1 TO -5	MODIFIED OR NEW USER REQUIREMENT	REF.

Transfer to FORM 3.5 System Functions and Features and FORM 3.4 Technical environment

2.7 User Cost/Benefits

User group:

ISSUE	BENEFITS	RATING +1 TO +5	COSTS	RATING -1 TO -5	MODIFIED OR NEW USER REQUIREMENT	REF.
Job Security						
Job Content 1. 2. 3. 4. 5. 6.						
Organisational procedures 1. 2. 3. 4. 5. 6.	No effect					
Personal policies 1. 2. 3. 4. 5.	No effect					

Summary:

Transfer to FORM 3.9 Social and Organisational Requirements

Form 3.1 - General System Characteristics

3.1 General System Characteristics			
System: ⬇ **Transfer from FORM 1.3 User Characteristics, 2.3 Task scenarios and FORM 1.10 Design Concept characteristics**	PRI. 1-5	ACH. * ** ***	REF.

Form 3.3 - Task Scenarios and Interaction Steps

3.3	Task Scenarios and Interaction Steps
System : **Scenario :**	
⬇ **Transfer from FORM 2.3 Task scenarios and FORM 2.4 New processes**	PRI: ACH:
USER	SYSTEM

Form 3.4 - Technical Environment requirements

3.4 Technical environment requirements			
System:			
⬇ Transfer from FORM 1.4 Technical environment	PRI.	ACH.	REF.
Main hardware required			
Software environment			
Output hardware			
Input hardware			
Standards to be applied			
Communication/network capabilities			
Other equipment			
Other characteristics			

Form 3.5.1 - System Functions

3.5.1 System Functions			
System: **Transfer from FORM 1.8 Review current processes** **FORM 1.9 Review other systems and products** **FORM 1.10 Design ideas and concepts** **and FORM 2.4 Propose new processes**	PRI. 1-5	ACH. * ** ***	REF.

Form 3.5.2 - System Features

3.5.2 System Features			
System: **Transfer from FORM 1.8 Review current processes** **FORM 1.9 Review other systems and products** **FORM 1.10 Design ideas and concepts** **and FORM 2.4 Propose new processes**	PRI. 1-5	ACH. * ** ***	REF.

Form 3.6 - User Interface design

3.6 User Interface Design

Interface name: **⬇ Transfer from FORM 1.4 Technical environment** **Transfer from FORM 1.10 Design ideas and concepts**	PRI.	ACH.	REF.
Purpose			
Features			
Components of interface			
Standards to be used			
Example screens			
Tools to be used for building interface			

Form 3.7 - User Support

3.7	User Support			
User Group: ⬇ **Transfer from FORM 1.3 User characteristics**		PRI.	ACH.	REF.
User training				
User document-ation				
Help facilities				
Local expert				
Telephone hot lines				
Other support				

Form 3.8 - Physical Environment

3.8 Physical Environment			
System: **⬇ Transfer from FORM 1.5 Physical environment**	PRI.	ACH.	REF.

Form 3.9 - Social and Organisational Environment

3.9 Social and Organisational Environment			
System : **⬇ Transfer from FORM 1.6 Social and Organisational environment**	PRI.	ACH.	REF.

Form 3.10 - Standards to Apply

3.10 Standards and styleguides to Apply				
System:				
STANDARD	**USER REQUIREMENT (METHOD OF APPLICATION)**	**PRI.**	**ACH.**	**REF.**

Form 3.11.1 - Usability test plan

3.11.1	Usability test plan
User sample and characteristics	
Method of recruitment	
Tasks or Scenarios ⬆ **Based on FORM 1.7 user goals or FORM 2.3 Task scenarios**	
Measures/Method of data capture	
Test equipment	
Environment ⬆ **based on FORMS:** **1.4 Technical environment** **1.5 Physical environment** **1.6 Social and organisational environment**	
Introduction to subject	
Assistance to be given	

Form 3.11.2 - Usability test results

3.11.2	Usability test results				
Tasks	**Task achievement**	**Mean Time and Standard deviation**	**Comparison with expert task time**	**Subjective rating of difficulty (1=easy 7=difficult)**	**Comments**

Form 3.12 - User Requirements Implementation Plan

3.12 User requirements implementation plan
System:
1. Design rationale
2. Form of prototype
3. Usability test method
4. Acceptability test method
5. Scope for iteration
6. User audit
7. System phasing

Part A

Introduction to the Handbook

Part B

User Requirements Framework

Part C

User Requirements Methods

Part D

References and Appendices

Phase 1.

User Context and Early Design

Phase 2.

Prototyping and User Test

Phase 3.

User Requirements Documentation