

Australian/New Zealand Standard™

**Guide to the development of application
software**

Part 1: Software user documentation

AS/NZS 4598.1:1999

This Joint Australian/New Zealand Standard was prepared by Joint Technical Committee IT/15, Software Engineering. It was approved on behalf of the Council of Standards Australia on 14 July 1999 and on behalf of the Council of Standards New Zealand on 28 June 1999. It was published on 5 August 1999.

The following interests are represented on Committee IT/15:

Australian Association of Chief Information Officers
Australian Bankers Association
Australian Computer Society
Australian Electrical and Electronic Manufacturers Association
Australian Information Industry Association
Australian Society for Technical Communication
Australian Software Metrics Association
Computer Aided Software Engineering Special Interest Group
Defence Science and Technology Organization
Department of Defence, Australia
Griffith University
IEEE Computer Society
Indigenous Information Industries Group
New Zealand Organization for Quality
Quality Society of Australasia
Software Process Improvement Network, Sydney, Australia
Software Quality Association
Software Verification Research Centre
Telstra Corporation
University of New South Wales
University of South Australia

Review of Standards. To keep abreast of progress in industry, Joint Australian/New Zealand Standards are subject to periodic review and are kept up to date by the issue of amendments or new editions as necessary. It is important therefore that Standards users ensure that they are in possession of the latest edition, and any amendments thereto.

Full details of all Joint Standards and related publications will be found in the Standards Australia and Standards New Zealand Catalogue of Publications; this information is supplemented each month by the magazines 'The Australian Standard' and 'Standards New Zealand', which subscribing members receive, and which give details of new publications, new editions and amendments, and of withdrawn Standards.

Suggestions for improvements to Joint Standards, addressed to the head office of either Standards Australia or Standards New Zealand, are welcomed. Notification of any inaccuracy or ambiguity found in a Joint Australian/New Zealand Standard should be made without delay in order that the matter may be investigated and appropriate action taken.

This Standard was issued in draft form for comment as DR 99190.

Australian/New Zealand Standard™

Guide to the development of application software

Part 1: Software user documentation

First published as AS/NZS 4598.1:1999

Published jointly by:

Standards Australia
1 The Crescent,
Homebush NSW 2140 Australia

Standards New Zealand
Level 10, Radio New Zealand House,
155 The Terrace,
Wellington 6001 New Zealand

PREFACE

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT/15, Software Engineering. This Standard is identical with and has been reproduced from BS 7649:1993, *Guide to the design and preparation of documentation for users of application software*.

The objective of this Standard is to provide designers of application software with guidelines for the design and preparation of user documentation for application software covering what information users need, how to decide what information should be provided (on paper and on-screen) and how to prepare the relevant documentation.

The term 'informative' has been used in this Standard to define the application of the annex to which it applies. An 'informative' annex is only for information and guidance.

As this Standard is reproduced from a British Standard, the following applies:

- (a) Its number does not appear on each page of text and its identity is shown only on the cover and title page.
- (b) In the source text 'this British Standard' should read 'this Australian/New Zealand Standard'.
- (c) A full point should be substituted for a comma when referring to a decimal marker.

References to British Standards should be replaced by references to the following Australian or Australian/New Zealand Standards:

<i>Reference to International Standard</i>		<i>Australian or Australian/New Zealand Standard</i>
BS		AS
3700	Recommendations for preparing indexes to books, periodicals and other documents	—
4000	Sizes of paper and board	—
4000-1	Part 1: Specification for A and B series of trimmed sizes of writing paper and certain classes of printed matters	—
4000-2	Part 2: Specification for untrimmed sizes	—
4884	Technical manuals	—
4884-1	Part 1: Specification for presentation of essential information	—
4884-2	Part 2: Guide to content	—
4884-3	Part 3: Guide to presentation	—
4899	Guide to user's requirements for technical manuals (based on the principles of BS 4884)	—
4899-1	Part 1: Content	—
4899-2	Part 2: Presentation	—
5097	Specification for loose leaf binders	—
5097-1	Part 1: Ring binders with metal mechanisms	—

BS		AS	
5261C	Marks for copy preparation and proof correction (extracted from BS 5261 : Part 2 : 1976)	—	
5515	Code of practice for documentation of computer-based systems	—	
5605	Recommendations for citing and referencing published material	—	
5641	Recommendations for loose-leaf publications	—	
5750	Quality systems	—	
5848	Specification for numbering of divisions and subdivisions in written documents (point-numbering)	—	
6738	Recommendations for presentation of spine titles	—	
7137	Specification for user documentation and cover information for consumer software packages	—	
7165	Recommendations for achievement of quality in software	—	
7581	Guide to presentation of tables and graphs	—	
ISO/IEC TR 9294	Information technology. Guidelines for the management of software documentation	3897	Information processing—Guidelines for the management of software documentation

© Copyright – STANDARDS AUSTRALIA/STANDARDS NEW ZEALAND

Users of Standards are reminded that copyright subsists in all Standards Australia and Standards New Zealand publications and software. Except where the Copyright Act allows and except where provided for below no publications or software produced by Standards Australia or Standards New Zealand may be reproduced, stored in a retrieval system in any form or transmitted by any means without prior permission in writing from Standards Australia or Standards New Zealand. Permission may be conditional on an appropriate royalty payment. Australian requests for permission and information on commercial software royalties should be directed to the head office of Standards Australia. New Zealand requests should be directed to Standards New Zealand.

Up to 10 percent of the technical content pages of a Standard may be copied for use exclusively in-house by purchasers of the Standard without payment of a royalty or advice to Standards Australia or Standards New Zealand.

Inclusion of copyright material in computer software programs is also permitted without royalty payment provided such programs are used exclusively in-house by the creators of the programs.

Care should be taken to ensure that material used is from the current edition of the Standard and that it is updated whenever the Standard is amended or revised. The number and date of the Standard should therefore be clearly identified.

The use of material in print form or in computer software programs to be used commercially, with or without payment, or in commercial contracts is subject to the payment of a royalty. This policy may be varied by Standards Australia or Standards New Zealand at any time.

CONTENTS

Page

INTRODUCTION	1
Section 1. General	2
1.1 Scope	2
1.2 Informative references	2
1.3 Definitions	2
Process	
Section 2. Integrated design and development of software products	4
2.1 Integrated development stages	4
2.2 Responsibilities for documentation activities	5
2.3 Local standards and procedures	6
2.4 Documentation processes	6
2.5 Sequence of documentation activities	6
Section 3. Project objectives, requirements and constraints	8
3.1 General	8
3.2 Collect and interpret project requirements and constraints	8
Section 4. Analysis	14
4.1 General	14
4.2 Collect details of users	14
4.3 Determine what documentation users need	19
4.4 Record users' needs for information	21
4.5 Produce the outline documentation plan	21
Section 5. Planning	26
5.1 General	26
5.2 Design the content, structure and style of each document	26
5.3 Set the editorial standards	26
5.4 Design the physical presentation of each document	28
5.5 Design the complete package	28
5.6 Decide how the documents will be developed	29
5.7 Decide how the documents will be produced	29
5.8 Decide how the documents will be updated	30
5.9 Estimate costs	31
5.10 Prepare schedules	32
5.11 Revise the design	32
5.12 Write the documentation plan	32
5.13 Review the documentation plan	33
Section 6. Development and review	34
6.1 General	34
6.2 Prepare and issue first draft	34
6.3 Check and review drafts	35
6.4 Prepare subsequent drafts	36
6.5 Validation and field trials	36
6.6 Prepare master pages	37

Section 7. Production and distribution	38
7.1 General	38
7.2 Submit master pages for production	38
7.3 Package the documents and software	39
7.4 Check the finished documents and package	39
7.5 Archive the master versions	40
7.6 Distribute the product	40
Section 8. Evaluation and updating	41
8.1 General	41
8.2 Evaluate the documentation	41
8.3 Update the documentation	41

Detailed design

Section 9. Document contents	43
9.1 General	43
9.2 Contents of awareness or appreciation documents	45
9.3 Contents of installation instructions	45
9.4 Contents of tutorials	46
9.5 Contents of quick reference documents	47
9.6 Contents of comprehensive reference documents	47
Section 10. Document structures	48
10.1 General	48
10.2 Structure of awareness or appreciation documents	49
10.3 Structure of installation instructions	49
10.4 Structure of tutorial documents	49
10.5 Structure of quick reference documents	52
10.6 Structure of comprehensive reference documents	53
10.7 Numbering schemes for pages, sections, illustrations and tables	54
Section 11. Document writing styles	58
11.1 General	58
11.2 Writing style for awareness and appreciation documents	58
11.3 Writing style for installation instructions	59
11.4 Writing style for tutorial documents	59
11.5 Writing style for quick reference documents	59
11.6 Writing style for comprehensive reference documents	60
11.7 Types of illustration	60
Section 12. Document writing techniques	62
12.1 General	62
12.2 Writing techniques	62
12.3 Vocabulary	67
12.4 Technical terms	68
Section 13. Document presentation	69
13.1 General	69
13.2 Page layouts	69
13.3 Use of colour	74
13.4 Presentation of text	74
13.5 Presentation of lists, tables and notes	78
13.6 Presentation of illustrations	80

	<i>Page</i>
13.7 Special symbols and icons	81
13.8 Quality and durability of the finished documents	81
13.9 Binding	81
13.10 Packaging	82
13.11 Cover information	82
Section 14. Document contents lists and indexes	84
14.1 General	84
14.2 Contents lists	84
14.3 Indexes	84
Tables	
1 Kinds of information for different learning stages	21
2 Types of documents and their uses	25
3 Methods of preparing master pages	29
4 Methods of producing multiple copies	31
5 The suitability of types of binding for different needs	82
Figures	
1 Integrated development process	4
2 Documentation development activities	7
3 The project objectives, requirements and constraints process	8
4 The analysis process	14
5 Example list of types of user	15
6 Example user hierarchy	15
7 Example task list	17
8 Example task hierarchy	17
9 Example chart of users and tasks	18
10 Example of information needed by one user type	20
11 Example of grouping information into documents	24
12 Example entry for an outline documentation plan	24
13 The planning process	27
14 The development and review process	34
15 The production and distribution process	38
16 The evaluation and updating process	41
17 Example structure of an awareness document	49
18 Example structure of a getting started tutorial	51
19 Example structure of a tutorial for normal use	52
20 Example structures of quick reference documents	53
21 Example structure of a task-related comprehensive reference document	55
22 Example structure of a software-related comprehensive reference document	56
23 Example of a table used to illustrate a complicated condition	64
24 Page grid for this British Standard	71
25 Page grid for an A5 page	72
26 Example A5 page	73
27 Example of the use of bold and italic and different typefaces and sizes	76

	<i>Page</i>
Annexes	
A (informative) Bibliography	86
A.1 Reference books	86
A.2 Use of English	86
A.3 Preparing user documentation	86
A.4 Typography	86
B (informative) Collected checklists	87
B.1 Project objectives, requirements and constraints	87
B.2 Analysis	88
B.3 Planning	89
B.4 Development and review	91
B.5 Production and distribution	92
Index	93

NOTES

AUSTRALIAN/NEW ZEALAND STANDARD

Guide to the development of application software

Part 1: Software user documentation

Introduction

Anyone who wants or needs to use application software needs accurate information about the correct way to use it. If the information is supplied in a convenient form and is easy to find and understand, users can quickly become proficient at using the product. Their view of the product is positive, with the result that their view of the supplier is positive too. Hence, well-designed documentation not only assists users and helps to keep down the costs of training and support but also enhances the reputation of the product, its producer and its suppliers.

Although many software products aim to have user interfaces that behave so naturally that very little separate documentation is needed, this is rarely possible. For example, sometimes the technology being used does not support such interfaces and sometimes the subject area is too complex to be handled in that way.

Documentation is an essential component of any product. Documentation design is crucial: the success or failure of an entire product can depend on it. The documentation is the first tangible item that a user sees, and so influences the first impressions of the product.

Users of application software products generally have one important feature in common: they might be experts in the tasks for which they wish to use the software, but they are not, initially, experts in using the application software itself.

For application software, there are special needs. To access the product at all, the user has to be able to operate the computer well enough to use at least the first few commands or instructions. In addition, application software can cause new users extreme anxiety and stress; some users are actually afraid of using it.

The information about the software that a user needs can come from all parts of the product:

- a) the packaging;
- b) the distribution medium, such as diskette or tape;
- c) the screen displays and on-line information;
- d) user manuals;
- e) other documents, such as labels, templates, charts and cards.

The guidance given in this British Standard covers specifically the design and preparation of d) and e).

Although the guidance given in this British Standard covers all the activities and all the design decisions that need to be made, some of the activities can be extremely simple to carry out in some environments, as demonstrated by the following examples.

- If there are already established typography and illustration standards and established development and production routes, very little design and planning will be needed in these areas.
- If the product is being developed for a single type of user with well-known user characteristics and well-defined tasks, very little user analysis will be needed.

Annex A is a bibliography of other books that might be useful to authors.

There is an index at the back of the standard.

Section 1. General

1.1 Scope

This British Standard gives guidance on the design and preparation of user documentation for application software.

Application software is software designed to help users perform particular tasks or to handle particular types of problem as distinct from software that controls the computer system itself.

For the purposes of this British Standard, application software includes:

- consumer software packages, that is, software products designed and sold to carry out identified functions, where the software and its associated documentation are packaged for sale as a unit;
- software for office applications such as word processing, spreadsheets, databases and electronic mail;
- business software, for example, software for recording and monitoring business activities such as stock control and processing orders;
- specialist software for use by professionals, such as accounting systems, graphic design systems and engineering design systems.

For the purposes of this British Standard, application software does not include:

- software engineering products for use by computer professionals;
- software for programmable electronic or mechanical systems;
- medical systems and manufacturing systems that require special safety-related documentation.

This British Standard gives guidance of two types and is structured to reflect this.

- a) Guidance on process is given in sections 2 to 8.

Section 2 gives an overview of the design and development process.

Sections 3 to 8 cover the separate stages and can be referred to when the stages are being carried out. There are checklists in annex B which can be used to check and record progress.

- b) Guidance on detailed design is given in sections 9 to 14.

Sections 9 to 13 cover the content, structure, style, technique and presentation of documents. Section 14 covers contents lists and indexes.

During each stage, the detailed design sections can be used as reference material.

This British Standard takes account of the fact that some of the documentation will be provided by the software itself, possibly as on-screen displays. It describes how to establish what information users need, how to decide which information should be provided on paper, and how to prepare documentation containing that information. It does not cover the design and preparation of the software's screen displays, nor of other on-line documentation.

1.2 Informative references

This British Standard refers to other publications that provide information or guidance. Editions of these publications current at the time of issue of this standard are listed on the inside back cover, but reference should be made to the latest editions.

1.3 Definitions

For the purposes of this British Standard the following definitions apply.

1.3.1 process

Activities for designing, developing and producing user documentation and for describing the processes to be carried out for each activity.

1.3.2 detailed design

The content, structure, style, technique and presentation of documents.

1.3.3 user documentation

Paper documents that are supplied with software to enable users to use the software.

1.3.4 document

A single item of user documentation such as a manual, a leaflet or a card.

1.3.5 product

A complete set of computer programs, procedures and associated documentation and data designed for delivery to a user.

NOTE. Also referred to as a software product.

1.3.6 software

That part of a product that is the computer program or the set of computer programs.

1.3.7 user

A person who uses a product.

1.3.8 computer

Any computer or computer terminal that users of a product use to communicate with the software. This may or may not be the computer on which the software runs.

1.3.9 project

The task of developing a product.

1.3.10 change control procedures

Control procedures that identify, document, review and authorize any changes to a product being developed. They ensure that the validity of changes is confirmed, that the effects on other items are examined and that those people concerned with the development are notified of the changes.

1.3.11 project manager

The person with overall responsibility for the management and running of a project.

1.3.12 product authority

The person with overall responsibility for the technical capabilities and quality of a product.

1.3.13 author

A person designing or developing user documentation.

1.3.14 technical contact

A person responsible for providing an author with technical information about a product or for checking the technical accuracy of drafts of user documentation.

Section 2. Integrated design and development of software products

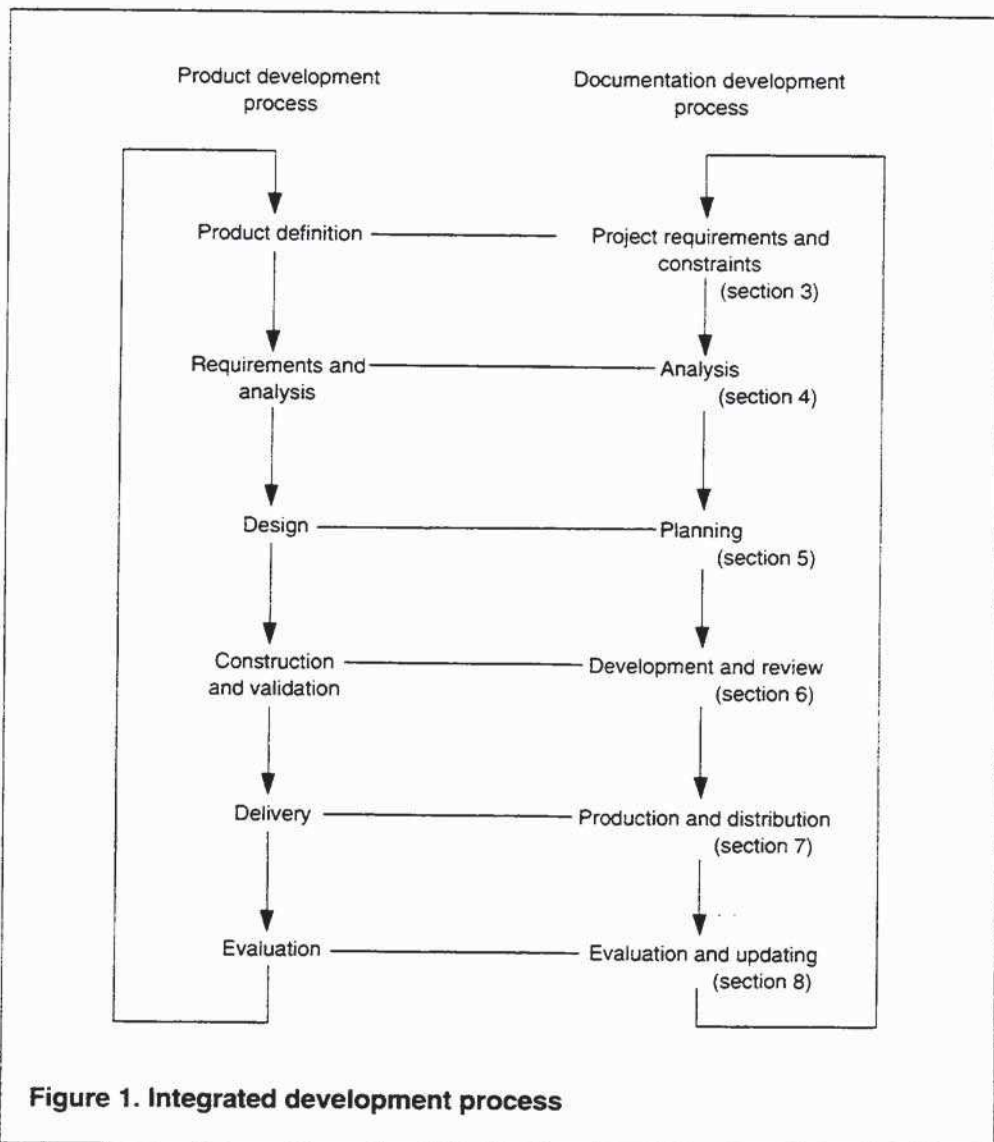
2.1 Integrated development stages

2.1.1 New products

If the separate components of an application software product are to appear to users as a coherent package, then all the components need to be designed together.

For new products or for new versions of existing products, therefore, the design and development of documentation should be a part of the design and development of the product as a whole, and not a separate exercise. Authors should be included, as equal members, in the team that is designing and developing the product.

Figure 1 illustrates the overall software development process, showing the documentation activities that are carried out at each stage. Whilst different companies call software development stages by different names and arrange them differently, they all carry out the activities suggested by the stages used in the illustration.



Documentation activities should be carried out in parallel with the software development activities as illustrated in figure 1.

Although authors should be involved throughout the development cycle, the work might not be spread evenly over time. For example, it is likely that the majority of the documentation development work will be needed after the design of at least part of the software has reached a reasonably stable state.

Project managers should be aware that, if documentation considerations are not taken into account in the design of the software, more user documentation might be needed than otherwise.

Project managers should ensure that authors are involved at all stages and are kept informed of progress. Change control procedures should address the needs of authors. Also, the author might need to press for particular software design decisions to be made earlier than the software development itself would require, where those decisions are important for the documentation.

If the product development plan involves building a prototype of the software, the processes covered in this British Standard can be used for developing associated prototype documents. When the prototype of the software is tested, the prototype documents should be tested too.

The following guidance should help ensure that an application software product gives users a consistent and well-designed image and user interface.

- a) Work done in the initial stages of product design to define user requirements should cover the entire product, that is, the software, the documentation, the training and the support. Otherwise, extra work will be needed to define requirements for documentation, and there is a risk that the resulting product will present an inconsistent user interface.
- b) Authors should be consulted about the design of those aspects of the user interface that will affect the way the software is documented, such as menus and screen displays.

2.1.2 Existing products

If documentation is being produced for software that already exists or has already been designed, the documentation activities needed cannot be carried out alongside the software development. This is the case if, for example, tailored documentation is being produced to support a particular implementation of an existing product. In such situations, the documentation design and development activities recommended in this standard still apply, although the spread of effort needed will be different from that for a completely new product.

2.2 Responsibilities for documentation activities

Where possible, the responsibility for the preparation of user documentation should be given to trained authors, rather than to software designers or software engineers.

If it is not possible for trained authors to be used, the responsibility for documentation should be given to someone other than those producing the software, for example those responsible for training or support or those who are in contact with existing or potential users.

If trained authors are to be brought in, checks should be carried out to ensure that they are suitable for the work to be done, in the same way as other suppliers are checked. The checks should include reviewing samples of the authors' work or samples that the authors believe are good.

The overall design of all the documentation for one product should be the responsibility of just one person. Although several authors might be needed for the documentation development activities, overall responsibility for the development of all the documentation for a product, including on-line information, should be given to just one person; that person should ensure that the guidance and recommendations of this standard are followed and that local standards and procedures (see 2.3) are followed.

2.3 Local standards and procedures

Local standards and procedures should exist for documentation development for each project. The local standards and procedures should be based on the processes recommended in this guide and should give the detailed procedures, standards and conventions that apply to the project.

Particular attention should be paid to the quality management system being used for the project, to ensure that all the features that affect the development of user documentation are reflected in the local standards and procedures.

Local standards and procedures should cover at least:

- detailed procedures for all activities;
- job titles or names of people with specific responsibilities within the project;
- details of any specific standards or conventions that have been set following the guidance in this guide.

2.4 Documentation processes

Figure 2 shows a recommended set of activities for designing, developing and producing documentation. See 2.5 for guidance on the sequence for carrying out these activities.

For documentation, as for software, as much planning and design as possible should be done before any development work is carried out. In this way, designs can be checked against the requirements and can be altered as necessary without wasting development work.

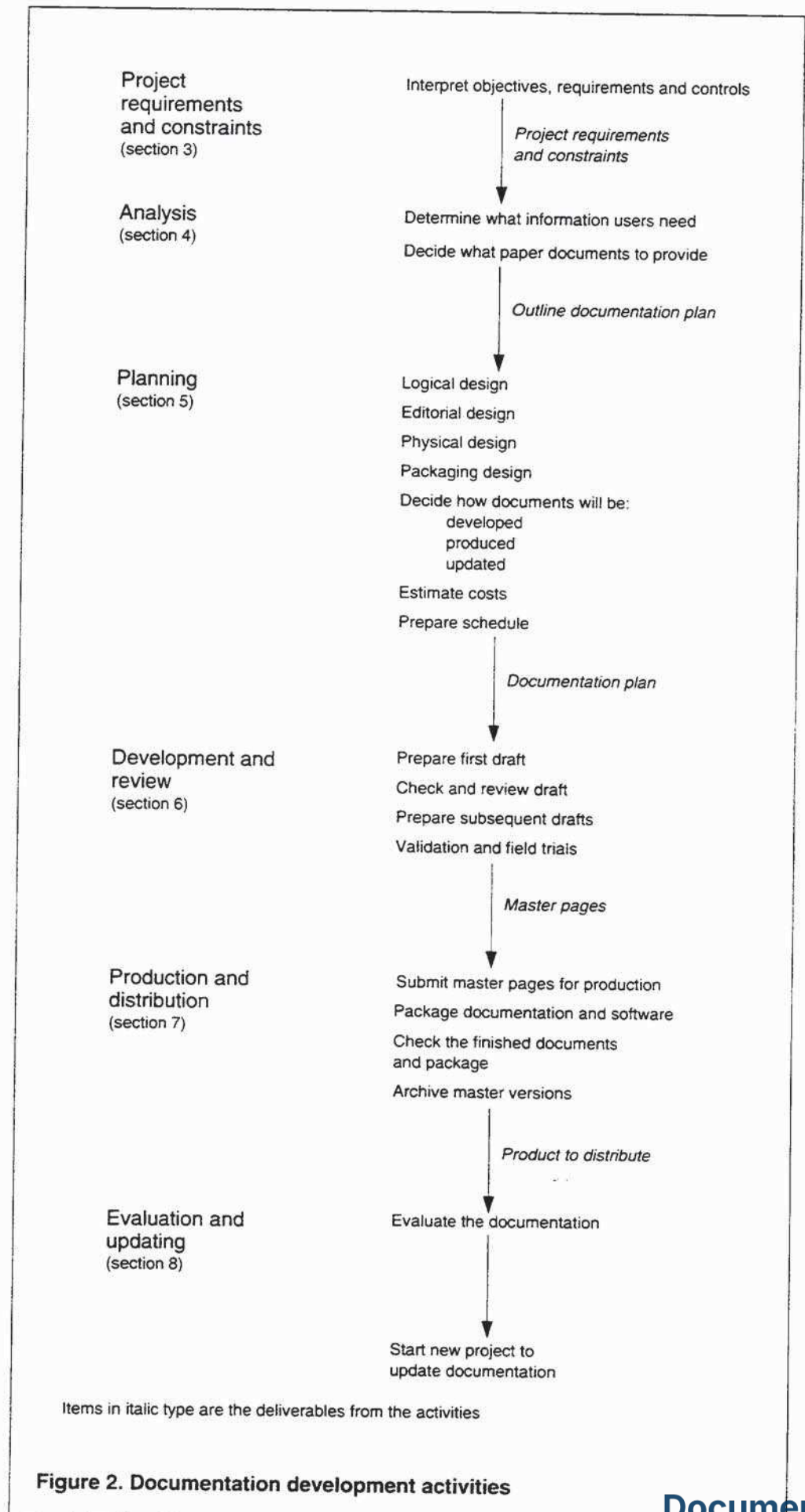
There is sometimes a temptation to neglect the user documentation design stages and simply edit the software's internal system documentation for users. It is unlikely that this will give the best solution either for the product developer or for the users. The design stages for user documentation should be carried out fully, to specify exactly what information should be given to users in what form. The resulting documentation is likely to be far more concise than it would otherwise be.

2.5 Sequence of documentation activities

Documentation design and production activities should not be carried out in a simple sequence because each design and planning decision is affected by other decisions and itself affects other decisions. For example, decisions about page layout are affected by decisions about content, structure and style of a document and the results of estimating costs might affect the decisions about how documents will be produced.

The development of documentation should also be an iterative process in which drafts are gradually refined until they are technically accurate and match the users' needs.

If there are significant changes to the software design at any stage, the documentation plan should be altered, following the project's change control procedures. In particular, if the software design changes during the development stage, a complete review of the documentation plan should be carried out; authors should not continue developing user documents based on an out-of-date plan.



Section 3. Project objectives, requirements and constraints

3.1 General

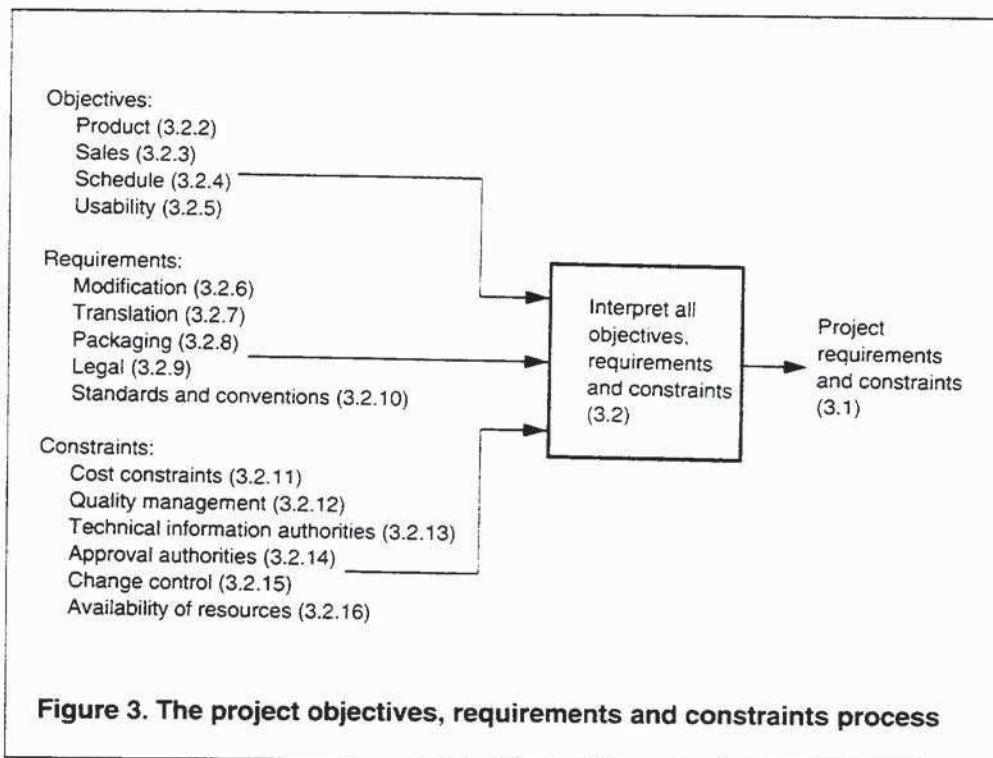
At the start of the documentation design, a framework should be established defining the requirements for the documentation component of the project and the constraints under which the documentation component has to be carried out. To establish this framework, the following should be collected and analysed to determine their effects on the documentation component of the project:

- the project objectives;
- the requirements and objectives for the product;
- the constraints under which the project is to be carried out.

As a result of the analysis, a list of requirements and constraints for the documentation should be prepared, specifying the source of each one. All the assumptions being made about the product and the project should also be recorded.

Figure 3 shows what information should be collected, what activities should be carried out and what output should be produced.

The checklist in **B.1** can be used to record the progress of the project.



3.2 Collect and interpret project requirements and constraints

3.2.1 General

The types of objective, requirement and constraint that should be considered for every project are shown in figure 3 and are covered in **3.2.2** to **3.2.16**.

If any objectives, requirements or constraints of types not covered by **3.2.2** to **3.2.16** are specified for the documentation, they should be investigated in detail to determine their effect on the project. The following points should be taken into account.

- Specific requirements for a maximum number of pages that should be produced, or for a document with a particular title, such as a 'user guide' or a 'reference card', might be based on a cost constraint or a schedule constraint, or on a market requirement. However, if they are based solely on a personal preference, they should not be treated as requirements for the documentation.

- Vague requirements for the style and content of documentation, for example for 'lots of pictures' or for 'not too much detail', often resolve into statements about the expected user profile or performance targets. Such requirements should be interpreted and checked thoroughly to ensure that they are accurate and realistic.
- Authors are sometimes asked to write documentation that can be used both on paper and as on-screen help text. Since the design criteria and user requirements for paper documentation and for on-line information are very different, each type of user documentation should be designed specifically for its own presentation medium. Information should not be duplicated unless there is a specific user need (see 4.1). If some text of the same kind is needed on both media for users carrying out the same tasks, then that text might be used on both types of documentation, in each place structured appropriately.

If authors believe that some of the stated requirements limit the design options too strictly, so that users cannot be given a suitable set of documentation, they should question the requirements, explaining their reasons and, where possible, suggesting some alternative solutions.

The author who is designing the documentation should use the flexibility that is allowed to the full, in order to give users the most convenient documentation possible. For example, it should not be assumed that all user documentation has to be in the form of a book or a card if this is not a stated requirement; other forms of paper documentation should be considered, such as keyboard templates, wall charts and labels, if these will better match the users' needs.

3.2.2 Product objectives

The overriding project objective should be the product itself. Other constraints for the documentation should not be allowed to alter or adjust the aims of the product in terms of the real-world problems that it should solve and the range of hardware and operating systems for which it is being developed.

For example, cost or schedule objectives should not be allowed to compromise the project to such an extent that users are given software or documentation that does not satisfy their needs.

3.2.3 Sales objectives

For documentation, the important features of the product's sales objectives are the target quantity of the product over its projected lifetime and the expected profile of sales over time: will demand be steady or will there be a period of peak demand?

3.2.4 Schedule objectives

When the project schedules are being defined, the following should be taken into account.

- Documentation development cannot be finished until after the software design is frozen.
- During the development stage, time is needed for authors to use the software or prototypes of the software.
- The documentation development schedule will impose constraints on the schedule for the rest of the project, in particular to allow time for technical contacts to supply information and to check the accuracy of drafts. The amount of time needed for these activities is always significant and it is essential that adequate allowance is made for them in the project's schedules.
- Draft documentation should be prepared for inclusion in product validation exercises, field trials and usability trials, and the documentation development schedule for preparing such drafts will affect the timing of these exercises and trials.
- When the documentation has been developed, further time is needed for translation, if required, and for printing and packaging.

The author should work closely with the project manager to agree overall schedules for the project. Once the schedules are agreed, authors should design documentation that can be prepared within the required time.

3.2.5 Usability objectives

When the usability targets are set for the product, documentation should be treated as an integral part of the product. Similarly, usability of the documentation should be measured in conjunction with measuring the usability of the software.

Measures of the usability of the documentation, independent of the usability of the software, might include the following:

- a) the time taken to learn what the nature of the document content is, particularly if more than one document is supplied;
- b) the time taken to understand the document structure and to learn how to use it;
- c) the time taken to find information, once the learning in b) has been done;
- d) the time taken to learn how to do a specified task using the instructions in the document;
- e) the time taken to incorporate amended pages.

Measures of satisfaction for the documentation could use responses to questions such as the following.

- How satisfied are you with the layout of the document?
- How satisfied are you with the time that it took you to find the information you needed?
- Treating each topic separately, how satisfied are you with the completeness of the information?

Durability targets should be set for individual documents according to the type and amount of use they will have.

3.2.6 Modification requirements

Unless there are very clear reasons for stating otherwise, authors should assume that some modification of the product will take place after the product has been distributed.

The plans for making modified versions of the product available to users should be investigated, because they affect all aspects of the documentation design. In particular, the author needs to know whether completely new versions of the product will be issued or whether amendments will be made available to existing users.

3.2.7 Translation and national cultural requirements

The translation requirements, if any, should be established, giving a list of languages that are required. For some products that are to be exported, there might be a legal requirement for the documentation to be translated; appropriate advice should be sought.

Translation requirements should be taken into account throughout the project as follows.

- Documents should not be translated until they are complete and have been approved; that is, each document should be translated only once into each target language.
- Translations should be made by native speakers of the target language, not by native speakers of the source language.
- Where possible, the same translators should translate the paper documentation and the information that appears on the screen, including the screen displays, system messages and help texts, to ensure that they remain consistent. After the translations have been carried out, checks should be made to ensure that the software and its documentation still match each other.

- If the process of translation reveals inaccuracies or ambiguities in the source text, these should be removed at the earliest opportunity.
- Clarity and simplicity are particularly important in documents that will be translated. The following should be avoided: the vernacular, jargon, humour, idioms, and similes, analogies and metaphors that might not translate correctly.

For products that are to be made available in other English-speaking countries, decisions should be made about whether special versions of the documentation should be prepared for those countries, using appropriate language and conventions. The countries in which the product will be used should be considered, because there might be special cultural issues that need to be taken into account, both in the software and in the documentation, in particular in examples. Where possible, someone who is a national of the target country should check all the user documentation to ensure that it is suitable for use in that country.

3.2.8 Packaging requirements

The packaging requirements for the entire product affect the documentation design. Consequently, at least the following should be established:

- whether the software and the documentation are to be packaged together;
- whether there is a house style for packaging;
- what medium will be used to distribute the software;
- whether the packaging should be the same as that used for previous versions or for similar products.

3.2.9 Legal requirements

A software product may have to satisfy legal requirements. For the documentation, at least the following should be investigated to see whether they apply and, if so, appropriate legal advice should be taken:

- requirements set by national legislation;
- requirements set by EC directives;
- copyright status of the document itself;
- copyright issues for text included in the document from elsewhere;
- data protection;
- acknowledgements;
- trademarks;
- licensing;
- intellectual property rights.

3.2.10 Standards and conventions

The requirements for standards to be followed should be determined at the start of the project. Any of the following might apply; they should all be investigated:

- standards publications (see also the list of references inside the back cover of this standard);
- standards applying in countries where the product might be used;
- company or product visual identity standards;
- company or product editorial standards;
- industry standards;
- market norms for the type of product;
- conventions, for example, terminology used by target users.

Authors should check that any local, company or product standards are suitable by comparing them with the recommendations in this guide.

3.2.11 Cost constraints

To product development staff, documentation is often one of the least understood aspects of a product and the project manager is often not prepared for the costs involved.

The project manager should take account of the following.

- Documentation costs have two components, the design and development costs and the production costs, and the budget needs to allow for both.
- If documentation design is not taken into account in the design of the software, the documentation might not be adequately concise, and hence documentation development costs might be higher than necessary.
- Documentation production costs per copy of a product can vary widely according to the number of copies produced, the physical design of the documents and the packaging and the production method used; in general, the more complex and professional the finished documents, the more expensive the production.

To assist in the negotiation of documentation cost constraints, authors should obtain estimates of costs for carrying out documentation development and production activities for a variety of types of documentation. Recommendations for estimating costs are given in 5.9.

Once cost constraints are agreed, authors should design documentation that can be prepared within the allocated costs.

3.2.12 Quality management

All the documentation development activities recommended in this British Standard should be carried out under the control of the quality management system being used for the software development.

Users of this British Standard are advised to consider the desirability of quality system assessment and registration against the appropriate Part of BS 5750 by a third party certification body.

BS 7165 provides guidance on how the following aims may be achieved for software-based projects, to help those responsible to meet their commitments effectively, efficiently and economically:

- a) a clear definition of what is to be achieved and when (specifications and plans);
- b) a description of the activities and functions that need to be performed (codes of practice);
- c) the control and monitoring of the way in which activities/functions are performed (quality control).

3.2.13 Provision of technical information

At the start of the project, it should be agreed who is to be responsible for supplying to the author technical information about the software.

The methods to be used to supply information should be specified. All the following should be used at various stages through the project:

- consultations with technical contacts;
- written specifications;
- the software itself, possibly development or prototype versions.

3.2.14 Approval authorities

One person, the product authority, should have overall responsibility for the technical capabilities and quality of a product.

At the start of a project, the people who are to be responsible for the following should be identified and agreed upon:

- technical accuracy of the documentation;
- usability of the documentation with the software and vice versa;
- suitability of the documentation for the market;

- editorial quality of the documentation, including both text and illustrations;
- legal issues;
- production of multiple copies of the documentation;
- packaging of the software and the documentation;
- final approval of all the documentation.

3.2.15 Change control

Change control procedures set for a project should be in accordance with the quality management system being used (see 3.2.12) and should take account of the requirements of the documentation activities as follows.

- Because documentation can be greatly affected by changes to the design of the software or to the training or support plans, the implications of such changes should be assessed before approval is given.
- When changes are made, authors should be informed immediately, so that they are never working with information that is out of date.

Authors and project managers should be aware that small changes to a system can cause major changes to documentation, whilst major changes to a system do not always require major changes to the documentation. For example, altering a system menu might be a small change to the software itself but might have major implications for the entire structure of a user manual. In contrast, completely revising the methods used within the software for some complex operation might be a major task in developing the software but might not affect the user's view of the software at all, and therefore might need no changes to the documentation.

3.2.16 Availability of resources

The availability of resources needed for developing documentation should be established at the start of the project. All the following should be investigated:

- hardware and software for using prototype or other versions of the software;
- prototype or other versions of the software to be used by authors to obtain information about the software being documented;
- technical contacts who will supply information, discuss technical details with authors, answer technical questions, and check drafts of the documentation;
- authors or other staff to review drafts of documentation as recommended in 6.3;
- hardware and software for producing the documentation itself.

Section 4. Analysis

4.1 General

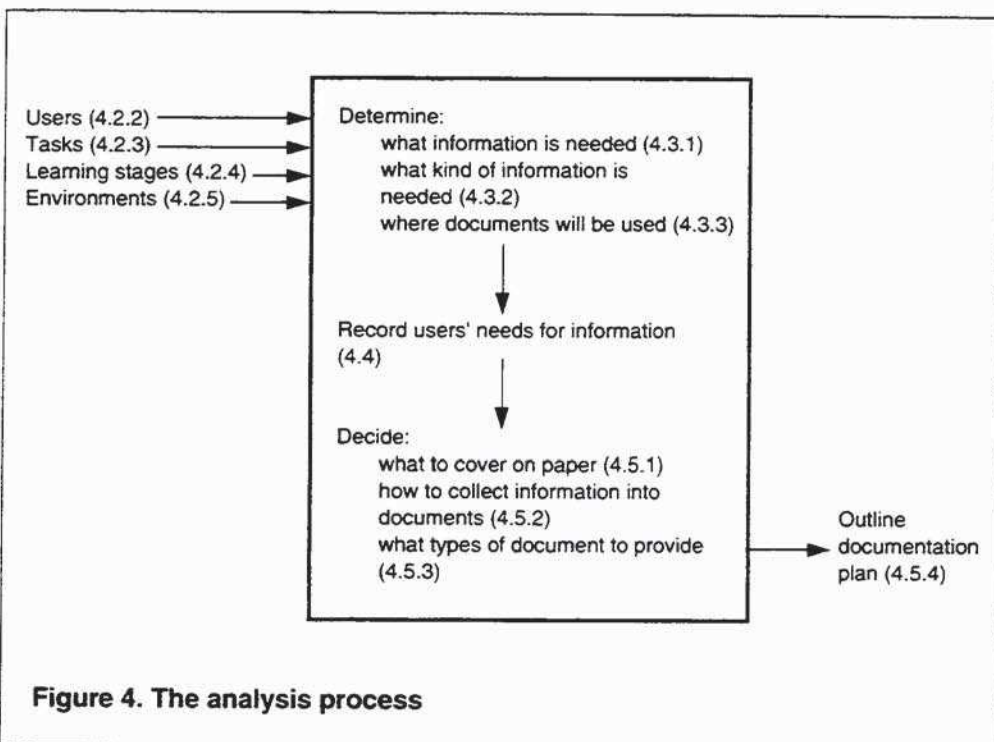
The aim of the analysis stage is to find out what information the users need and to decide what set of documents should be developed to meet those needs.

User requirements should be determined for the whole product together, including those necessary for designing documentation.

Similarly, the analysis of user requirements should address the whole product together, so that the software, the documentation, the training and the support complement each other and do not overlap unnecessarily.

Figure 4 illustrates the information that should be collected, the activities that should be carried out and the output that should be produced in the analysis stage.

The checklist in **B.2** can be used to record progress.



4.2 Collect details of users

4.2.1 General

Information should be collected about the users or potential users of the product and the tasks they will carry out with the product. The author should work with the others involved in the project to collect and agree the types of information described in 4.2.2 to 4.2.5.

If the product is being developed for a specific organization, or if the software has already been designed and documentation is being prepared for a specific implementation in a specific organization, then the information about users and their tasks should be collected directly from those users. In developments such as these, the information about users and their tasks is complete, so the analysis stage is greatly simplified.

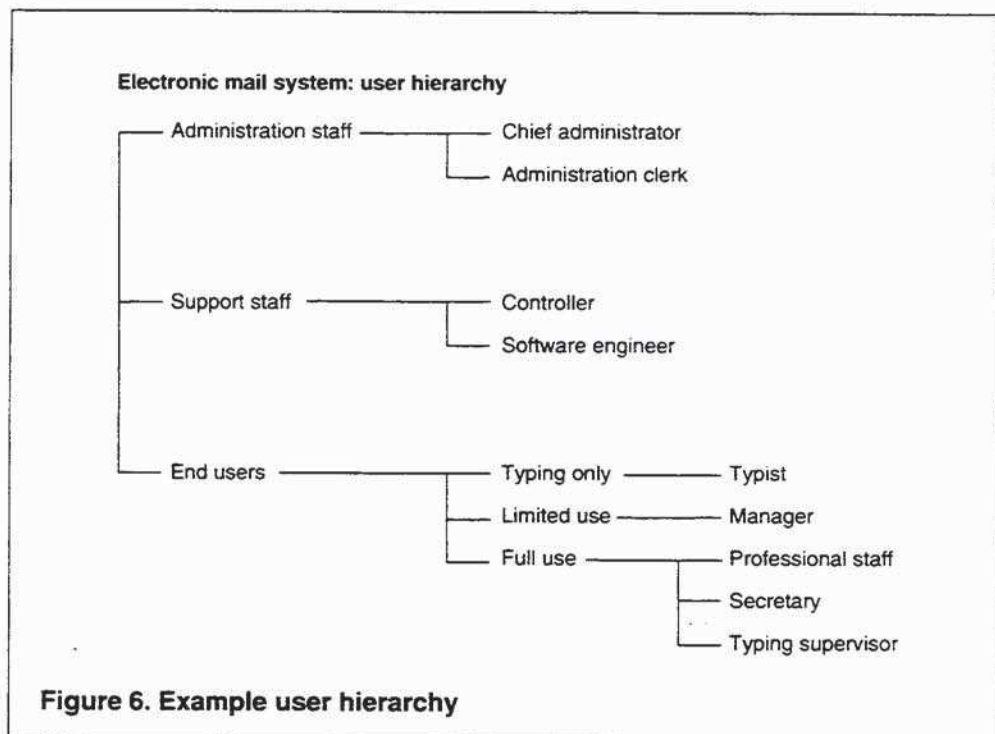
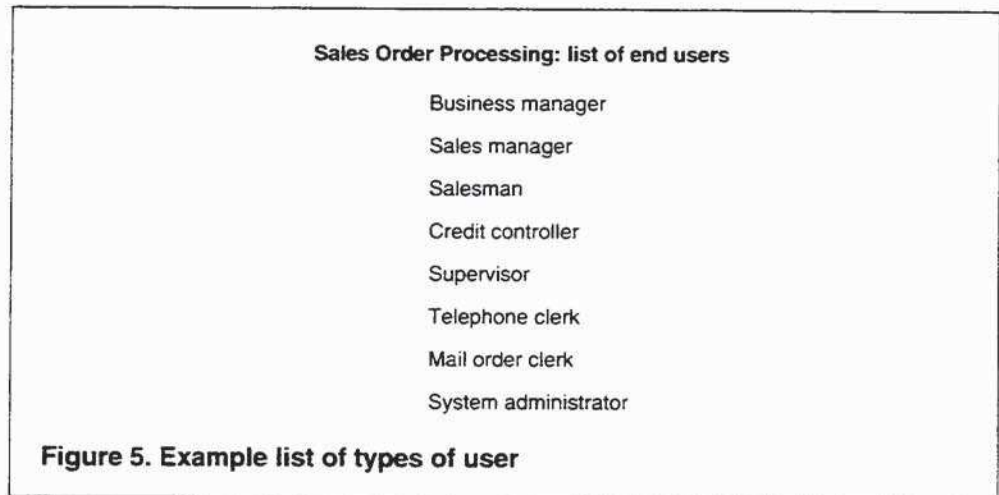
4.2.2 Users

A list should be drawn up of types of user who will be using the product. For designing documentation, it can sometimes be useful to group types of user into a hierarchy, so that individual documents can be targeted at several types of user.

The aim of the hierarchy is to group together types of user who will have the same

type of interaction with the software, not to reproduce the user organization's organization chart.

Figure 5 shows an example of a list of types of user for a sales order processing system. Figure 6 shows an example of a hierarchy of users for an electronic mail system.



For each type of user, a user profile should be drawn up giving details of those characteristics that will be common to all users of the type, such as:

- expected experience of computer systems;
- expected knowledge of the application topic, that is, of the real-world subjects handled by the software;
- expected knowledge of similar products;
- language and culture.

This information should be stated either in absolute terms, for example that users of one type will have special qualifications, or in terms of how users of the type vary from the norm, for example that users of one type have more experience of using keyboards than the norm.

If the product being developed is a consumer software package it is likely that it is for use by a single user; therefore, there will be no list of users and no user hierarchy. In this case, documentation design should be based primarily on the tasks that the user will carry out (see 4.2.3) and the learning stages through which the user will go (see 4.2.4).

Also for consumer software packages, the information about potential users might be vague; there might be a wide variety of types of potential user, with a wide variety of experience, skills and knowledge. In this case, details should be defined for the most naive of the potential users, and the documentation should be prepared for those users.

4.2.3 Tasks

For each type of user, a list should be drawn up of tasks they will carry out using the product. Where possible, the tasks should be collected into groups of related tasks, that is, tasks that are always carried out together, tasks that are similar or tasks that involve similar steps.

A task list for the system might not have the same breakdown as the list of users; however, a list of tasks should be devised for each type of user, even though this means that some tasks are listed for several different types of user.

To draw up a list of tasks, the real-world tasks that the user will want to carry out should be listed first, for example 'send a mail message' or 'amend an order'. These real-world tasks should then be broken down into the actions that match the functions provided by the software, for example 'create a document'.

When drawing up the lists of tasks, authors should use the following techniques, if they are helpful.

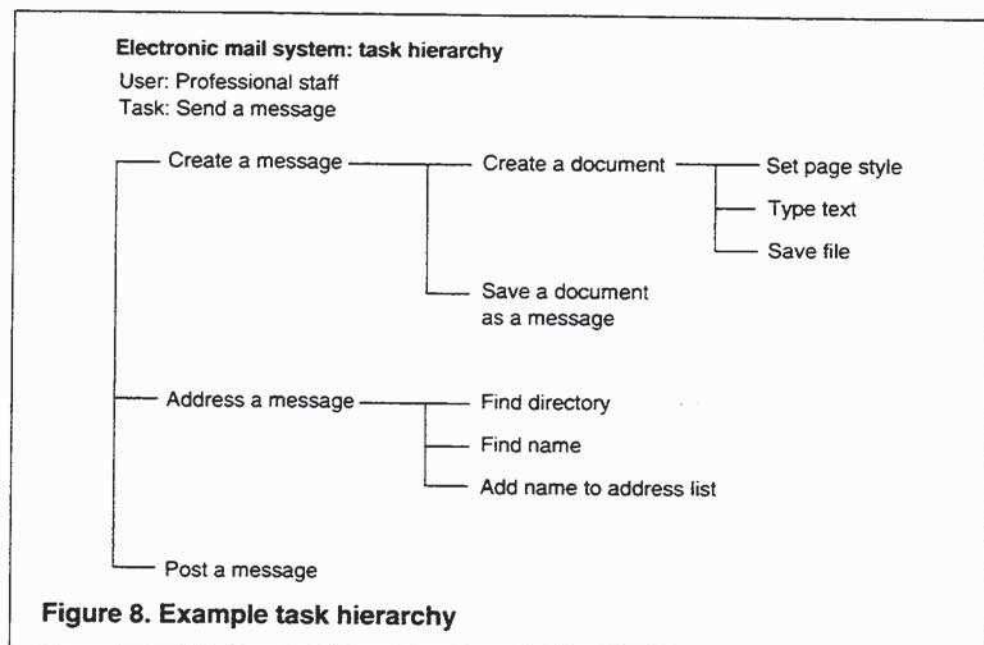
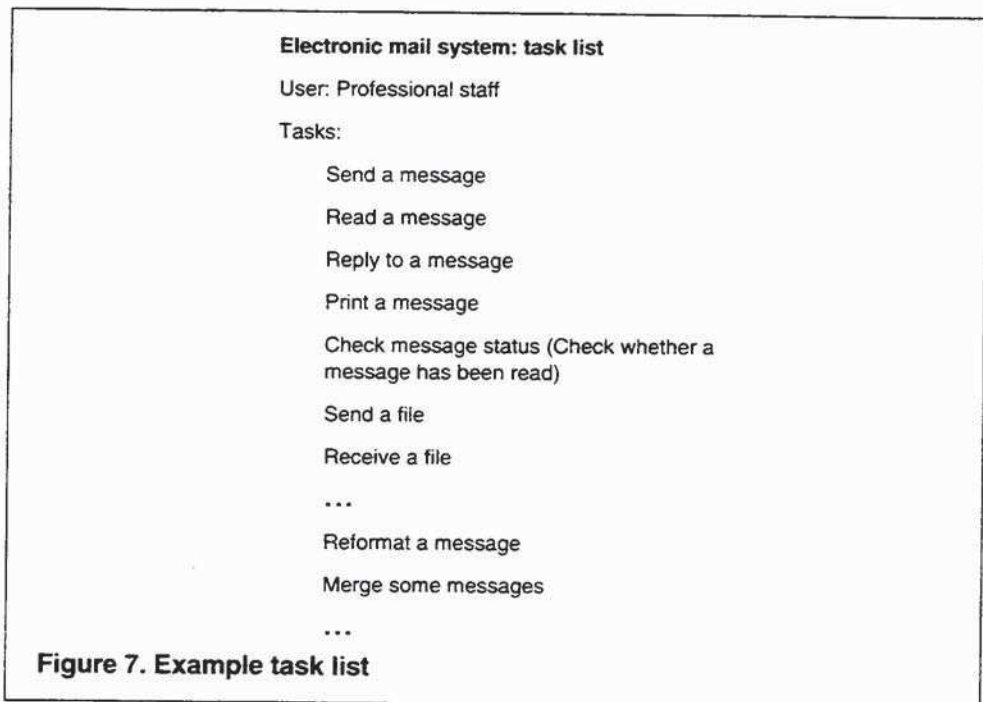
- Use as a basis the set of activities or goals for which the product is being developed, breaking down each activity or goal into the sets of tasks to be carried out by the various types of user.
- Define a set of scenarios where each scenario considers one type of user using the product to achieve one goal and draw up a list of tasks for each scenario.
- Consider categories of task such as evaluation, planning, installation, administration, operation, end-use and diagnosis and, for each type of user, list all the tasks that can be carried out.
- Use as a basis the functions that are directly supported by the software design, for example, the set of menus provided and the entries on those menus. To define the task lists for users, list the set of tasks that each type of user will carry out using each function.

By comparing the lists of tasks for the different types of user, the author should check that the user hierarchy is correct. At this stage, it often becomes clear how the hierarchy of users of the software differs from the user organization's organization chart. For example, for the electronic mail system used in figure 6, typists interact with the system in a completely different way from typing supervisors, although, on an organization chart, they would be in the same group.

If the following information is available and will be useful, it should be specified for each group of tasks:

- why the tasks are carried out;
- the sequence in which the tasks are carried out;
- what the prerequisites for the tasks are, for example what has to have happened before these tasks can be carried out;
- when and how frequently the tasks are carried out;
- what sub-tasks are needed to carry out the tasks.

Figure 7 shows an example of a part of a task list for the electronic mail system used in the example in figure 6. The tasks listed are the real-world tasks that a user will expect to be able to carry out using the software. The task list is for the professional staff. Figure 8 shows an example of one of those tasks, sending a message, broken down further into a hierarchy. The real-world tasks that a user might want to carry out, such as addressing the message, are broken down into the set of sub-tasks that the user will have to carry out using the software.



4.2.4 Learning stages

The way in which the performance of users of each type changes over time as they learn about and use the product should be analysed, because it will help to clarify what type of information users will need at various stages.

The following list gives an example of a set of stages that is suitable for the majority of products:

- before using the product;
- first-time use, normally referred to in documentation projects as getting started;
- normal use;
- exploitation of advanced features.

If users will have to install the software, this should be included as a separate stage.

Figure 9 contains an example of part of a chart showing what tasks different types of user carry out with an electronic mail system, annotated to show how the tasks are allocated to the different learning stages.

Tasks	Typist	Manager	Professional staff	Secretary	Typing supervisor
Create a message	S N		S N	S N	S N
Address a message			S N	S N	S N
Post a message			S N	S N	S N
Read a message		S N	S N	S N	S N
Reply to a message		S N	S N	S N	S N
Print a message	S N		S N	S N	S N
Check message status		N	N	N	N
Send a file			N	N	N
Receive a file		N	N	N	N
...					
Reformat a message	E		E	E	E
Merge some messages	E		E	E	E
...					

S = Getting started N = Normal use E = Exploitation

Figure 9. Example chart of users and tasks

4.2.5 Environments

For each type of user and each set of tasks, the author should establish what the working environment is or will be and what hardware will be available. The aim should be to collect as much information as possible about factors that will influence decisions about the types of document to provide. For example, if the software is to be used in a storeroom or warehouse, it is necessary to know whether documents will have to be carried around or whether there will be convenient places to store them.

The following are examples of the types of question that should be asked.

- What type of computer or terminal will the user have?
- Will users need to use a telephone or other equipment at the same time as the computer?
- Will the software be used in warehouses, storerooms or shops?
- Will the software be used in dirty, dusty or oily environments, such as workshops?
- Will there be any flat space to use for putting documents down?
- Will there be a convenient wall or other vertical surface such as a partition that can be used for displaying documents?
- What are the minimum hardware requirements for each type of user? For example, will all users need to have a keyboard, a mouse, a printer or any other device?

4.3 Determine what documentation users need

4.3.1 Decide what information is needed

The author should consider each user type and each group of tasks separately and decide what information is needed. For each, only the information that users actually need to carry out the tasks should be included. There may be much information available about the software and how it carries out functions, but it may not all be relevant.

To decide what information to give to users in paper documentation, it is necessary to know what information the software itself gives to users in the user interface.

If the product is being designed at the same time as the documentation, the decisions about what information should be conveyed by the user interface should be made at the same time as decisions about other documentation (see 4.5.1).

If a version of the software already exists, the author should seek full details of the user interface.

Figure 10 shows an example of part of a list of information needed by one type of user for one type of task; only the information the user needs is included in the list. The user shown in the example is the same user as in figure 8 and the task is to create a message.

As good an estimate as possible should be made of how much information users are likely to need, to help in working out how big documents are likely to be. The following techniques should be used if they are helpful.

- Count anything that can be counted, for example screen displays, keys, concepts, tasks, steps, transactions.
- Assess how much information is needed about each item, for example a page of awareness information for each idea, two pages of instructions for each task, one illustration and one page of text for each screen display, one page for each ten steps.
- Prepare a prototype document for one item and use its size as a basis for calculations.
- Refer to documentation for comparable users, tasks and products and assess from that how much information will be needed for the current product.

4.3.2 Decide what kind of information is needed

Users use information in a variety of ways, at a variety of times and for a variety of reasons. Users therefore need different kinds of information and authors should determine what is needed for each type of user and each learning stage. The following kinds of information are suitable for many products.

- *Appreciation or awareness information.*

This introduces the product to potential users. It tells them what the product can do and how it can be used, and helps them to decide whether the product is appropriate.

Electronic mail system: information needed

User: Professional staff

Task: Create a message

Information needed at awareness stage:

- How easy and fast it is to send a mail message and get a reply

Information needed at getting started stage:

- Basic skills
- Overview of the process
- How to create a message
- Sub-tasks:
 - Create a new document
 - Save the document with a chosen name, in a chosen folder
 - Set the margins
 - Type in text
 - Correct typing mistakes
 - Delete text
 - Insert text
 - Move text
 - Take a back-up copy
 - Print a document
 - Save a document as a message
- Fast keys
- Advice on saving files frequently
- Advice on checking documents before saving as messages

Information needed at normal use stage:

- Procedures for creating a message
- Fast keys
- Page layout options
- How to set up headers and footers
- How to use automatic page numbers
- ...

Figure 10. Example of information needed by one user type

- *Tutorial information.*
This teaches users how to carry out some tasks using the software, by taking them through all the steps in detail, teaching all topics from first principles.
 - *Quick reference information.*
This gives users a quick method of looking up factual details that they need when using the software.
 - *Comprehensive reference information.*
This is information that users need to look up while they are using the software. It includes:
 - task-related information, explaining what steps the user needs to perform to carry out specific tasks;
 - software-related information, describing or defining how the various components of the software operate or can be used.
- For many products, there is a need for both types of comprehensive reference information for various users.

Table 1 shows the kinds of information that are most likely to be needed by users at the different learning stages.

Table 1. Kinds of information for different learning stages				
Stage \ Document kind	Appreciation	Tutorial	Quick reference	Comprehensive reference
Before using the product	✓			
Installing the software	✓	✓		✓
Getting started		✓		
Normal use		✓	✓	✓
Exploitation of advanced features		✓	✓	✓

4.3.3 Decide where documentation is to be read

The author should find out where users will read the documentation. For example, appreciation or awareness information is rarely used by someone actively working at a computer although reference information frequently is, and quick reference information has to be available at a glance.

At this stage too, the storage and transport needs should be defined, in particular:

- where the documents will be stored;
- whether it will be necessary for several users to share a single copy of a document or whether users will have a copy each;
- whether documents will be used in several different places, and therefore need to be carried around;
- whether documents will be needed always and only near a computer.

4.4 Record users' needs for information

The user requirements resulting from the activities described in 4.3 should be recorded in a chart or table or by some other convenient method. Examples are shown in figures 9 and 10. Care should be taken to include all the requirements, because some method has to be found for satisfying all of them, not necessarily using paper documents.

4.5 Produce the outline documentation plan

4.5.1 General

Deciding what set of documents to provide is the most important part of the documentation design process because it determines how much user documentation there will be and how well the documentation will give users what they need.

The user requirements defined as a result of the processes described in 4.3 and 4.4 should be used to draw up an outline documentation plan, which specifies what documents will be provided.

Before it is decided what documents to provide, the requirements should be examined to determine the following.

- a) Will any of the requirements be satisfied by other parts of the product, such as the training, the user interface or on-line help texts? If so, is it necessary to repeat that information on paper?

For some information, it might be more convenient for users if the information is included on the software's screen displays, rather than being printed on paper. For other information, an on-line tutorial might be provided, rather than a tutorial on paper.

- b) Is it necessary or sensible to provide paper documentation to cover every user requirement? If not, which requirements are essential?

If there is a requirement for users of one type to have both tutorial and reference information about the same set of tasks, and if user documents were produced containing all this information, those users would have a large amount of paper documentation, containing much duplicated information. It might be better if those users were given tutorial information for only a few tasks, to teach the general principles, and reference information about all the tasks.

4.5.2 Decide how to group information into documents

The author is now able to decide how the information that users need should be put into one or more documents.

For software packages that are not aimed at a range of different types of user, the factors that should influence the way information is grouped into documents are: when the user will need the information and how the user will want to access the information, as follows.

- *Learning stages.*

Users need different information at the different learning stages, that is, before using the product, installing the software, getting started, normal use and exploitation of advanced features. Information for the different stages should be kept separate, either as separate major sections in a single document or in separate physical documents.

- *Kinds of information.*

Users want to access different kinds of information in different ways.

- Users normally read appreciation information when they are not using the software.
- Users want to follow the instructions in tutorial information one-by-one as they carry out the operations using the software.
- Users want to be able to look up reference information when they are using the software and to find particular topics quickly.

The different kinds of information, therefore, should be kept separate, either as separate major sections in a single document or in separate physical documents.

Table 1 can be used as an aid for deciding how information should be grouped into separate documents.

For software that is aimed at a range of different types of user, the following general guidance should be followed.

- Information for sets of users who carry out the same groups of tasks can be grouped into documents.
- Information for users carrying out different sets of tasks should be in separate documents.

Different kinds of information for one set of users can be put into a single document. For example, one document could contain some awareness information, some tutorial information and some reference information all for the same user, provided the different kinds of information are clearly separated and clearly differentiated.

When authors are deciding how to group information into documents, they should take the following guidelines into account.

- Information for different types of user can be in the same document provided that those users carry out the same groups of tasks in the same way.
- Different numbers of copies might be needed for different types of information. For example, each user might want a copy of quick reference information, but a single copy of a getting started tutorial might be shared between several users. Account should be taken of such differences before grouping different types of information into a single document.
- Information needed by users who work in different environments can be in the same document provided that the document can be designed to suit all the different environments.

- Secure information needed by some users should not be in the same document as information for other users. For example, it is normally not advisable to put information about administering the security of a system in the same document as information for end users.
- To avoid collecting too much or too little information into a single document, the estimated amount of information needed (see 4.3.1) has to be taken into account. In general, if the amount of information to be given in one document is too large, consideration should be given to splitting the document into smaller documents, for example:
 - keeping the appreciation and tutorial information separate from the reference information;
 - keeping different types of comprehensive reference information separate, for example, keeping task-related procedural information separate from software-related information.

Figure 11 shows an example of how the user needs resulting from the example in figures 6, 7, 8, 9 and 10 can be grouped into documents.

4.5.3 Decide on the type of document

For each document identified as a result of the process in 4.5.2, the author should decide what form of document is best, taking into account where the documents will be used and how they will be stored. The details in table 2 should be taken into account.

4.5.4 Write the outline documentation plan

The results of the activities described in 4.5.1 to 4.5.3 should be recorded in an outline documentation plan. Where possible, this outline plan should cover all documents, including screen displays and on-line help systems as well as paper documents.

For paper documents, the outline plan should simply list the documents, describing for each:

- its purpose, that is, the types of user, the tasks and the learning stages it supports;
- its contents;
- the kind of information it gives, for example, appreciation, tutorial;
- the type of document it is, for example, manual, card;
- the estimated size of the document.

For each document, a title should be derived from the name of the product and a suitably shortened combination of these features, that is:

product name, user types, groups of task, learning stages, kinds of information, type of document

For example:

‘PQRS product, manager, enquiries, normal use, quick reference, card’

shortened as a title could give one of the following:

‘Manager’s PQRS enquiry reference card’

‘PQRS Manager’s Enquiries’

Similarly:

‘XYZ product, all users, getting started tasks, getting started tutorial, booklet’

shortened as a title could give:

‘Getting started with XYZ’

This information is needed in the documentation plan (see 5.12) and is normally included in a document’s preface, foreword or equivalent section.

Figure 12 shows an example of the type of information that might be included in an outline documentation plan. The details are for the manager’s handbook identified in the example in figure 11.

The outline documentation plan can be regarded as the first stage in the development of the documentation. The planning and development stages that follow in sections 5 and 6 add more detail to this plan and more detail to the documents themselves.