HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Forest Products Technology

Jukka Saukkonen

STRUCTURED DOCUMENTATION PROCESSES – A CASE STUDY

Master's thesis for the degree of Master of Science in Technology submitted for inspection, Espoo, 29 November, 2000.

Supervisor Professor Pirkko Oittinen

Instructor Oskar Lönnberg M.Sc.(Tech.)

HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Forest Products Technology

ABSTRACT OF MASTER'S THESIS

Author Jukka Saukkonen	Date 29 November 2000 Pages 90
Title of thesis Structured Documentation Processes – A Case Study	
Chair Media Technology	Chair Code AS-75
Supervisor Professor Pirkko Oittinen	
Instructor Oskar Lönnberg, M.Sc.(Tech.)	

Two of the most important objectives of starting to use SGML in technical documentation are usually speeding-up and simplification of the documentation process. The purpose of this master's thesis was to find whether these objectives are actually met while migrating from the traditional word processing documentation to SGML-based documentation. Also the weaknesses of the new SGML-based process were identified for further development by conducting a usability study.

The theoretical study is based on general process theory and available software and hypermedia process research results as well as the knowledge of benefits and drawbacks of structured documentation implementations. Based on the theory, this thesis introduces the phases of both the legacy and new SGML processes from the technical point of view.

The delivery cycle of documents of both legacy and new processes were measured with five writers. The results showed that there was no statistical difference between the authoring environments, although the SGML measures show higher efficiency on average. The amount of new material written into documents was seen to correlate well with the authoring time spent per page in the legacy environment, but yet not significantly at all in the SGML environment. Time spent on background research seemed to correlate well with the amount of new material in a document in both environments. The total delivery cycle was shorter in the SGML-based process, as the number of local phases decreased. The more comparable and in the technical sense more important postprocessing time was also shorter in the new process.

The usability evaluation of the documentation process was done using observation and questionnaires with the same writers who participated in the authoring measurements. Observation and open questions revealed some advantages but also many weaknesses in the tools and the documentation process, which need to be solved in a way or another. Also quite many problematic areas emerged that can, fortunately, be solved with more training and practice. Closed questions revealed that the SGML tools are found usable, on average, but practical matters in the documentation process are still quite hard and some of them are not fully adopted. All in all, the SGML-based documentation process satisfied most of the participant writers quite well as such.

Keywords Structured documentation, SGML, technical documentation process, process measurement, documentation process usability	Language English

TEKNILLINEN KORKEAKOULU
Puunjalostustekniikan osasto

DIPLOMITYÖN TIIVISTELMÄ

Tekijä Jukka Saukkonen	Päiväys 29.11.2000 Sivumäärä 90
Työn nimi Rakenteiset Dokumentointiprosessit – Yksittäistutkimus	
Professuuri Viestintätekniikka	Koodi AS-75
Työn valvoja Professori Pirkko Oittinen	
Työn ohjaaja DI Oskar Lönnberg	

Kaksi tärkeintä päämäärää, joiden takia SGML:n käyttö teknisessä dokumentaatiossa yleensä aloitetaan, ovat dokumentointiprosessin nopeuttaminen ja yksinkertaistaminen. Tämän diplomityön tarkoituksena oli selvittää, saavutetaanko tavoitteet siirryttäessä perinteisestä tekstinkäsittelydokumentaatiosta SGML:ään. Myös uuden SGML-pohjaisen prosessin heikkouksia kartoitettiin käytettävyystutkimuksen avulla.

Teoreettinen tarkastelu perustuu yleiseen prosessiteoriaan ja saatavilla olevaan ohjelmisto- ja hypermediatutkimuksen tuloksiin, kuten myös tietämykseen rakenteellisen dokumentaation käytön eduista ja haitoista. Näiden teorioiden pohjalta tämä diplomityö esittelee sekä vanhan että uuden dokumentointiprosessin vaiheet teknisestä näkökulmasta.

Dokumenttien läpimenoaika mitattiin sekä vanhassa että uudessa prosessissa viiden kirjoittajan kanssa. Tulokset osoittivat, ettei kirjoitusympäristöjen välillä ollut tilastollista eroa, vaikka SGML-mittaustulokset kertoivatkin keskiarvollisesti paremmasta tehokkuudesta. Dokumenttiin kirjoitetun uuden materiaalin määrän nähtiin korreloivan hyvin sivua kohti kuluneen kirjoitusajan kanssa vanhassa ympäristössä, muttei ainakaan vielä merkittävästi SGML-ympäristössä. Taustatyöhön kulunut aika näytti korreloivan hyvin dokumenttiin lisätyn uuden materiaalin määrän kanssa molemmissa ympäristöissä. Vaikka kokonaisläpimenoaika olikin pidempi SGML-pohjaisessa prosessissa, paikallisten vaiheiden määrä väheni ja vertailukelpoisempi, teknisessä mielessä tärkeämpi paikallinen läpimenoaika myös lyheni.

Dokumentointiprosessin käytettävyyden evaluointi suoritettiin tarkkailuna ja arviointilomakkeilla niille samoille kirjoittajille, jotka osallistuivat myös kirjoitusprosessin mittauksiin. Tarkkailu ja avoimet kysymykset paljastivat joitakin hyötyjä, mutta myös monia ratkaisua kaipaavia heikkouksia käytetyissä SGML-työkaluissa ja dokumentointiprosessissa. Esille tuli myös monia ongelmakohtia, jotka onneksi voidaan ratkaista koulutuksen ja lisäharjoituksen avulla. Suljetut kysymykset paljastivat, että SGML-työkalut mielletään käytettävyydeltään kohtalaisen hyviksi, mutta dokumentointiprosessin käytännön seikat ovat vielä aika vaikeita eikä niitä kaikkia ole vielä täysin omaksuttu. Kaiken kaikkiaan, nykyinen SGML-pohjainen dokumentointiympäristö tyydytti suurinta osaa osallistuneista kirjoittajista varsin hyvin.

Avainsanat Rakenteinen dokumentaatio, SGML, tekninen dokumentointiprosessi, prosessin mittaus, dokumentointiprosessin käytettävyys	Kieli Englanti

PREFACE

This Master's thesis has been done in Pitäjänmäki, Helsinki, at the Customer Documentation section of the Professional Mobile Radio (PMR) division of Nokia Networks during the summer and fall of 2000.

I would like to warmly thank the supervisor of this thesis, Professor Pirkko Oittinen at Media Technology laboratory of Helsinki University of Technology, for her evaluation, instructions, comments and encouragement in the creation of the thesis.

I am most grateful to the instructor of this thesis, M.Sc. Oskar Lönnberg, for his patience and endless interest in the subject matter in the middle of his busy schedule. I would also like to thank Marketta Kirla, everyone in the PMR Customer Documentation section and other parts of Nokia Networks for comments and support throughout the creation of the thesis. Rami Sjöqvist deserves a special word of thanks for proofreading the thesis.

I wish to express my gratitude to Hannakaisa for the invaluable encouragement and support throughout my studies. I also would like to warmly thank my parents, Sakari and Eila, for supporting me in every possible way throughout all my years of studying.

Espoo, 29 November 2000

Jukka Saukkonen

TABLE OF CONTENTS

TERMS AND ABBREVIATIONS

3G	The third generation of mobile communications
Accessibility	Access by everyone regardless of disability
AHM	Amsterdam Hypermedia Model
ANOVA	Analysis of Variance
AUWG	Authoring Tool Accessibility Guidelines Working Group
BPR	Business Process Re-engineering
CALS	Continuous Acquisition and Life-Cycle Support
CDP	Customer Documentation Production
CGM	Computer Graphics Metafile
CMIF	CWI Multimedia Interchange Format
CMM	Capability Maturity Model
COMAS	Configuration Management System
CSS	Cascading Style Sheets
DIV	High level structural element which takes over the role of chapters, sections etc.
DMS	Document Management System
DOMAS	Document Management System
DSSSL	Document Style Semantics and Specification Language
DTD	Document Type Definition
DWB	Documentation Workbench

EORM	Enhanced Object-Relationship Model
FOSI	Formatting Output Specification Instance
GQM	Goal-Question-Metric
Groff	GNU Troff, or Groff, is a system for typesetting documents in the UNIX environment
HCI	Human-Computer Interaction
HDM	Hypermedia Design Model
HFPM	Hypermedia Flexible Process Modeling
HTML	Hypertext Markup Language
ISO	International Organization for Standardization
ISO 9001	A standard for the quality system of an organization
NED	Nokia Electronic Documentation
NET	Nokia Networks
NTC	Nokia Telecommunications
OO	Object Oriented
OOHDM	Object-Oriented Hypermedia Design Method
PDA	Personal Digital Assistant
PDCA	Plan-Do-Check-Act
PDF	Portable Document Format
PMR	Professional Mobile Radio
PS	Postscript
QIP	Quality Improvement Paradigm

R&D	Research & Development
RMM	Relationship Management Methodology
SEI	Software Engineering Institute
SGML	Standard Generalized Markup Language
SISU	Content plan (Sisältösuunnitelma). List of customer documents included in a release or other entity.
SMS	Short Message Service
SPICE	Software Process Improvement and Capability Determination
TIFF	Tag Image File Format
UML	Unified Modeling Language
VPS	Validation and Printing Service
W3C	World Wide Web Consortium
WML	Wireless Markup Language
WYSIWYG	What You See Is What You Get
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language

1 INTRODUCTION

1.1 The Research Problem

Technical documentation in Professional Mobile Radio has been written with Interleaf word processing application but there has been a need for developing the environment into on-line documentation. The documentation consists of documents about PMR infrastructure products. Thus, a project for converting the documentation into SGML has been started.

Two of the most important objectives of SGML migration are speeding-up and simplification of the documentation process. In this case, the documentation process is understood as all the phases in creating standard technical documentation. Professional Mobile Radio is interested in finding out whether the desired changes really occur. The usability of the documentation process and tools needed is also desired to be paid attention to because its weight increases while the number of technical writers increases.

1.2 The Theoretical Background of the Study

The study is based on the process theory, more exactly operational mechanisms of software processes and available research material on hypermedia processes. Hypermedia approach is included mainly because the upcoming on-line documentation issues. The emphasis is on technical documentation process issues. Both the authoring and publishing end of the process are considered. In order to measure and improve the documentation processes, also the existing, mainly software-oriented, process measurement and process improvement methods have to be investigated.

Usability and usability evaluation methods are examined in order to take usability issues into consideration when improving the documentation process and related tools. In addition, theories and standards of structured documentation are examined so that all the features of the new environments will be known.

Converting technical documentation into structured form should introduce many advantages. A more automated and rapid documentation process is considered an important value-added factor, in addition to generally recognised standards, good managing properties of large amounts of documents, retrievability, software and platform independence, reusability and communication channel independence. The control of the process is intended to improve the quality of documents and the productivity of documentation development.

1.3 The Research Method

The theoretical study is limited to the definition of a process and related concepts, as well as structured documentation and its differences to the traditional documentation.

The assignment, in addition to the process theory study, is to measure the actual time that the documentation process takes with all the intermediate phases. The concept of actual time differs from the elapsed calendar time. To succeed with measurements, the measurement points must be clearly defined. The documentation process can be divided into main phases that are writing and finalising the document. The total delivery cycle is made up of these two phases. By reviewing and comparing the measurement results, conclusions are made whether the total delivery cycle speeds up as has been predicted. Furthermore, the usability of the documentation process is considered for its methods and tools with a usability test.

1.4 The Goal of the Study

The study has two distinct goals. The first goal is to get an understanding of the rapidity and simplicity of the structured documentation process. This requires familiarising oneself with the present customer documentation processes, general process theory and research results of the usability field.

The second goal is to find out to what extent the migration to structured documentation has changed the documentation process in practice. In order to consider the use of structured documentation justified, the delivery cycle of a document should shorten. In the quick moving telecommunications industry the flexible and prompt ability to react to the customers' needs is important also for the documents part, which requires that the processes are appropriately optimised. At the same time, repeatability and predictability of the process betters so that the final result will always be equal in quality, no matter who are doing the actual work.

2 STRUCTURED DOCUMENTATION

2.1 Introduction to Structured Documentation

The structure of information can be defined as the role of information without the knowledge of how it is presented. In other words, in a structured document the logical and layout structures are separated but can be associated with the same information content. The same document can have many different layout structures associated with it depending on e.g. the publishing medium /55/. An example of a simple structure of a journal article is illustrated in Figure 1.

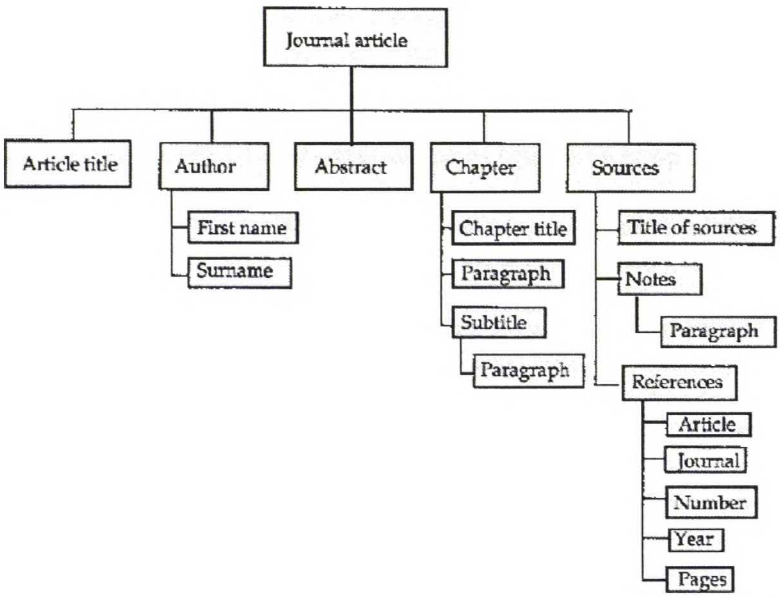


Figure 1. *The Structure of a Journal Article /16/*

Specific markup is the type of markup used in traditional text processing. The commands used by the language are specific to the text processing software and usually has to be re-entered in order for the text to be processed by a different text-processing program. Interactive word processors allow an author to add specific markups to documents while writing. This allows the generation of a visually pleasing document but the coding does not display the structure so, for example, from a centred element it can not be seen whether it is a paragraph or a caption. This is a drawback that affects to the availability of search engines for the documentation /7, 11, 17, 66/.

Another disadvantage is that style changes affect the markup process so that it has to be repeated to reflect the changes. Markuping with control words is time-consuming, error-prone and requires a high degree of operator training especially when complex typographic results are desired /11/.

A step closer to structured documentation is generic markup. Generic markup is the term that describes the process of assigning generic names to markup. Macros for typesetting languages were the beginning of generic markup. A macro is a software instruction, which executes a series of instructions. The style information on the appearance of a document is kept separate from its structure and content. With word processors that use style sheets, the same level of generic markup can be obtained. Figure 2 illustrates an example of a WordPerfect document revealing the style codes being used. A stylesheet describes the appearance of a document. Titles can be marked with a "title" style, paragraphs can be marked with a "paragraph" style and so on. The generic coding process is a single phase. The user recognises every element in the document and codes it with a generic identifier which describes it best. Code markups do not show how the context or parts of it are processed typographically. The processing information of the coding can be given case by case /7, 66/.

```
C:\WP\SGML\REVEAL.WP Doc 1 Pg 1 Ln 1" Pos 1" { [Style On:Title]ANGELICA[HRt] [Center]Angelica Archangelcia
L.[Style Off:Title][HRt] [Style On:Author]James M. Stephens[Style Off:Author] [Style On:Head
1]INTRODUCTION[Style Off:Head 1][HRt] Angelica is a European perennial plant ...
```

Figure 2. *An Example of Generic Markup /66/*

The needs for managing structured documents are increasingly broad and occur in many application areas. For that reason, there are several international standards developed for defining markup languages for documentation. Although the idea of generic coding was introduced already in the late 1960's, the ISO standard 8879:1986 for structured documentation was published in 1986. Since then SGML (Standard Generalized Markup Language) has been the actual metalanguage for describing structures of documents. Structured documentation is based on markuping that adds information to a document about the structure, where the markup may be used by a human or by a machine. The rules for the structure and the markup are given in a document type definition (DTD); the markup itself is not standardised. Various organisations have created their own DTD for specific purposes. The most important ones, also seen as de facto standards, of these are Docbook DTD developed by the Davenport Group and HTML (Hypertext Markup Language) DTD developed by W3C /17, 29, 37, 55/.

In the beginning of the standardisation process SGML was set some important criteria that it was desired to meet /55/:

- System-independence – The user is not tied to any specific vendor now or in the future.
- Device-independence – Documents are not related to any specific types of hardware.
- Language-independence – The standard is equally useable all over the world. It must be usable for Latin based alphabets and non-Latin based alphabets, as well as other forms of writing.

- Application-independence – The standard can be used for a wide variety of documents from many different sources. Documents can be very simple or very complex: technical or text containing a wealth of linguistic information.

During the development of SGML some other criteria were consciously also taken into account. Many of them gave SGML the flexibility and usability it is known for today. The most fundamental issues were /55/:

- Support for existing conventions; there has to be the possibility of effecting what is termed retroactive conversion of documents prepared using existing word processing packages.
- The inclusion of contents other than text, such as graphics or sound information.
- The markup has to be usable by humans and machines.

SGML standard /11/ states that "SGML can be used for publishing in its broadest definition, ranging from single medium conventional publishing to multimedia database publishing". In practical sense, SGML can be used for:

- Encoding documents
- Interchanging documents between several parties
- Working with documents in different ways, even taking information from tables and presenting it as business graphics
- Re-using elements of information many times on different occasions.

Professional publishing systems, such as SGML based publishing or desktop publishing systems, suit best for particular products or production issues in the technical documentation. Traditional word processing offers little in either sophisticated layout or powerful processing but may be useful in, for example, the initial editing phase. The following Figure 3 shows a comparison of these three publishing worlds in terms of complexity of page layout and volume of pages. The worlds overlap in the middle and continue to do even more so while the technologies improve /61/.

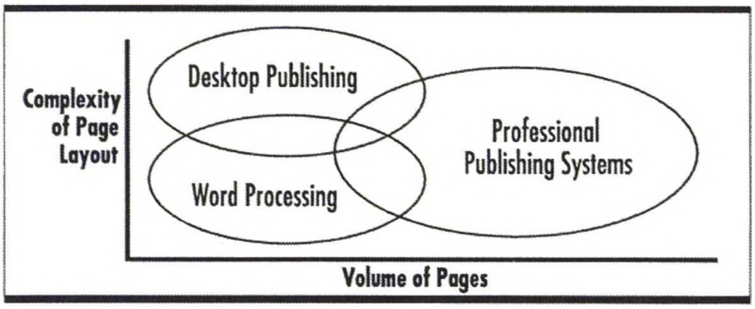


Figure 3. *The Worlds of Different Publishing Systems /61/*

SGML is quite complex to implement and contains a lot of features that are very rarely used. Its support for different character sets is also a bit weak, which is something that can cause problems on the web where people use many different kinds of computers and languages. It is also difficult to interpret an SGML document without having the definition of the markup language (the DTD) available /9/. Because of the complexity of SGML, related tools have been quite difficult and expensive to produce until the recent years /7/. For all these reasons, the Extensible Markup Language (XML) was developed. XML is a subset of SGML that is designed to make it easier to take in use and interchange structured documents over the Internet than with its predecessor. Unlike SGML, XML does not require the presence of a DTD. If no DTD is available, either because all or part of it is not accessible over the Internet or because the user failed to create it, an XML system can assign a default definition for undeclared components of the markup. XML is seen as the future format of structured documentation, which will displace SGML in the majority of applications in time /9, 29, 63/.

2.2 Benefits of Structured Documentation

The benefits of structured documentation, in practice SGML, used as technical documentation are very similar to the criteria that were set by the SGML development committee. SGML makes information independent of the software and hardware vendors as well as the publishing and delivery medium. Because of the standard status of SGML and its human interpretable file format, the SGML files can be used also in the future. Information, which can be a huge mass, is managed, updated and maintained in a neutral SGML database and can be delivered on paper, on CD-ROM or through networks /15, 29, 37, 55/. Also retrieval and browsing facilities are very advanced in the most of commercial SGML solutions compared to the traditional layout-oriented word-processing applications. Because of the document structure of SGML documents, the desired parts of documents can be found precisely. The retrieval procedure is similar to that of the database systems /7/.

An information unit or units can be re-used in any desired new combination that makes up a new document by any authorised user. Practically this means that work once done does not have to be repeated in order to author tailored documents for different purposes. Since the automation and smaller information units save time, money and resources as well as simplify the structure of documentation, the productivity of work can be seen to grow /7/.

Another viewpoint to the benefits of structured documentation is introduced by Elovainio himself in the same reference /7/. His classification is divided in three classes: reusability, availability and ability to check completeness. Reusability means ability to reuse the produced material in many purposes, for example producing several publications from an information database. Availability means the searchability features of structured documents. Capability of checking completeness stands for checking the completeness of structured documents by a computer program so that they do not include anything additional, out-of-the-structure data.

For many industries, technical information is part of a deliverable product, or a product in itself that must be rigorously maintained. SGML is useful as a tool in an integrated information management strategy. From a company's high-level management point of view its benefits are /2, 37/:

1. Increased productivity

This is achieved by organising the information in a meaningful hierarchy, with once-made document styles and software and hardware independent coding which eliminates the endless data translation cycles.

2. Reusability

Involves both human and machine readability features that help users to manage and maintain the different uses of documents.

3. Information longevity

Software and hardware independence assures the information viability.

4. Improved data integrity

The risk of losing information is reduced since no filtering to another formats is needed in order to have the data structured, readable and reusable.

5. Better data control

Information elements can be defined and manipulated with machine readable attributes for information management purposes. These controls can be used to hide and reveal information depending on the end user or automate the information flow, for example changes in the source data can trigger many changes in other applications.

6. Shareability

Documents can be built out of various document modules from all around the organisation. For example, a copyright statement may be maintained by the legal department.

7. Portability of information

Documents can be exchanged easily among different, networked environments.

8. Flexibility beyond traditional publishing

The variety of publishing media is wide and extended beyond traditional publishing.

9. Ability to participate in global markets

Using structured documentation will become a key benchmark of large businesses that are ready to do business internationally.

An important issue is that SGML is not concerned with the way or ways in which the information is presented. The writer does not have to concern himself with the layout rules during the process of authoring the context, but the case-specific layout is attached to the document afterwards /37/. There are two layout standards for presenting SGML documents, a ISO 10179 standard called Document Style Semantics and Specification Language (DSSSL) and a MIL-M-28001 included standard Formatting Output Specification Instance (FOSI) /37, 55, 61/.

The benefits of separating the structure and layout can be seen as follows: /12/

- When the structure for documents has once been defined, the layout can be added or updated automatically.
- When the layout is changed, every single document needs not to be updated separately.
- Consistency within the documentation is achieved with a company wide structure definition.
- Re-organisation of the structure is simple.

SGML should not be used as a formatting tool exclusively but as a structural tool as well. SGML becomes a way of enforcing style and template rules. The disadvantage of information modules that do not mandate a structure is lack of future searching and reuse capabilities /14/.

2.3 The Structured Documentation Environment

Structured, or in this case SGML, documentation environment requires knowledge of the documentation methods of the production unit and connections of documentation to the other parts of the information system. In SGML environment, it is not a question of authoring documents but managing the information process. While migrating to structured documentation, conversing, authoring, storing and delivery issues as well as connections to other information systems have to be solved /7/. As the case studies of Association of American Publishers and New Oxford English Dictionary /55/ revealed, the design of DTDs is an essential part of the application of SGML.

Clear and precise requirements for the system are essential for the development phase of documentation. The needed investments and benefits have to be based on exact expectations of the system. The requirements can be divided into business supportive and system based requirements. However, requirement specifications can not be set before the legacy system has been analysed thoroughly /7/.

During the migration, small-scale piloting of the first phase is also important. With piloting, the benefits of the new methods and tools can be evaluated. Also the usefulness of the present tools and methods in the new environment can be estimated. The best base for the implementation would be the design phase of a new product /7, 61/.

2.4 SGML Tools Requirements

Tools for structured documentation should meet some requirements in order to be usable in documentation systems /7/:

- An SGML product should implement the essential functionality and features of the SGML standard.
- Compatible documents should be transferred from one system or software to another without any conversion in between.
- Products should be completely tailorable by the users. This is possible by including a good application-programming interface in the product package.
- Products should be natively SGML-based so that there would be no need for conversions when opening or saving documents.
- Document type definitions should be applicable as is without changes in syntax or document structure.
- Stylesheets should be transferable between different applications without the creation of any application-specific presentation models.
- Information format of products should be compatible.
- Products should be combinable with other SGML or non-SGML products.

When authoring tools are considered, there should be a possibility to at least make cross-references, prepare indexes, make various kinds of lists, insert logically structured tables, insert bitmap or vector figures, represent mathematical formulas and equations and make hypermedia links /55/.

2.5 Reasons and Goals for the SGML Migration

Continuos development of the documentation has been seen as an essential factor in the customer delivery process. Consistent and well-managed documentation enables more flexible and shorter delivery times. For that, an electronical document archive is a functional solution /7/.

The essential goals for the development of customer documentation are the following /7, 37/:

- Consistent structured documentation makes it easier to author and postprocess information
- Consistent customer documentation speeds up the customer deliveries
- Common use of information from a common database betters the availability of information and updateability of information is thus easier
- The more centralised utilising of information produced in different units betters e.g. composing and delivering information.

Documentation is a part of the product and it is authored continuously in all the phases of the development process. In addition to the documents themselves, the product also affiliates with administrative information such as versioning and metainformation /7/.

The development of documentation requires rationalising and re-developing the methods and possibly choosing new tools. Primarily aspects of concern are efficient production, reusability and delivery of information. When the research on documentation development was started in Nokia Networks, every one of these alternatives was considered. The conclusion was that a standardised way of presenting information assures it a long life cycle, reusability and software and device independence. Also, SGML could immediately be adopted into use, since SGML related tools, such as editors, databases and parsers were already available /7/.

From its own perspective, the NTC Pilot project presented by Elovainio of VTT /7/ was considered to have the following SGML advantages:

- The present device environment can be used after the SGML migration and the present text-processing tools do not have to be abandoned completely
- The documentation can be structured with the current specifications
- Documents can be delivered in SGML format, as the customer needs it.

SGML has some qualities that make it a justified choice as a documentation method. It assures that the documentation is coherent as well as allowing contextually and layout-wise tailored publications for many different media. The information also has a long life cycle /7/. SGML also suits well for maintaining technical documents which should always be up-to-date /16/.

3 A Survey of the Process Theory

3.1 Definition of a Process

According to Humphrey /21/, "a process is a defined set of steps to accomplish a task. A defined process is one that is described in sufficient detail so engineers can consistently use it. Defined processes help in planning and doing a job." Kontio /27/ has also listed a wide variety of matters the word "process" can refer to:

- A class of similar flows of events
- A single occurrence of events (i.e., a project)
- A technology or a way of doing things (e.g., a methodology)
- The set of activities, people, and artifacts needed to produce a result.

Technical documentation process is, when applying the definitions of Kontio, a set of activities needed to produce a document release. On the other hand, technical documentation improvement process is, strictly speaking, .

A plan is not the same as a process. A plan includes estimates of the size of the product to be produced and how much time is need. It also includes a schedule of project steps and an estimate of possible risks involved. Each plan is unique to a specific job. Even when you have a defined process, you still need a plan for each project. When a process is designed, it is done on the basis of the typical project structure /21/.

A process consists of various activities. To manage these activities, they are combined into manageable groups called subprocesses or phases, which combine to form the overall development process. By splitting a process into phases helps people to better conceptualise what needs to occur and what are the relationships between various activities /21/.

Some other terms related to the term process need to be clarified and differentiated /27/:

- Process instance: actual set of activities that are enacted.
- Process class: a set of processes that have similar characteristics.
- Process model: (normally) a representation of a process class.

Adapting to a suitable development process consists of three different identifiable levels: the first is to identify the required process stages, the second is to choose a development process framework within which these stages can be placed and the third is to select the methods used to implement these stages. The selection of development stages will be based on the particular activities that must be carried out. In general, however, almost all processes will incorporate analysis, design, implementation, evaluation and management stages. A more difficult task is the identification of a suitable process framework for the development stages. The specific choice will be dependent on a number of factors. The implementation methods that can be used to support design activities are desired to be business-specific /33/.

The first goal of this study is to find out how rapid and simple the documentation process really is. One manner to measure those factors is getting a clear view of the case-specific process steps and conducting process measurements to them.

3.2 The Importance of Processes

A process provides guidelines for doing the work. A process is more than an aid for planning and doing a task, it is also a framework for learning. With a process, one has precise knowledge about the work and data helping to plan and improve. One also works more efficiently because there is no need to stop and think what to do next and with whom. The quality can also be assured when process elements are controlled /21/.

Processes are needed in today's world, because of increased competition in the business world, to streamline the corporation: eliminate redundant work, resolve bottlenecks and focus effort on critical areas. Also customers have higher demand for quality and service is expected to be a part of the product. Processes oftentimes automate the work that can increase white-collar productivity and better the predictability of process performance /27/.

In business companies, vertical organization creates interface overhead, sub-optimization and diffuses responsibility. In reality, work is organized and customers are served by a process that crosses functional boundaries repeatedly. Figure 4 illustrates the horizontal processes of working in an organization /27/.

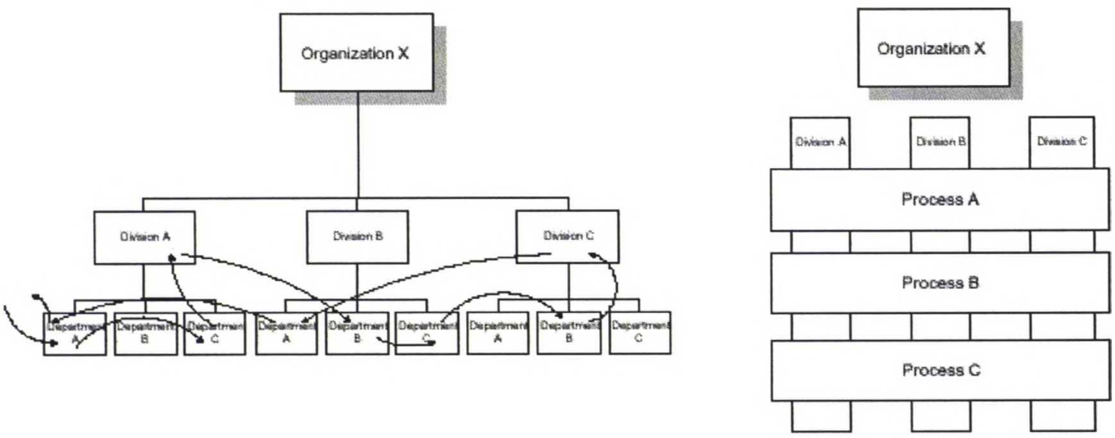


Figure 4. *Vertical Organisation vs. Horizontal Processes /27/*

3.3 Workflow

Workflow is defined as the computerized facilitation or automation of a business process, in whole or part. In other words, workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules, to achieve, or contribute to, an overall business goal or result. Workflow may be manually organized but in practice most workflows are organized with computerized support /18, 61/.

Though work flows horizontally, the process organization does not lack vertical hierarchy. The hierarchy is defined in terms of actions, which in process organization are: /32/

- Task
- Activity
- Sub-process
- Process.

Tasks are defined as elemental work or discrete elements of work. Activities are defined as groups of interrelated tasks. A sub-process is formed from a set of activities and a process is respectively a collection of sub-processes /32/.

Workflow is often associated with Business Process Re-engineering (BPR), which is concerned with the assessment, analysis, modeling, definition and subsequent operational implementation of the core business processes of an organization or other business entity /18/.

An individual business process may have a life cycle ranging from minutes to days, or even months, depending upon its complexity and the duration of the various constituent activities. Such systems may be implemented in a variety of ways, use a wide variety of IT and communications infrastructure and operate in an environment ranging from small local workgroup to inter-enterprise /18/.

3.4 Software Process Models - Overview

Following Kontio's /27/ definition, a software process is a set of activities needed to produce a software work product. Also, a software process model is a generic description of a class of software processes. A software process model tries to address most or all aspects of software engineering.

All the most relevant software process models consist of the same major phases and their outputs. A software process starts with a feasibility study, of which a business case is the result. The business case describes the business opportunity and motivation. The second phase is requirement analysis specification, which defines the user, or customer, requirements. The third phase outputs an architecture specification, which defines the high-level structure of the system. The system and architecture are implemented according to the design specification, which is done before the coding phase. The last phase is testing of the product /27/. The process is illustrated in Figure 5.

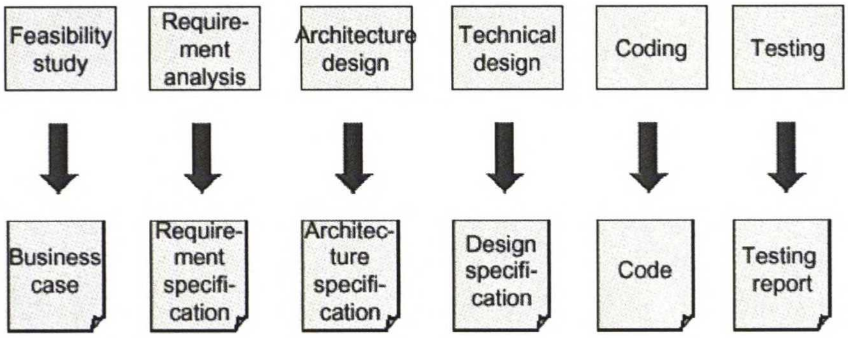


Figure 5. *Software Development Process and Artifacts /27/*

Generic software process models, classified by Sommerville /56/, are code and fix, the waterfall model, evolutionary development, formal transformation and reuse-based development. Code and fix is simple and common. It is an ad hoc combination of informal design, code, debug, and test until the product can be released. The waterfall model is suitable for well-understood but complex and large projects. It has separate and distinct phases of specification and development (Figure 6). It is possible to go back to a previous phase to change the specifications but it may be critical for the project. Evolutionary development is done incrementally developing program modules or through-away prototypes. Prototypes are done to clarify user requirements.

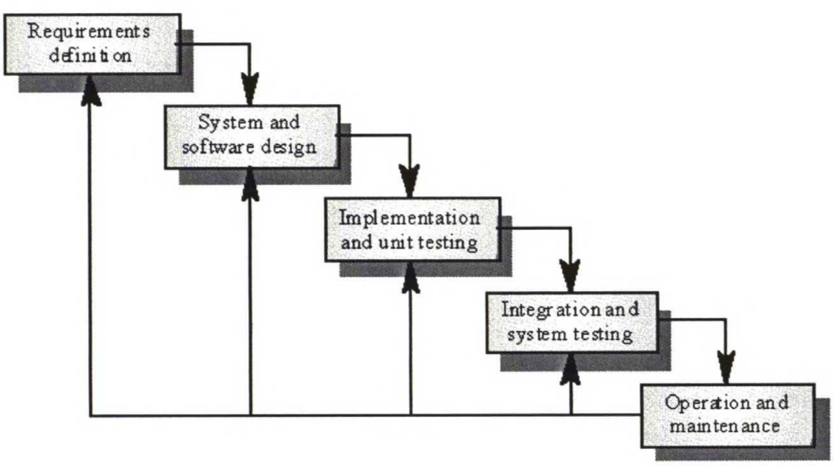


Figure 6. *The Waterfall Model /56/*

Large systems are usually made up of several sub-systems. The same process model need not be used for all subsystems. Prototyping suits for high-risk specifications and waterfall model is good for well-understood developments /32/. A risk-driven approach, called the spiral model, has been developed by Barry W. Boehm /3/. It is based on risk assessment after every development phase, as shown in Figure 7. The radial dimension of the model represents the cumulative costs when finishing the steps. The angular dimension represents the progress made in completing each cycle. Each loop of the spiral from -x-axis clockwise through 360 degrees represents one phase. One phase is split roughly into four sectors of major activities which are objective setting, risk assessment and reduction, development and validation, and planning for next phases.

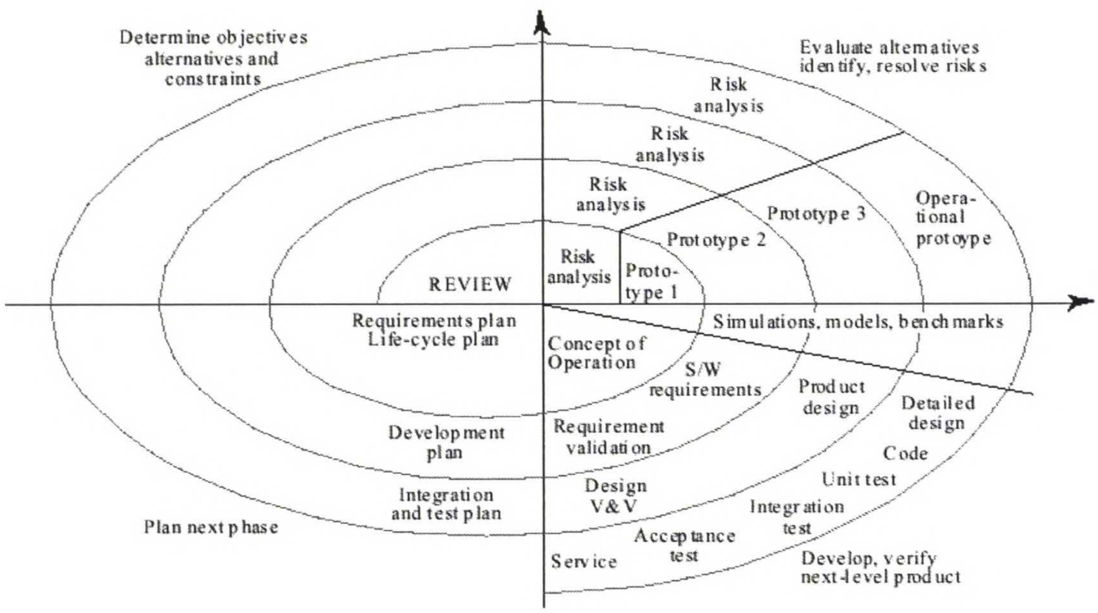


Figure 7. *Spiral Model of the Software Process /3/*

3.5 Customer Documentation within Software Development

Documentation development life cycle within planning-driven software development is derived from the traditional waterfall model of software development. There is no need to wait until the product is nearly complete before the documentation planning and development are started. The publication development process proceeds simultaneously with the product development process. The later the publication's life cycle begins in relationship to the product life cycle, the more difficult it will be to plan and produce high-quality documents. The activities occurring early in the development process emphasise planning rather than writing. The usefulness of the parallel cycles and early involvement gives better understanding of the organisation's hopes and plans for the new product. If very detailed contents specifications can be produced while the product development section is specifying the design, blanks can be started to fill in as soon as pieces of product begin to take shape. As a result, it is more likely to have subsections of the publications ready for early prototype testing, alpha and beta releases than if all the product details should be final before the publication's planning begins /32/.

Documentation development life cycle is claimed also usable within an evolutionary software model, such as a rapid-prototyping model. Technical communicators have traditionally been interested in producing more usable documents and are often very sensitive to the needs of their audience. Consequently, the opportunity for early testing of prototype publications is welcomed. Unlike the waterfall model, no closure is reached on the content specification until early design ideas are tried out on the user community. The iterations continue until a design that fulfils usability goals is reached /32/.

3.6 Process Management and Improvement Processes

Process management is a continuous process that monitors the status of the software process, plans and implements improvements in it, and maintains and develops the infrastructure for its management. Process definition is a central supporting and communication mechanism for the process management process /27/. Figure 8 illustrates the process management process. It consists of a set of concurrent processes, such as process analysis, performance measures or roll-out process. However, the process definition is a central supporting and communication mechanism.

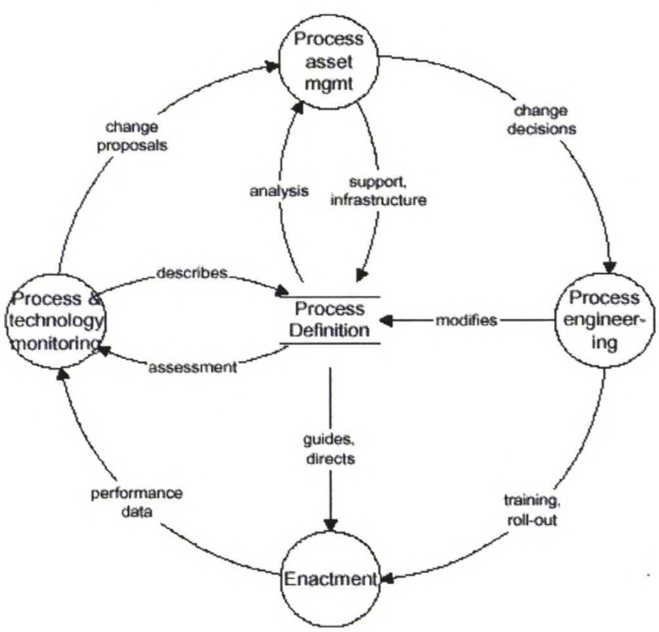


Figure 8. *Process Management Process /27/*

The need for improving a process can have a variety of reasons. The need may arise when external or internal customers make a complaint about the product quality, time schedule, flexibility or such. Long cycle times, poor productivity or high costs are very noticeable deficiencies. Bad performance can be detected also through benchmarking. There might be a demand for better consistency and predictability. Also new, more effective technologies, new business prospects, distribution of decision making or management preferences may be an impulse for an improvement process /27, 64/. A practical approach to process improvement process is illustrated in Figure 9. Feedback from enactment is used to the change the process definition. Process class definition acts as a template for a project plan. The project plan, accompanied by guides, acts as a set of instructions for daily work.

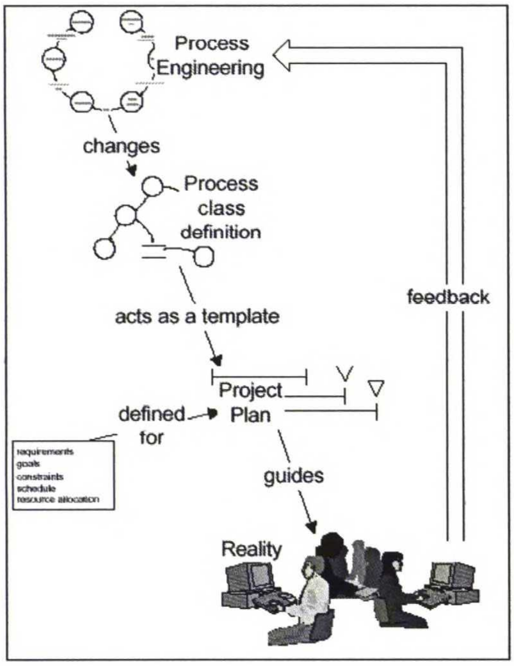


Figure 9. *Process Improvement Process in Practice /27/*

Venna /64/ introduces a comprehensive BPR-model which consists of the following six steps:

1. Find out the position of the company, its objectives and create a favourable atmosphere for change
2. Recognise the core processes of the company and choose the re-engineered processes
3. Analyse the current processes
4. Plan a new process
5. Implement the change
6. Establish grounds for continuous improvement.

For the needs of this study, only steps 3-6 are relevant since the scope is on structured documentation processes. In addition to this, the first two steps have already been taken and they are, in most cases, taken care of by the business management of a company /64/.

Processes can be illustrated for analysis in many ways but one choice is a process map in which the functions of the company are drawn in columns and processes as arrows penetrating them horizontally. A business activity map describes the completed actions, workflow and the connections between functions. A relational diagram models how the work is done as a flow from an event to another. The goal is, whatever method is used, to learn to understand the current process and why so happens /64/.

Planning a new process can be divided into four steps although the process is usually iterative /64/:

1. Define the performance requirements for the process

It is important to have precise process requirements in this phase. The goal should be ambitious and customer-oriented.

2. Consider alternative solutions for reaching the goals

New solutions should be striven without hesitation and limits to the imagination. Nothing should be taken for granted.

3. Develop a long term view on the process

It should be considered carefully how the process is likely to be like in the future.

4. Evaluate the alternatives

The effects and the costs of the new process should be calculated so that there will not be any negative surprises ahead.

The implementation of processes and information systems needs a pilot version to ever be successful. If problems occur, they can be solved without any extensive drawbacks. The pilot can even be stopped if the is an urgent need for resources in the current functions of the company. Often the most cumbersome problem seems to be the change resistance among the staff. The solution is to get the staff committed to the new processes and technology and let them be involved in the development process /64/.

Business process re-engineering and continuous improvement require a management system for process definitions and guidelines to keep the workings updated and maintain communication to the staff and the developers of processes /27, 64/

For major process improvements in organisations there are reference frameworks and software process standards such as CMM (Capability Maturity Model), SPICE (Software Process Improvement and Capability Determination), ISO 9001 / 9000-3 and Quality awards. Further options are to make a comparison between organizations, benchmarking or to learn from experience, better known as Quality Improvement Paradigm (QIP) /56/.

3.7 Measuring Software Processes

As the reference /8/ states: "Measurements are the basis for detecting deviations from acceptable performance." They are also the basis for identifying opportunities for process improvement. To effectively manage and control software development projects, a measurement process should be incorporated into their decision making and reporting process. Measurement costs money, but it can also save money through early problem detection and objective clarification of critical software development issues /51/.

The measurement process becomes a complete Plan-Do-Check-Act (PDCA) process that is integrated into the development process and the project manager's decision making process. The PDCA cycle that a project can use is /51/:

- Plan: Identify the issues or questions the project and manager has and then determine the data and measures to be collected to address them.
- Do: Collect data and, based on the issues identified and using baseline data, derive graphical representations to better illustrate the data trends.
- Check: Analyze the trends, graphs, data, etc. to better understand the issues and the performance towards their resolutions.
- Act: Report the results, recommend improvements, and identify new issues and questions.

Reference /8/ suggests similar phases as the four key responsibilities of process management. The names of those phases are define (P), measure (D), control (C) and improve (A). The letters in parentheses signify respective phases of the PDCA cycle as the content of the phases of these two processes are identical.

The question of what data to collect and how to define that data is driven by the current issues, projected issues, and characteristics of the software development process and products. The list of issues for measurement can be identified with a list of project specific issues and concerns, by setting priorities for the issues, and determining, which of the issues can be addressed with the software measurement process /51/. Goal-Question-Metric (GQM) framework provides a tool for organizations to decide which measurements to collect. GQM links process goals with the critical questions that must be answered to achieve the goals, and identifies data items needed to collect measurements /22/.

The process measurement has five perspectives that have to be considered in the check phase. The first concern is performance: facts about what the process is producing at the moment. The second concern is stability: is the process acting predictably. The third concern is compliance: the processes have to be sufficiently supported and faithfully executed. In addition, it has to be examined if the organization is fit to execute the process at all. The fourth concern, capability, states that the process has to be capable of delivering the products that meet requirements. The fifth and last concern, improvement, states what could be done to improve the performance or reduce variability or even what would move the mean of production to a more profitable level /8/.

Change in the process is inevitable as the development proceeds and requirements become better defined: cost, performance, and schedule estimates are refined; personnel, machine, test, and support resource requirements are identified; and insights from the measures themselves are gained. As the issues change, the new information should be used to adapt the measurement process to address the changes /51/.

Once the data collection has started, measurement definitions should not be changed. Changing a definition after the data collection has started will produce variations in the trends that could confuse the analyses and camouflage performance or related problems. When determining what data to collect, the preferred data is that which is a direct result of the developer's process. To assist in identifying the issues and defining data, the Software Engineering Institute (SEI) has developed frameworks for common software measures. The frameworks are based on a series of checklists that a project uses simultaneously for issue identification and measurement definition /51/. The topics of the available checklists are shown in Table 1.

Table 1. *SEI Core Measures /51/*

Unit of Measure	Characteristics Addressed
Physical source lines of code Logical source lines of code	Size, reuse, rework
Staff hours	Effort, cost, resource allocations
Calendar dates for process milestones Calendar dates for deliverables	Schedule, progress
Problems and defects	Quality, improvement trends, rework, readiness for delivery

3.8 The Technical Publishing Process

Every publishing system supports most or all of the processes illustrated in Figure 10. The collection phase involves, for example, keying, scanning and encoding information. The information usually needs some pre-processing before it is ready to be used for the create process. The material needs to be stored, retrieved or distributed after authoring. Also by-products might need the same material or some processing into different forms such as tables or indices are maybe desired. Finally the product is made in print, on-line or CD-ROM format /61/.

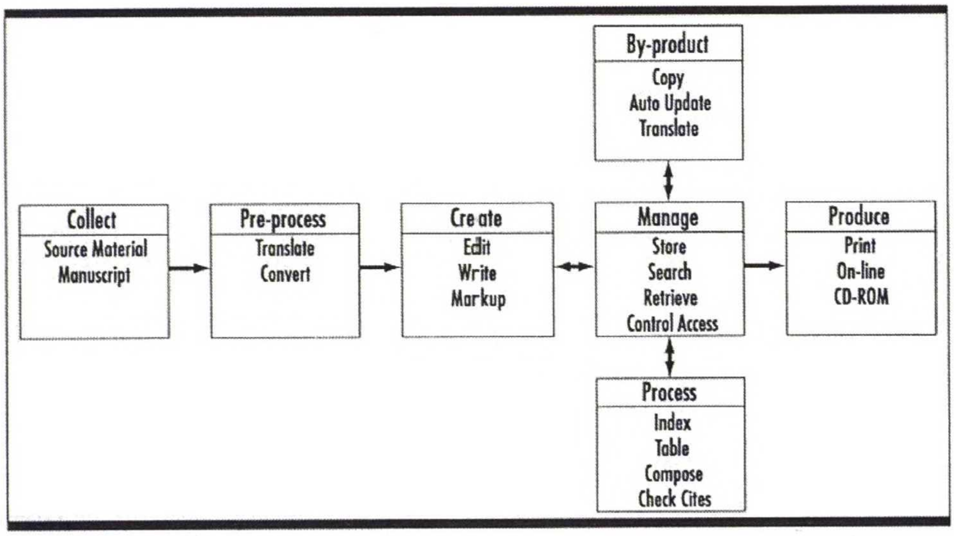


Figure 10. *The Publishing Process /61/*

3.9 Hypermedia Processes

3.9.1 Comparison of Hypermedia and Technical Documentation

The most common hypermedia format is HTML. Since HTML syntax is defined with the SGML metalanguage, they are very much related. HTML DTDs are just a couple of those many ready-made industry DTDs that can be produced with SGML. The main characteristics of HTML are links between information units, electronic publishing format in WWW and markup that is concerned with the layout as well as the logical structure of the document /28/.

Electronic publishing and browsing is getting more and more popular also in technical communication due to the limitations of paper format. Modern electronic documentation is an interactive application with searching, indexing and browsing capabilities. Many kinds of new media types, such as video, animation or sound, can be attached to it /15, 49/. In fact, the publishing format often is, and increasingly will be, HTML or XML-based XHTML (Extensible Hypertext Markup Language) which are both hypermedia languages. The authoring and processing languages for electronic documents are most often SGML or XML /28/. The used technologies are dependent on the current tools available /15/.

When moving to the web environment, there may be a need to break down an enormous pile of information into its components to get more manageable and maintainable pieces and to ensure future reusability. This concept, also referred to as modularity, has been adopted loosely from object-oriented programming. Technical communicators may go through a phase transition, as the system becomes much more complex, emerging and interactive /47, 49, 65/.

3.9.2 Hypermedia Engineering

Hypermedia engineering is defined by Lowe and Hall /33/ as "The employment of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of hypermedia applications." In other words, the term is about the employment of an engineering approach, which is managed, scientifically based and yet practical and pragmatic. The term engineering also refers to the need for appropriate process, which is critical to the effective development of large-scale hypermedia applications. At present, many aspects of the development process are either overlooked entirely or performed in a very ad hoc manner. Development methodologies tend to focus only on the design and implementation of applications, neglecting the early stages of the models and stages such as evaluation, risk analysis, determination of objectives and alternatives and planning /33, 34/. Figure 11 illustrates the current situation of the hypermedia development practice compared to the well-established waterfall software development process model.

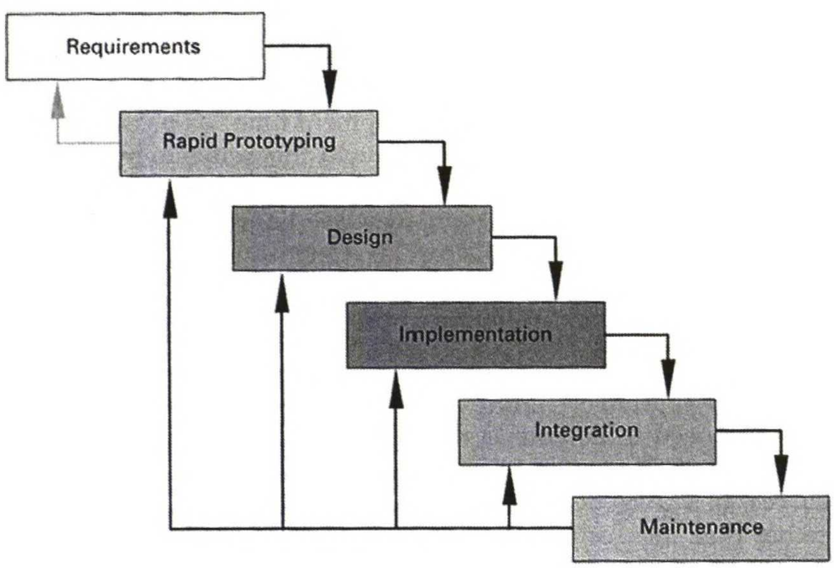


Figure 11. *The shaded areas show the focus of current hypermedia development practice mapped onto an existing software development process /33/*

The approach taken to developing hypermedia applications is extremely diverse. Large-scale SGML-based development draws significantly from database development practices where as CD-ROM development often draws from desktop publishing and cinematic development. Web development is typically very ad hoc and has little consistency in the approaches adopted /34/.

Many of the hypermedia engineering problems are similar to those that software engineering has had during the past thirty years while it has become a significant sub-discipline of computing science and applications engineering. Thus, hypermedia engineering can learn much from other disciplines, including software engineering /33, 45, 65/.

Still, there are also specific and development approaches, methodologies and tools only applicable in hypermedia engineering. A fundamental difference are the aesthetic and cognitive aspects of hypermedia design. Also the finer-grained maintenance affects on the maintenance management within the development process /33, 35/.

3.9.3 Hypermedia Processes and Process Models

The goal of hypermedia development has two major components, product and process. The first part deals with the creation of a hypermedia application and the final product. Its goal is to develop high quality hypermedia applications, which have the optimal balance of desired characteristics. The second part of the goal deals with the act of development itself. The goal of the process is to carry out the development in the most cost-effective, repeatable and manageable manner possible consistent with achieving the desired application /33/.

Hypermedia engineering is about managing all aspects of hypermedia development lifecycle. This involves using a process which is best suited to creating the desired application, but also doing so in a way that meets the goal of hypermedia development /33/. The types of activities that might possibly be included in a hypermedia process model are given in Table 2.

Table 2. *Possible Activities Included in a Hypermedia Process Model /33/*

• Project planning	• Feasibility analysis	• Architectural design
• Functional analysis	• Non-functional analysis	• Navigational design
• Information analysis	• Navigational analysis	• User-interface design
• Information capture	• Content design	• Testing
• Navigation verification	• Process measurement	• Project management

As hypermedia design has so far been very much based on ad hoc method, formal frameworks and design models are being researched. Hypermedia frameworks focus on scalable information representations. An example of a hypermedia framework is Matilda. One representative example of formal design representations is called Hypermedia Design Model (HDM). HDM incorporates a set of design elements which are used to define an application. The application is composed of typed entities, which are in turn composed of components. Entities and components are connected by structural links or application links. An HDM schema is a set of entity and link type definitions, and a schema instance is a set of actual entities and links. HDM provides a framework for hypermedia development, but does not address the development process itself /33, 34/. Also an improved HDM2 has been introduced by Garzotto, Paolini and Schwabe in 1993 /10/.

An open hypermedia system, which has simplified the many aspects of hypermedia development with a flexible management of content and structure, is called Microcosm. Microcosm reduces the hypermedia authoring effort with the concept of generic links, which can be manipulated with numerous architecture-specific filters. A similar type of concept is Hyper-G and its evolution, Hyperwave. Hyper-G is a client-server, Internet-based hypermedia system which provides session support, access control and a rich data model. AHM (Amsterdam Hypermedia Model) is a hypermedia design model combining the Dexter model of hypertext and the CMIF (CWI Multimedia Interchange Format) model of multimedia. AHM supports composition of multiple media into flexible presentations with synchronisation between the media /33/.

Software development processes have been defined with process models, abstract representations of the structure of the process. Equivalent models for the entire development lifecycle of hypermedia applications are only now beginning to be developed. One approach to design process models is called the Object-Oriented Hypermedia Design Method (OOHDM). In OOHDM, applications are developed using four incremental stages of development. The first stage is domain analysis, where a conceptual model of the application is constructed with OO modelling technique primitives and hypermedia specific add-ons. An adaptation of the UML (Unified Modeling Language) notation are used for modelling. Model diagrams emphasise the structure of the model and the relationships within. The other stages are navigation design, abstract interface modelling and, finally, implementation. A similar object-oriented approach is also EORM (Enhanced Object-Relationship Model). Besides, Relationship Management Methology (RMM) provides an approach to the design of hypermedia structures. RMM is based on entity relationship modelling which is an adoption from software development. It provides steps for modelling the entities in an application, the organisation of these entities, how this is used in supporting navigation, design of user-interfaces and run-time behaviour design /33, 34, 35, 44/.

Hypermedia Flexible Process Modelling (HFPM) described by Olsina /44/ is a model that adds more focus on design and construction issues not dealing with early lifecycle or maintenance tasks. With HFPM, a hypermedia project is broken down to tasks and phases. A phase is a grouping of strongly related tasks performed in certain order. A relevant set of observable characteristics of the artefact, process, and resource entities are chosen to improve and control the development process. To help in choosing the entities, the software engineering-based GQM (Goal-Question-Metric) approach can be used. GQM starts by analysing project goals and refining them into a set of questions that can be answered quantitatively. The questions then specify the measures that should be collected in order to track progress toward the goals. HFPM is claimed to cover all the essential phases and activities of a hypermedia project, produce greater visibility that contributes to project planning and scheduling and helps to establish milestones and metrics, consider a balanced use of logical and physical modelling and promote human understanding and communication /45/.

A number of factors influence the adaptation of a hypermedia process. Development time frames are critical in the field that has the rapid rate of changing technologies and, in some cases, commercial imperatives. The scale of application has a major effect. Small-scale application projects, which involve only a few people, do not even need a formal development process. If information is provided in multiple forms, these various forms impose requirements and constraints that have to be considered. The developer's expertise has a major impact on the selection of a development process as well as inherent risks and uncertainties. The most common process models used in hypermedia development are the waterfall model, incremental development process model, prototyping process model and the spiral development process /33/.

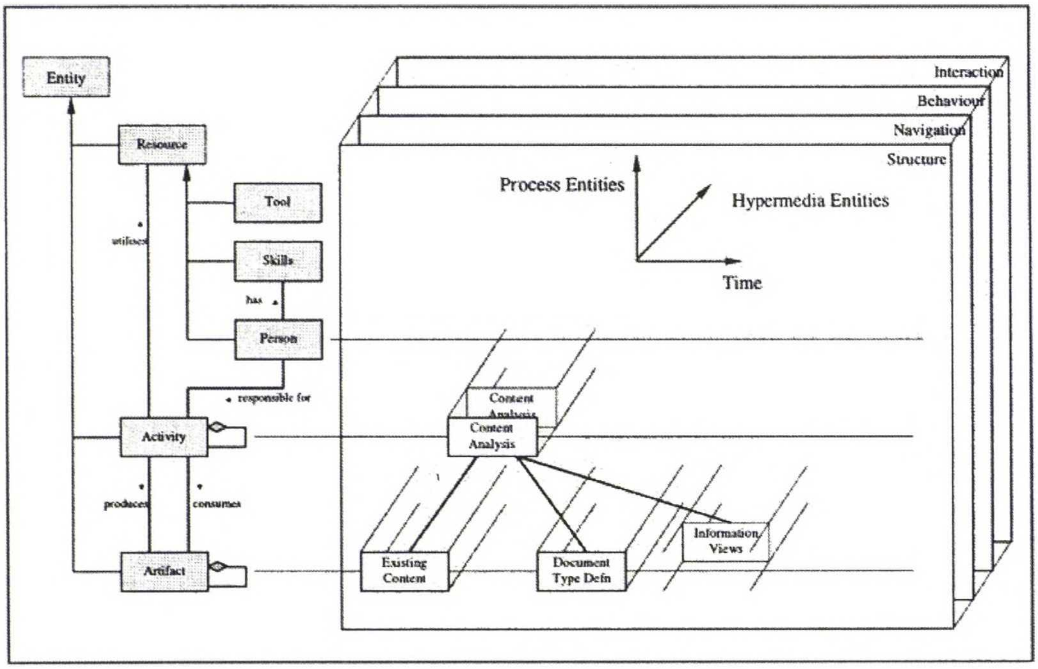


Figure 12. *Hypermedia Development Process Reference Model /34/*

Lowe and Webby /34/ applied the major hypermedia concepts and process modelling concepts from other disciplines to compose a reference model shown in Figure 12. The model contains three dimensions: a process entity dimension, a hypermedia dimension and a time dimension.

A series of process entities occur in this three-dimensional space that have defined interrelationships. The process entity dimension defines the classes of entities that can exist and the relationships that exist between entities of different classes. The entities can be divided into three classes: resource used in the development, activities that are carried out and input or output artefacts generated by the various activities. The second dimension of the model defines the hypermedia focus of the model entities as a set of layers. It includes hyperbase and access structure, navigation within the hypermedia application, behaviour over time and user interaction /34/.

Lowe and Webby /34/ applied the reference model in a typical SGML-based documentation project. Figure 13 shows a high-level process obtained. The project consisted of a development productivity study and comparison of conversion methods from HTML markup material into SGML markup material.

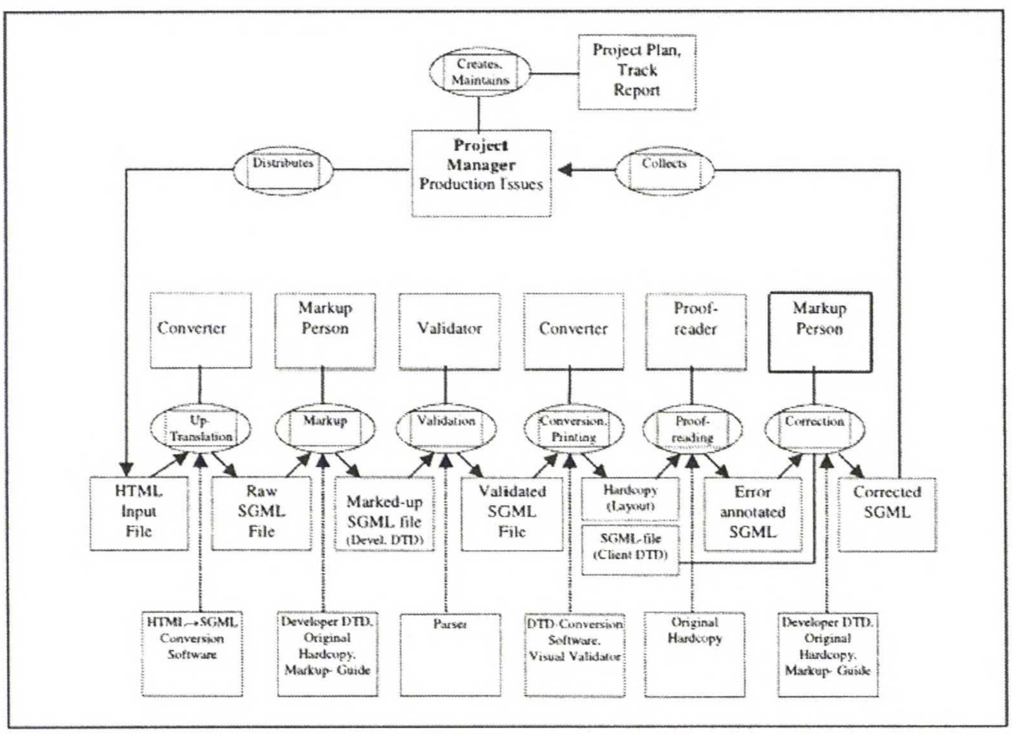


Figure 13. *Typical Process Model for an SGML Markup Project /34/*

Although the research was still ongoing, there were already some early results. It was found that prototyping is, indeed, a valuable mechanism for refining client requirements and obtaining an initial understanding of design issues. However, prototyping should be done in a much later phase than is used to because it potentially encourages the client to identify a much broader range of requirements, causing significant problems later in the development. Many requirements may be found unacceptable and impossible to implement when, for example the SGML DTD development is finally completed. Also group-oriented work was found to be useful so that a sense of ownership, causing that resistance in quality evaluation, can be avoided. This concerns creative work such as designing graphics or other specific content /34/.

Hypermedia development processes and process modelling will continue to evolve and improve in the future. More hypermedia specific issues have to be paid attention to in the development phase. Also product evaluation techniques need improvements. No suitable metrics or heuristics to assist in interpretation of metrics exist /33/.

3.9.4 Hypermedia Process Measures and Project Management

Mendes, Hall and Harrison /39/ proposed and validated their hypermedia metrics using a quantitative survey and a quantitative case study. The used variables were hypermedia document size, connectivity, compactness, stratum, link generality and link representation. They were defined as follows:

- Hyperdocument size refers to the number of HTML-documents or other kind of documents that the hypermedia application has.
- Connectivity refers to the number of links that the hypermedia application has. The links considered here are either structural or referential.
- Compactness indicates how inter-connected the documents are in a hypermedia application, for example the level of cross-referencing.
- Stratum indicates to what degree the hypermedia application is organised for direct reading, for example the linearity of a hypermedia application.
- Link generality refers to whether the link applies at a single anchor only or at multiple anchors.

The analysis /39/ showed that hyperdocument size, connectivity and link generality were all confirmed as possible metrics by the test groups. Compactness was suggested as a metric for web systems but it was not shown to illustrate the development effort for systems that store the links in linkbases, as Webcosm /39/ used in the case study. The result obtained by the study was that the more a hypermedia application has documents, links and coupling among documents, the more time authoring takes. The time spent on authoring may be reduced by organising it into different phases so that different authors can tackle different parts of the project. The time used in linking may be reduced by adding useful information about types and context within links so that they do not have to be updated every time they are referenced. The same advice applies to reducing the time spent on coupling among documents.

There is little knowledge of how to scope and plan projects in the hypermedia domain, including creation of web sites, CD-ROM titles and SGML documentation. As a result, most project planning and subsequent development processes are currently ad hoc and ill defined. In contrast, software projects are better understood and those techniques should be taken advantage of when developing such techniques for hypermedia project planners. According to Lowe and Hall /33/, the important characteristics for early estimation and resource scoping need to be identified in the future.

3.10 Technical Documentation Process

3.10.1 Steps of Implementing the SGML Documentation Process

The following steps are to be taken to implement the SGML process /7/:

1. Document type definitions and lay-outs are designed and implemented
2. The work flow of the documentation process is defined; such as converting, authoring, maintaining and delivering documents
3. Software components are chosen and integrated into a whole.

The main components of the technical documentation process, authoring, maintaining and delivering documents, are explained in the following three subchapters. The SGML conversion process is, however, out of this study's scope.

3.10.2 Authoring SGML documents

Authoring is the process of preparing a document set, which a PDF book including the whole document library /65/. For authoring SGML documents, a document type definition and proper tools for using the DTD are needed. Also a document conversion process from the legacy Interleaf documents also has to be implemented /7/.

The traditional word and text processing systems usually have coding information within the content. The meaning of this information is to separate logical elements of the text from each other and define typographical layouts for them. The production of layouts is time-consuming and has many aspects to be taken care of. Usually these codings are finalised in the postprocessing phase by a specially trained person called a maker-up /7, 17/.

Text documents can be coded in three different phases. At first the structure of a document and included attributes are analysed and all the elements are specified and named, such as paragraphs, headings, margin texts, header or footers. The second step is to define the processing rules with control marks for all the element types. The processing rules are used later in a document's life cycle to process the contents for layout and archiving purposes. The third, final step is to add these defined control marks to the context /7/. The ease of authoring in both XML and SGML is dependent on how well the document structure meets the author's needs and on how advanced the authoring application is /63/.

If the authoring of integrated data involves more than one author, there needs to be some workflow techniques in use. This is called collaborative authoring. The term also includes specifications that are concerned with file manipulation control, including file naming conventions, file versioning and file variants /4/. The tools used for authoring in a development process should support information exchange and collaboration between software designers and technical communicators /6/.

Reference /53/ suggests the following authoring requirements for a publishing process:

1. Content authoring should only take place once.

The benefits are that the needed resources are not multiplied and fewer errors are introduced. Also no platform-dependent formatting should be used so that flexible re-use of content is fulfilled.

2. Content authoring should be facilitated.

The efficiency of authoring improves, in terms of time and proactive support for correct encoding of data, with sophisticated tools. Good tools usually hide excessive details of the markup from the user.

3. Metadata authoring should be facilitated.

Authoring metadata, or underlying definition of the contents, may either be a part of the authoring process or it may be a separate work item. The former approach benefits from the information that is present at the moment of authoring. For example, metadata authoring can be done by a metadata-authoring tool automatically during the computer system login session of the author. The latter approach has the benefit being an author-independent task that can be regarded as a more objective approach. However, the third party does not have an access to the variables of the authoring environment so that the description may not be as rich and automated.

3.10.3 Document Layout and Publishing

The most relevant benefit of SGML is that documents are independent of formatting and processing conventions. The same information can be presented and used in different ways in SGML-based environment. To be able to apply layout standards, FOSI or DSSSL in the case of SGML as presented in Chapter 2.2 Benefits of Structured Documentation, the user must have experience in typographic design, basic knowledge of SGML and understanding of the requirements of the structure and formatting of documentation. The structure of a document must be defined so well that the desired layout can be applied based on it /7/.

As technical documentation is moving from paper-oriented publishing into on-line publishing, the ways of making good use of documentation increase /49/. The same document content can be used in different presentations by applying media-specific style sheets, such as W3C's recommendations CSS (Cascading Style Sheets) or XSL (Extensible Stylesheet Language), into a structured content either in the publishing end or in the client end. The content may also be split into smaller document units in order to be readable e.g. in mobile platforms. In a network environment, the delivery mechanism may be either a push or pull type of delivery. The operations model of the Web is that connections are initiated from the client to the server and the connections may not be continuous. Thus, the pull type of delivery is currently the most common delivery mechanism /53/.

Reference /53/ suggests the following delivery requirements for a publishing process:

1. The user may be provided with asynchronous services.

In addition to the pull metaphor based information retrieval, the user can be provided with a service with push-type channels to deliver the content. Typically, the receiving medium can buffer the content until the user wants to consume the received content.

2. A diversity of access media needs to be supported.

A publisher needs to support menu access media in order to reach as many end-users as possible. This applies to, for example, slow network connections and the quality of terminal devices.

3. The delivery needs to take place with an increased speed.

A requirement that addresses the content object encoding formats, transfer protocols and network types. The speed becomes a critical factor because the typical size of a transferred object is increasing due to multimedia objects.

3.10.4 Project Management

The long-term success of technical documentation products within an organisation requires understanding the overall processes involved. The project manager has constraints that limit the effectiveness of the process /1/. Planning of documentation projects should be approached from the customer requirements perspective /14/.

Limited time and resources are always a problem. The technical documentation process must also fit within larger processes of operating a company. Formalized procedures and team participation direct the work in successful processes. An efficient control system for archiving and making backups can save time that is not usually added to project estimates. Overspecialization and uncoordinated development can limit the unity of the development /1/. In large departments, the specialization makes sense in order to increase efficiency. Each specialist contributes to a productivity gain in the organization as a whole /1, 14/. The mere act of applying project management to documentation may even slow down the management of the process. By working on overlapping projects and parallel projects, a documentation effort can accomplish several goals with one set of tools. The choice of tools for document creation and maintenance may be limited by the platform, which was chosen by the enterprise for other purposes than documentation. Technical documentation managers cannot ignore that their processes must produce deliverables that meet or exceed customer expectations /1/.

The task of managing technical documentation involves coordinating contradictory business objectives, dissimilar technologies and various personal priorities toward completion of projects. The processes must be repeatable, consistent and controllable to be effective /1/. A study /14/ conducted among large telecommunication companies presented that in one of the companies the reuse of document modules of previous documentation releases was able to reduce the total development cycle by 40 percent. However, the introduction of reuse means ideas about authorship and ownership need to change and employees need to collaborate constantly in order to succeed.

3.10.5 Storing SGML Documents

SGML documents can be stored on server like any other files in the system. In that case, a disciplined folder structure is recommended. In many cases the amount of information is so huge that it cannot be managed, stored, searched or retrieved without putting it into a database /7/. Of important strategic value for businesses is, beside producing and delivering technical documents, also the improvement of mechanical efficiency and cost effectiveness of a wide variety of organisational applications, processes and information flows brought by SGML and management systems /5/. In this context, information means data that has a meaning in some context for its receiver.

Database models can be divided into three categories: relational, object-oriented or hybrid databases. Although relational databases are nowadays the most common manner of managing databases, object-oriented databases are considered the most suitable solution for document information. They enable even complicated data modelling features compared to relational models. Also, the object-oriented way of thinking is similar to SGML structures, such as attributes. However, object-oriented databases still lack a proper query language, optimisation of queries and the control of parallelism /7/.

For the time being, at least three applicable implementation solutions exist for storing and using document-based information: storing the document information in relational databases, storing entire documents in relational databases and applying documents themselves as textual databases, and utilising the structure of documents by using it as a database model and linking the hierarchy tree's nodes to structured documents /7/.

3.10.6 Technical Documentation Workflow

Workflow management is an essential part of implementing an SGML documentation system. In this context, workflow can be seen as a flow of tasks liked to each other with connectors /61/. In the SGML environment workflow has been defined as "a process of getting data into and out of an SGML-based data management system and doing something with it".

Workflow management is an essential part of SGML implementation and includes tasks, decisions and their relations. Workflow is usually presented graphically, which facilitates the implementation planning /26/. When a workflow solution has been implemented, there are many commercial workflow managers available that provide a graphical tool for the creation of projects. Creating a project with such a tool involves drawing a flow consisting of tasks, decisions and connectors /61/.

3.10.7 Technical Documentation Process Measures

Laru /32/ has considered two general performance measure frameworks, derived from software engineering environment, as appropriate for customer documentation process. They are goal-question-metric, described in Chapter 3.9.3 Hypermedia Processes and Process Models, and performance pyramid, illustrated in Figure 14. At the base of the pyramid the individual performance measures are classified into quality, time or cost. The management of the work is done on the way to the top of the pyramid. Also the usefulness of functional measurement and measurement of specifications' accuracy in planning-driven development was observed.

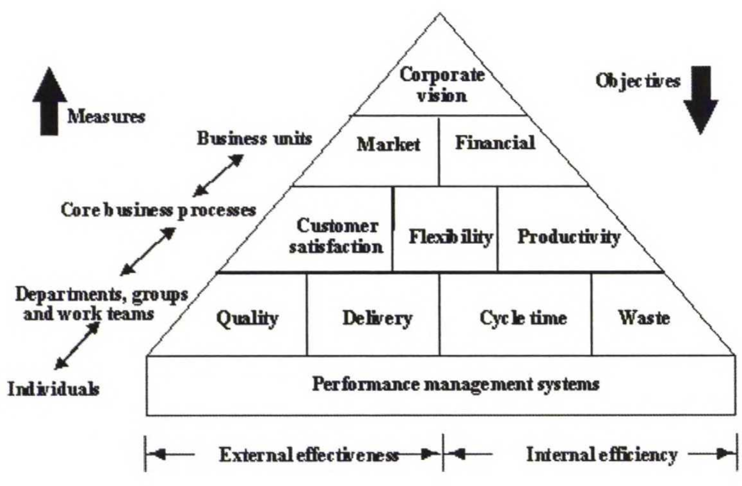


Figure 14. *Performance Pyramid /10/*

The performance measures determined and implemented for customer documentation within software development process were /32/:

1) Completeness of customer documentation,
2) Accuracy of design documentation schedule,
3) Throughput time of customer documents,
4) Changes in release documentation contents,
5) Extra work and
6) Relative resource usage.

It was also suggested that the following measures should be considered in another similar situation: 7) size of release customer document library, 8) percentage of customer documents tested and 9) number of fault reports /32/.

Based on her benchmark study, Hackos /14/ recommends constant time and resource monitoring in order to decrease the time spent on both valuable and invaluable activities of a documentation process. After the monitoring results have been analysed, the process improvement comes through better training or mentoring and better understanding of business issues.

3.10.8 Technical Documentation Process in Practice

Juntunen /23/ has presented an action plan to implement the training documentation process. However, its basic ideas can be applied to other documentation processes as well. The action plan consists of the following five steps:

1. Unify the processes

It is very strongly suggested to combine the documentation process with the product process and its implementation.

2. Make a technical solution

A technical solution for tools and storage should be made.

3. Introduce the process

The process should be first introduced to the team leaders and then to the teams. The presentation of the process should be well conceived.

4. Establish the documentation organisation

The documentation organisation should be defined.

5. Put the process into action

The process will be taken into action with the tools and storage solutions. This requires training for the writers and managers to use the system. The transition to the new process is where the real difficulty lies.

The following Figure 15 describes the customer documentation process in the SGML-based system after the SGML migration at the Fixed Switching of Nokia Telecommunications in 1998. The migration was similar to this case study so that it can be seen as a model example to some extent. The SGML-based documentation process did not change the actual customer documentation process very much. Still, the documentation tools were changed completely /26/.

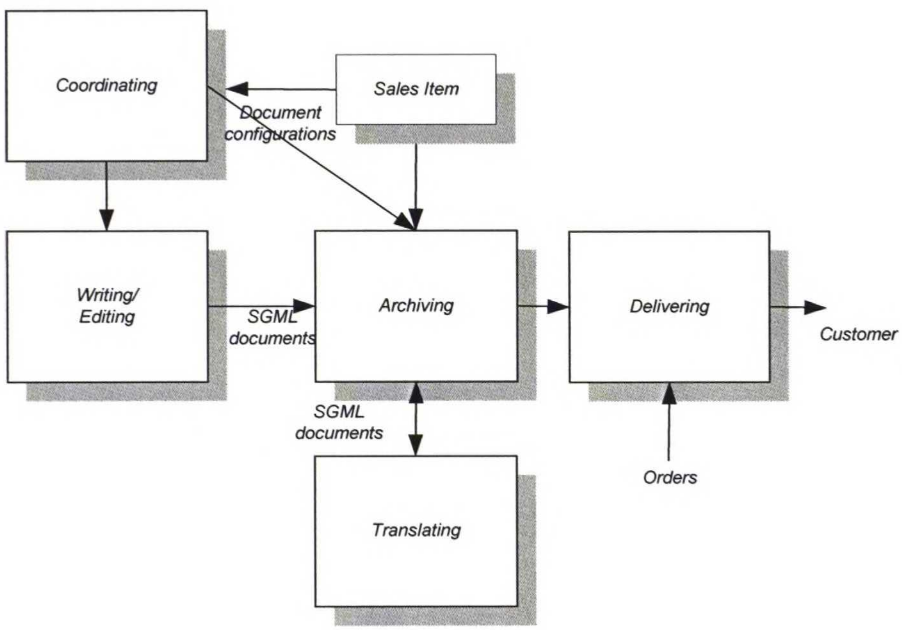


Figure 15. *Customer Documentation Process in an SGML-based Documentation System /26/*

The roles of editorial and production staff may change through SGML migration. Many of the production phase tasks are automated and some postprocessing tasks, such as proof generation, page inspection and data management, will most likely be done by editorial staff. In spite of automation, some ambiguous and unique material still needs manual handling /61/.

The SGML-documentation process is considered to be more automated than the traditional publishing process. Documents can be processed with algorithms into many different forms after the context has been authored /37/. SGML coding greatly enhances the ability to produce automatically tables of contents, indexes, cross-reference tables and such /61/.

Database practices bring consistency and repeatability to the SGML-based development /34/.

Some disadvantages of the SGML environment have also been discovered. Printing is problematic because of conversions from SGML to Postscript (PS) or Portable Document Format (PDF) and dependence of the specific DTD causes problems for delivering or transferring documents /26/.

SGML implementation also needs to take into consideration cultural issues and not focus only on technical development and financial analysis. Benefits need to be described sufficiently so that users can see how automating some processes allows them to focus on what they do best: creating information, not formatting and accessing it. As a real case, this was learned in the implementation project of Douglas Aircraft Company /61/.

4 USABILITY

4.1 Definition of Usability

According to ISO 9241 Ergonomic requirements for office work with visual display terminals standard /54/, usability is defined as "the effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments." The definition leads to design objectives of a system and provides the means for explicit measurements for usability.

Usability applies to all aspects of system with which a human might interact. Almost all computer features have at least some user interface components. Even a facility to transfer between two computers will normally include an interface to trouble-shoot the link when something goes wrong /41/.

Usability is not a single, one-dimensional property of a user interface. Usability has multiple components and is traditionally associated with these five usability attributes /41/:

- Learnability: The system should be easy to learn so that the user can rapidly start getting some work done with the system.
- Efficiency: The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.
- Memorability: The system should be easy to remember so that the casual user is able to return to the system after a period of not having used it without having to learn everything all over again.
- Errors: The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur.
- Satisfaction: The system should be pleasant to use, so that users are subjectively satisfied using it.

The main questions of usability are what are the good and bad areas of the design and how to fix the bad parts and strengthen the good parts. It is not needed to know how bad the design really is but to know that there is a usability problem and why users use it in erraneous ways /42/.

The requirements can be different for different user groups. Ease of learning is the most important factor for novice users, ease of remembering is the most important for casual users and efficiency and productivity of use is the most important for experienced users /52/.

Usability is a key concept within a broader term Human-Computer Interaction (HCI). The term HCI was adopted in the mid-1980s for describing a new field of study. The focus of interest was broader than just the design of the interface and was concerned with all the aspects that are related to the interaction between users and computers. The importance of HCI has been shown by evidence of increased productivity and improved safety /46/.

Usability testing is often needed because designers do not necessarily construct transparent and easy to use systems that require no support documentation. However, usability testing is often considered as lengthening development cycles and leading to an unwanted financial investment. The costs may be paid back later with better machine utilization, flexibility, and product quality. There will likely never be a comprehensive list of guidelines available for designing usable systems. Guidelines are often too rigid or too generic to be applied easily to the situation at hand /38/.

4.2 A Systematic Approach to Usability

The recommended steps on a systematic approach to usability improvement are /41/:

1. Recognise the need for usability in the organisation.
2. Usability must have management support, which often means that designers must be able to change their initial design ideas according to user needs.
3. Specific resources need to be dedicated for usability.
4. Systematic usability engineering activities must be integrated into the various stages of the development lifecycle.
5. All user interfaces must be subjected to user testing.

However, if this five-step plan seems to be too much, there is a simple plan to start from: one of the existing user interfaces should be picked and subjected to a simple user test by defining some typical test tasks. Five representative, potential customers or users should be chosen to perform tasks with the system. The usability problems likely to be found should be solved in the next release by using iterative design /41, 42/.

The other goal of this study is to find out how efficient the PMR customer documentation process really is. In addition to process measures, one manner to find out concrete needs for improvement is to conduct a usability study.

4.3 Evaluation Methods

Evaluation is concerned with gathering data about the usability of a design or product by a specified environment of work context. Four reasons for doing evaluations are identified: /46/

- Understanding the real world, for example when testing if a prototype fulfils users' needs.
- Comparing which design is the best for the purpose.
- Engineering towards a target or some form of metric, such as better usability of the product than competitors have.
- Checking conformance to a standard such as screen legibility.

The goal of evaluation is to find out how the system should be developed. Evaluations are meant for testing the functionality of a system and user satisfaction on the system. Also the overall opinions about the system are considered. There are two main classes of evaluation. Formative evaluation is for evaluating an unfinished product during the design process and the goal is to improve the product in the making. Summative evaluation means assessing the overall quality of an interface or a product after it has been developed /24, 46, 52/.

There are many different evaluation methods for different situations. The chosen method depends on available time, budget, staff, technology and other resources. Also the amount of users and their expertise have an effect /41/. The most simple methods are thinking aloud and user and task observation /52/. The list of systematic evaluation methods is presented in Table 3.

Table 3. *Evaluation Methods /52/*

• Heuristic evaluation	• Observation
• Thinking aloud	• Focus groups
• Questionnaires	• Logging actual use
• Interviews	• User feedback
• Performance measurement	

The methods fall in three categories: testing, inspection and inquiry /41, 50/. In testing /41/, representative users work on typical tasks using the system and the evaluators use the results to see how the user interface supports the users in their tasks. Examples of testing approach are thinking aloud and observation methods. In inspection /41/, usability specialists examine usability-related aspects of a user interface with in-depth knowledge of general usability rules and measurements. An example of the inspection approach is a cognitive walkthrough. In inquiry /41/, usability evaluators obtain information about users' opinions and understanding of the system by talking to them, observing them using the system in real work or letting them answer questions verbally or in written form. An example of the inquiring approach is observation.

The most relevant methods for this study were chosen from Table 3 and they are examined here in-depth. An introduction to all methods can be found from Ryökkynen's thesis /52/.

A think-aloud method involves having a test subject use the system while continuously thinking out loud. The goal is to identify the users' major misconceptions, in which the method is considered an effective one. The strength of the thinking-aloud method is the wealth of qualitative data it can collect from a fairly small number of users and with low costs. The users' comments often contain vivid and explicit quotes that can be used to make the test report more readable and memorable. The main disadvantage of the method is that it is not very well suited to most types of performance measurement such as time or task completion data. The method may feel unnatural and the more advanced users may find it complicated and delaying. Also the way and speed of learning may appear different from an authentic situation /38, 41, 52/. The technique can be used in any stage of product development /19/.

Observation is done by visiting at least three regular users and interfering as little as possible with their work. The observer takes notes unobtrusively so that the test situation would resemble the actual working situation. However, toward the end of the visit, it would be reasonable for the observer to help the users as a reward for participating in the study and also to learn about the things they wanted to do and why could not do them themselves /19, 41, 52/. The benefit of the method is that the users carry out actual working tasks. It does not remove the user and the product from the context of the workplace such as is the case in a laboratory environment. The deficiency of observation can be seen to be randomness, which would consist of selectivity of the observations, inaccuracy and neglects. The problem is also that the test situation is not controlled by the observer but the user himself. The technique is best suited for early stages of development, when more information about the issues surrounding the use of product rather than actual metrics are needed /19/.

The inquiry method can be accomplished by means of questionnaires. There are two types of possible question structure in questionnaires: closed questions and open questions. With closed questions, the respondent is asked to select an answer from a choice of alternative replies. Closed questions usually have some form of rating scale associated with them. The responses obtained on the different rating scales are converted into numerical values and statistical analysis is performed. With open questions, the respondent is free to provide his own answer /46/. The weakness of the rating scale questions is that the previous answers can affect the latter ones. Questions based on statements should be formed as different direction statements and they should be claimed in random order /52/. To ensure ease-of-use of the questionnaire, only a small number of different types of questions should be mixed in the same questionnaire. Also, the possible rating scales for replies should be the same throughout the questionnaire. A problem with questionnaires and interviews is that people tend to give answers they think they ought to give, especially to sensitive questions /41/.

The best test results are obtained if the test users correspond to the actual users of the system with their physical and mental features, education, career background and product knowledge as much as possible /52/.

4.4 Required Number of Test Users

Tom Landauer and Jacob Nielsen have showed in their research /40/ that the number of usability problems found in a usability test with i users is:

$$Usability\_Problems\_Found\ (i) = N(1-(1-\lambda)^{i}) \quad (1)$$

where, i is the number of test users

N is the total number of usability problems in the interface and

λ is the probability for finding any single problem with any single test user.

After several projects of research, the mean number of problems in the interface, N, was 41 and the probability for finding any problem with a single user, λ, was 31%. Plotting the curve for λ=31% gives the result illustrated in Figure 16. The figure shows that the first three users come up with most of the usability problems of the system. Adding more users gives less and less new data because the same observations repeat every time. After the fifth user, the most important problems, 85%, are already discovered. To find out all the usability problems at least 15 users are needed. The preferred method is to conduct three different tests with five users /40, 41, 43/.

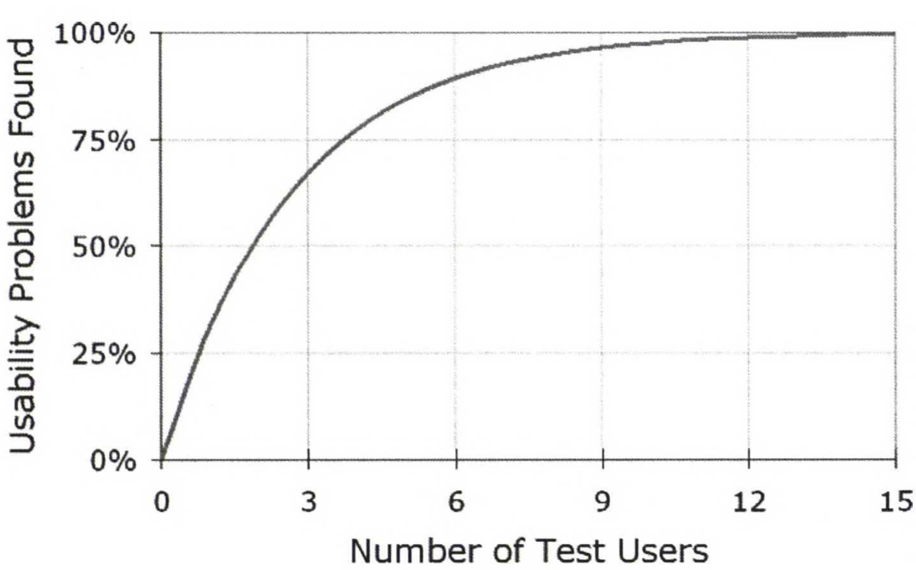


Figure 16. *Number of Test Users Needed for Finding Usability Problems /43/*

However, Formula 1 applies only for comparable users who are using the system in fairly similar ways. If there are highly distinct groups of users, the test needs to include people from all different groups /43/. Nielsen /43/ suggests that a test needs to include 3-4 users from each category if testing two groups of users and 3 users from each category if testing three or more groups of users.

4.5 Usability Metrics and Measurements

Often there is a need to run two sets of usability studies: one to collect numbers and another to get insight into the user's behaviour and thinking. The first study gives the numbers and the second study gives the information needed to redesign the system /42/.

The simplest usability metric is success rate when performing a representative task. It means recording the percentage of test users capable of accomplishing what they were asked to do. The other two main usability measures are task performance and subjective satisfaction. Task performance is usually measured by the amount of time it takes an average user to perform an average task. Subjective satisfaction is usually measured by the user's answers to a questionnaire with free form statements. It is also possible to measure the quality of the outcome of the user's task. For example, if the goal is to book the cheapest airline ticket to London, it can be measured if the user, in fact, booked the cheapest ticket or, if not, how it is compared to other ticket prices /42/.

Success rate measure can be collected while running a thinking-aloud study. Most other usability metrics only work if the users are let concentrate on using the system without interruptions. This concerns particularly any time measures when it is desired to know how fast users can perform.

Planning of a usability measurement starts out making the goal clear. The goal can be defined as e.g. usability or improved customer perceptions of the quality of a company's user interfaces. The goal is broken down into components like the usability attributes given in Chapter 4.1 Definition of Usability. Two of them are learnability and efficiency of use. The components need more quantification. For example, the component efficiency of use can be quantified as the average time it takes users to perform a certain number of specified tasks. Other quantifiable usability measurements may also include the ratio between successful interactions and errors, the number of tasks of various kinds that can be completed within a given time limit, time spent learning the system, subjective satisfaction of a user and comments of users. The next thing is to define a method for measuring the performance. Two obvious alternatives are bringing test users to the lab to perform a list of tasks or observing a group of users at work in their own environment. Finally, it needs to be defined how the collection of the data from the study is carried out. Following the previous example it can be done by a stopwatch or the computer measure. Nevertheless, it is important to have a clear definition of when a task starts and when it stops /41, 52/.

4.6 Stages of a Usability Test

Kanerva and Rojek /25/ have defined a usability-test process that includes planning and designing the test, collecting and analysing the data, and communicating the results of the test. Every step of the process depends on the step prior to it. Thus, the good foundation brings better chance of collecting good data and finally useful information about the system.

A usability test, or the collection phase of the usability process, has typically four stages /41/:

1. Preparation

Preparing the test environment, computer systems, material and instructions ready as specified in the test plan. Everything should be ready on the arrival of the user.

2. Introduction

The experimenter gives the test user a brief explanation of the purpose of the test and introduces the test procedure. Especially for an inexperienced experimenter, it is good to have a checklist of important points to cover during the introduction. After the introduction, the user is given instructions and asked to read them. If any questions emerge, they should be handled before running the test.

3. The test itself

During the test, the experimenter should normally refrain from interacting with the user and should definitely not express any personal opinions or indicate whether the user is performing well or poorly.

4. Debriefing

After the test, the user is debriefed and asked to fill in any subjective satisfaction questionnaires. The questionnaires should be administered before any other discussion of the system. The experimenter should write a brief report on the experiment as soon as possible while the events are still fresh in the experimenter's mind and the notes still make sense.

After collecting the data, the test results are collected and analysed. The material includes observation sheets, questionnaires, interview notes, logs and recordings. The analysis report describing all the found usability problems is written based by all the collected information. The report should discuss the results of the tests and actions that should be taken based on these results. Detailed report with a list of possible solutions helps stimulate discussion and ideas among the evaluation team /20/.

4.7 User Interface Design

The basis of user interface design is to keep the focus of the design on users and their processes. The designed system should be useful, efficient and easy to use at the same time. Usability is an important criterion for the development of an application for internal customers of a company, although it is often neglected. Usability design is often overlooked because of shortage of resources. The available low resources are mostly used to develop new tools without thinking either the environment as a whole or the processes of the users. Small investments into usability design pay themselves back quite soon, if the application has a large number of users inside the company /30/.

User interface design should be based on straightforward solutions. Function chains should be simplified into direct contact solutions so that the user has as little intermediate phases as possible when aiming at a goal. Complicated function chains are a consequence of adding new features to software without thinking enough about the goals of a regular user /31/. The difference is illustrated in Figure 17.

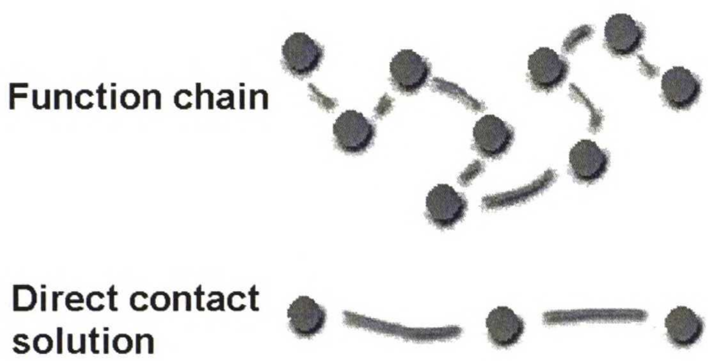


Figure 17. *Function Chain vs. Direct Contact Solution /31/*

The impression given by the learning curves is often that a system can either be easy to learn or efficient but initially hard to learn. One solution to merge the best parts of both learning curves is to provide a user interface with multiple interaction styles. The user can start with the easy-to-learn interface and later change to another that is more efficient to use. The typical way is also to include accelerators in the user interface. Accelerators are user interface elements that allow the user to perform frequent tasks quickly even though the more general way is still available. Typical examples of accelerators are function keys, command name abbreviations and the use of double clicking. Other solutions include the use of descriptive field labels and appropriate choice of default values /41/.

4.8 Usability of Authoring Tools

Authoring Tool Accessibility Guidelines Working Group (AUWG) of W3C has made a specification that provides guidelines for Web authoring tool developers. Its purpose is two-fold: to assist developers in designing authoring tools that produce accessible Web content and to assist developers in creating an accessible authoring interface regardless of disability. These tools can be either WYSIWYG HTML and XML editors, tools that offer the option of saving material in a Web format or transform documents into Web formats, tools producing Web oriented multimedia, tools for site management or site publication, or tools for management of layout /62/.

Each of the guidelines include a list of techniques that may be suggested strategies, references to other accessibility resources or examples of how deployed tools satisfy the checkpoint. The proposed seven guidelines, edited from the perspective of user-interface design and usability needs, are the following /62/:

1. Support accessible authoring practices.

If the tool automatically generates markup, many authors will be unaware of the accessibility status of the final content unless they expend extra effort to review it and make appropriate corrections by hand. Since many authors are unfamiliar with accessibility, authoring tools are responsible for automatically generating accessible markup, and where appropriate, for guiding the author in producing accessible content.

2. Generate standard markup.

Conformance with standards promotes interoperability and accessibility by making it easier to create specialized user agents that address the needs of users with disabilities. In particular, many assisting technologies used with browsers and multimedia players are only able to provide access to Web documents that use valid markup. Therefore, valid markup is an essential aspect of authoring tool accessibility.

3. Support the creation of accessible content.

Well-structured information and equivalent alternative information are cornerstones of accessible design, allowing information to be presented in a way most appropriate for the needs of the user without constraining the creativity of the author. Yet producing equivalent information, such as text alternatives for images and auditory descriptions of video, can be one of the most challenging aspects of Web design, and authoring tool developers should attempt to facilitate and automate the mechanics of this process. For example, prompting authors to include equivalent alternative information such as text equivalents, captions, and auditory descriptions at appropriate times can greatly ease the burden for authors. At the same time, the tool can reinforce the need for such information and the author's role in ensuring that it is used appropriately in each instance.

4. Provide ways of checking and correcting inaccessible content.

Many authoring tools allow authors to create documents with little or no knowledge about the underlying markup. To ensure accessibility, authoring tools must be designed so that they can identify inaccessible markup, and enable its correction even when the markup itself is hidden from the author.

Authoring tool support for the creation of accessible Web content should account for different authoring styles. Authors who can configure the tool's accessibility features to support their regular work patterns are more likely to accept accessible authoring practices. For example, some authors may prefer to be alerted to accessibility problems when they occur, whereas others may prefer to perform a check at the end of an editing session.

5. Integrate accessibility solutions into the overall "look and feel".

When a new feature is added to an existing software tool without proper integration, the result is often an obvious discontinuity. Differing color schemes, fonts, interaction styles, and even software stability can be factors affecting author acceptance of the new feature. In addition, the relative prominence of different ways to accomplish the same task can influence which one an author chooses. Therefore, it is important that creating accessible content would be a natural process when using an authoring tool.

6. Promote accessibility in help and documentation.

Web authors may not be familiar with accessibility issues that arise when creating Web content. Therefore, help and documentation must include explanations of accessibility problems, and should demonstrate solutions with examples.

7. Ensure that the authoring tool is accessible to authors with disabilities.

The authoring tool is a software program with standard user interface elements and as such must be designed according to relevant user interface accessibility guidelines. When custom interface components are created, it is essential that they be accessible through the standard access mechanisms for the relevant platform so that assisting technologies can be used with them.

Some additional user interface design considerations apply specifically to Web authoring tools. For instance, authoring tools must ensure that the author can edit using one set of stylistic preferences and publish using different styles. The style preferences of the editing view must not affect the markup of the published document. Authoring tools must also ensure that the author can navigate a document efficiently while editing, regardless of disability. Authoring tools should therefore provide an editing view that conveys a sense of the overall structure and allows structured navigation.

Compared to a non-native SGML authoring tool, an SGML-aware authoring tool should have benefits for authors as they deliver real-time comprehension of and compliance to SGML semantics. A good SGML editor constantly maintains validity and provides real-time contextual aids to authoring /13/. What tool support may be required for an author may correspond to useful features for readers, and vice versa. Dual requirements are important because they help design more complete user interfaces for browsing and authoring tools, and they also reduce the total design effort as the two otherwise disparate areas are unified /60/.

The challenges in developing authoring tools are designing, authoring and maintaining complex entities that are still consistent and ensure hypertext usability issues. The tool should be able to help structure the information; for example, some sort of map of the structure helps seeing the depth of the structure /58/. To help hypertext authors cope with the vast number of nodes and hypertext links, tools should make it easy for authors to create and label nodes and links so that designers can concentrate on design guidelines and principles. Like all other electronic tools, a comprehensive range of editing facilities such as copying, moving, insertion, deletion, formatting, etc. should be provided for users. Facilities should also be provided to import and export text or graphics files. As the hypertext database grows in size, it will be difficult for authors to keep track of the nodes and links. It will be a tremendous help to authors if the tools are able to keep track of what the authors have done so far and capture the hypertext database with its nodes and links in a graphical form with the choice of viewing it on the screen or be printed out. Tools should allow links to be established with external facilities with ease. During the authoring process, a very essential feature an authoring tool should have is the capability to toggle between author and user mode so that authors can test ideas. In interactive systems, support for capturing and representing users' needs help in further design of hyperdocuments /59/.

5 INTRODUCTION TO THE CASE STUDY ENVIRONMENT

5.1 Structured Documentation Practices in Nokia Networks

Nokia is a Finnish industrial company applying SGML in the production of technical documentation. The rapid growth of Nokia sets all the time new requirements to the SGML based technical documentation. DTDs are in the core of these solutions and they were already available for PMR. Also the SGML tools had already been chosen before the actual SGML migration project.

The process of writing an SGML document involves the creation of the document content using an authoring tool of choice, the appropriate DTD and appropriate templates created for the DTD. Once a version of the document is created, it is archived in the database. The publishing of the document means applying the appropriate layout rules to the document according to the distribution medium /57/. The Figure 18 illustrates the general writing process used in Nokia Networks drawn with the notation of the thesis.

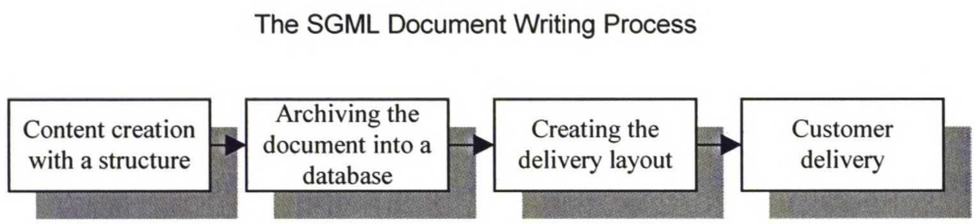


Figure 18. *The SGML Document Writing process in Nokia Networks /57/*

The notation of process illustrations used in this thesis consists of blocks and connectors. Blocks illustrate process phases and connectors illustrate the transition from a phase to another. A connector may also lead backwards as a back feed to some of the previous phases. There may also occur conditional statements. If the output of a phase is not acceptable for the next phase there may occur a back feed called NOK (Not ok). On the other hand, if the output is acceptable, there may occur a transition called OK.

5.2 Old Environment

5.2.1 The Present Customer Documentation Process of PMR

The PMR customer documentation process is a subprocess of the product process. The process covers a set of activities when documentation is updated in connection with new system or network element releases, or in connection with R&D projects or product programs.

In the product design and prototyping phase the technical writers receive their first input information from the software designers for the documentation. With the given information writers review existing documents and evaluate needs for changes, such as need for improvement, increased clarity or structural changes. In the product integration phase, writing continues. Document drafts are used in functional tests of the product. During the functional testing draft documents are reviewed in customer document review meetings with participants from the R&D department. During the piloting phase of the product process, the documents are finalized according to review results and officially checked and approved. However, some documents can be ready, approved and archived even before the last product milestone. After the approval of a document it may be translated into some other language, if needed. In most cases, translations are done by subcontractors. There is an independent translation subprocess defined for that purpose.

The customer documentation process document does not define the actual work tasks needed to accomplish the pre-defined milestones. For that reason, interviews were needed to understand how the technical communicators and other documentation development related staff actually perform their daily tasks that were not documented in any process descriptions or even known or controlled by any single staff member. The documentation process that was detected with the interviews and their analysis is illustrated in Figure 19 and it is examined more closely in the following subchapters.

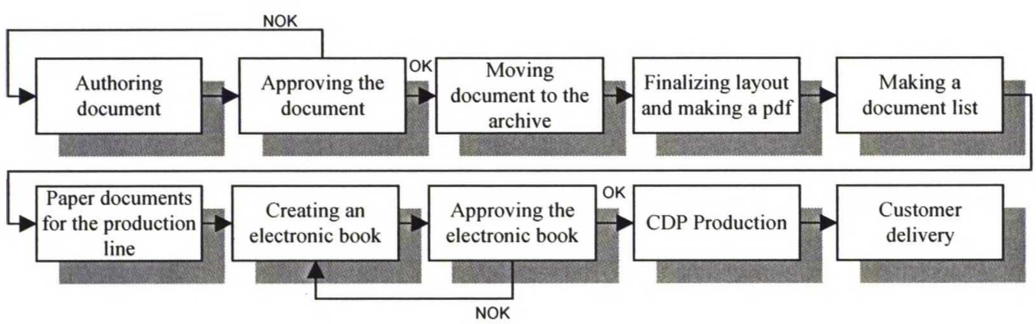


Figure 19. *Legacy Documentation Process*

5.2.2 Documentation Planning for a Product Release

In the feasibility study phase of the product process, the documentation project responsibles are chosen and effort estimations are made. Also the structure of documentation is given on a general level. In the product specification phase the project manager collects information, evaluates the existing documentation and studies the product specifications so that the needs for change can be identified. The documentation plan and finally the customer documentation project plan are made based on this information. Broader analysis of the planning phase is out of the scope of this study.

5.2.3 Authoring and Approving of the Documentation

Technical writers may have different kinds of tasks to accomplish with the documents they are responsible for. Some documents may be completely new for the product. Most of them are just updates to the documents written for the previous product release. In addition to writing completely new documents and updating and correcting previous documents, some material comes from software designers as ready-made that just has to be copy-pasted into the document in question. Customer documents are written and updated by certain dedicated writers for every release. More experienced writers usually take care of completely new documents and others do mainly document updates.

The writing and editing process at the computer is only a part of the work of a technical writer. In many occasions, background research and planning takes the majority of the time spent on a document. Software designers provide some information themselves but some material has to be collected through meetings and review sessions. In some cases, technical writers need to go to the test labs to test the examples in the document in order to find out if they work as they have been written. The figures for the documents are drawn by either documentation assistants or technical writers themselves. The working mode depends on each writer's preferences.

Tools issues and data management is also a part of a writer's everyday work. There may be some tool problems or a new thing to learn with tools every once in a while. Data management is an important part of the workflow since version control and accessibility are critical in case of illness or job rotation. Also data backups and appropriate naming conventions can be critical.

After the content of a document has been written and it has been commented and reviewed by a specialist of the field, the PDF file is sent electronically to the designated checker and approver for an approval. If there is still something to correct or add, the document needs to be updated by the author. After the iteration round, the final approved document and its approval form are delivered from the author to a documentation archivist.

5.2.4 Postproduction of the Document

When the archivist receives the notification, the document and all the related files are moved to the archive, which is accessed only with read-access by technical writers. After the move, the archivist finalises the layout of the Interleaf format document and makes the final PDF out of it. After this phase, the archivist updates the document information such as document code and version, in a document list. Document lists are based on product-specific document classes. When all the documents related to the product release are done to this point, the archivist makes updated PDF files out of all the document lists. The final task for the archivist is printing paper documents of all the release-related documents for the use of the production line and copying house. At this point, the archivist informs a document assistant that an electronic book for the product release CD-ROM can be made.

5.2.5 Producing the Electronic Book

The template for the electronic book is done with the Adobe PageMaker 6.5 desktop publishing program. The electronic book consists of the cover page, which has links to previously mentioned document lists and the PDF documents. The document lists have links to the actual document files. The cover page of the final electronic book is illustrated in Figure 20.

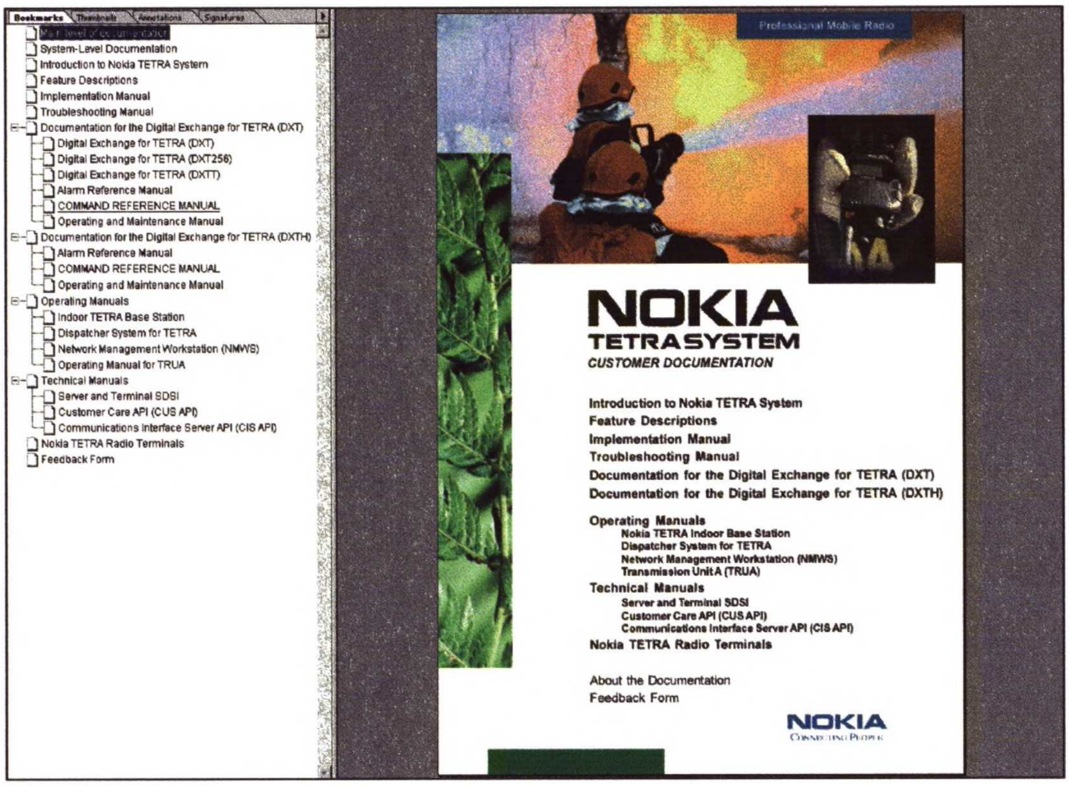


Figure 20. *The Cover Page of PMR's English Customer Documentation Set*

Creating the electronic book for a new release usually starts from the basis of the previous electronic book. Document lists are replaced with the updated ones and the links to the actual documents are revised. Bookmarking of document lists is needed as well as making heading bookmarks manually within single documents. This is needed because PDFs are produced by printing Interleaf documents with Acrobat Distiller printer driver. The procedure does not output the desired heading bookmarks automatically. When the electronic book is constructed and ready, it is sent for an approval to product program managers and product line managers. When everything is set, the assistant sends a notification to the NET-wide Customer Documentation Production (CDP) unit that the product release documentation set is ready for the customers. From that point, CDP staffs take care of the production of CD-ROMs and delivery packages. The maximum time that the mass production phase may take is two weeks but in most cases it has taken only two days. The production time depends heavily on the CDP's backlog of that moment.

5.2.6 Documentation Tools

Writing the actual authoring process is based on Interleaf 6.4 word processing environment as seen in Figure 21. Interleaf can be seen as a semi-structured word processor as it demands the user to use pre-defined styles for each and every document component. However, styles can be modified by the user and style attributes are based on the visual appearance. In other words, logical structure does not exist in the sense of structured documentation. Some documents are also processed with MS Word but they are not in the scope of this study or the SGML migration project overall.

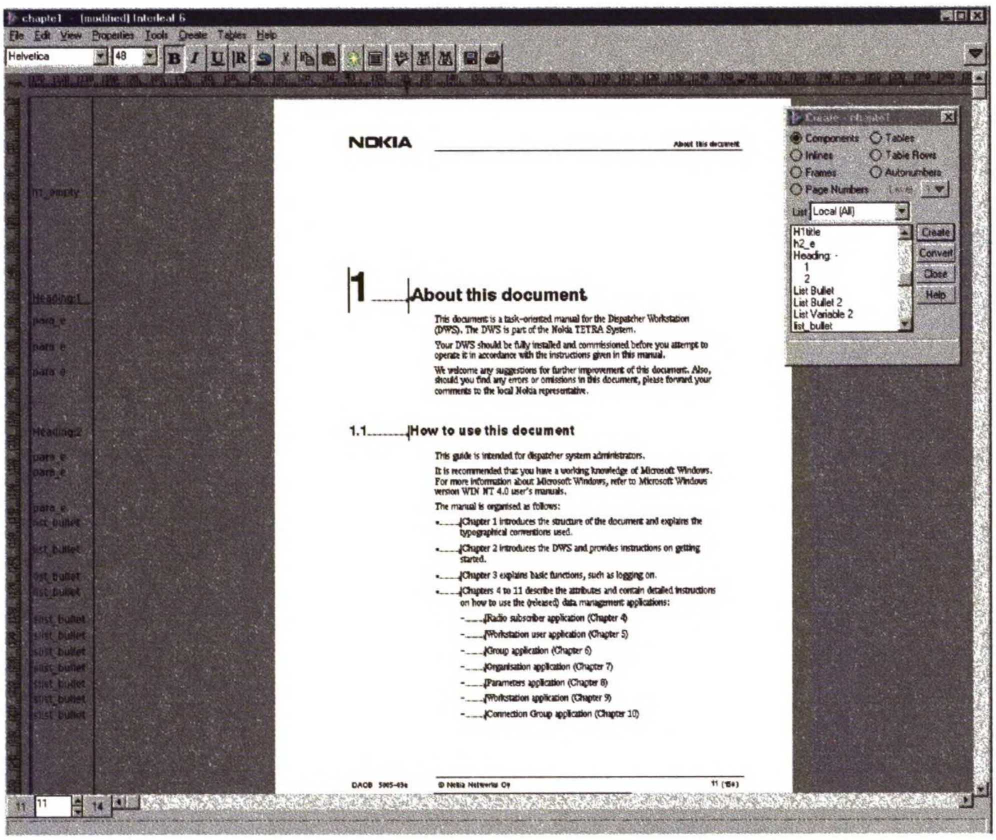


Figure 21. *A View of the Interleaf Word Processing Program Seen by a Writer*

Documentation graphics are drawn with both Micrografx Designer 7 and Visio 5. Some graphics have also been drawn with Interleaf's own drawing tools. Because of constant vector format problems, graphics have mostly been imported into Interleaf documents in TIFF format. The drawback is that the original drawing tool specific vector formats must also be saved in a separate folder along with the TIFF graphics folder in case of forthcoming updates. Screen captures are also widely used and they are processed with Paint Shop Pro 5 image processing program.

Writers have their ongoing writing projects in their Interleaf desktops. These desktop files are also stored on a network drive that is considered a safe place in case of crash situations. Documents are archived electronically on a separate server in both Interleaf and PDF format. PDF files are produced with the printer driver feature of Adobe Distiller program. Model versions of every single approved document in every supported language are archived in the physical archival depository. There is no document or version management system being used at the present.

5.3 New Environment

5.3.1 The New Customer Documentation Process of PMR

The new PMR customer documentation process is based on the previous version as discussed above but it was written in a much more detailed level defining actual work tasks, methods and workflows. Still, the change of documentation technology did not affect essentially the phase outputs and pre-defined milestones of the documentation process. The word process should be understood, in this context, as a workflow or as a low-level process that passes documents, information and tasks from one person to another.

When the legacy documentation process, presented in the Chapter 5.2 Old Environment, was assessed, there appeared a need for developing a new way of working for the technical communicators. A new system and tools unavoidably change the way of working. Hence, it is better to make changes systematically than let them happen randomly /30/. The challenge is to implement a new process that is efficient and straightforward but also as simple as possible for the users of the system.

When the documents have been approved and archived in electronic form and the SISU (SIsältöSUunnitelma; content plan or documentation configuration) has been frozen, the customer documentation process stops and delivery process takes over. In practice, this means that these phases of the delivery process are not carried out in the local customer documentation department but centrally outside the organisation in the external CDP service. However, there are two more local phases of the customer documentation process before the actual customer delivery. Figure 22 illustrates the new customer documentation process taken into use during the SGML migration.

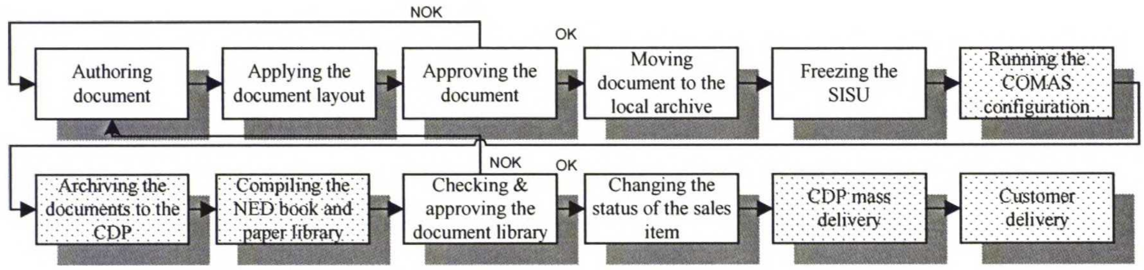


Figure 22. *The SGML-based customer documentation process. Delivery process phases are marked with dotted background to differentiate the local and centralised phases from each other.*

5.3.2 Authoring SGML Documents

There are three major DTDs used in Nokia for SGML-based customer documentation purposes. The chosen document type definition for PMR use is called Neutral DTD. Neutral DTD has been designed for general paper documents. Its grammar has been drawn from the analysis of existing paper documents. Therefore, the structure is generic, somewhat layout-oriented and contains structures familiar to writers of paper documents /57/. The original purpose of the Neutral DTD was use as a generic and conversion DTD, although later it has also been used for authoring /48/.

The basic ideas and responsibilities of authoring remain unchanged compared to the legacy procedure described in Chapter 5.2.3 Authoring and Approving of the Documentation. Besides a new authoring editor, a noteworthy difference is migration from the Interleaf desktop file system into Windows-based file handling.

The chosen SGML authoring editor in the study environment is Adept and its current version 8.2. Adept is a native SGML application, which means that users work in a true SGML standard environment with elements, attributes and entities. No conversion to other formats is needed in the opening or saving phase. Figures are inserted into a document as graphic entities, after which a link to the external figure file is inserted into the right position of a document. Tables are based on the CALS (Continuous Acquisition and Life-Cycle Support) definitions. Figure 23 illustrates the view from a technical writer's point of view.

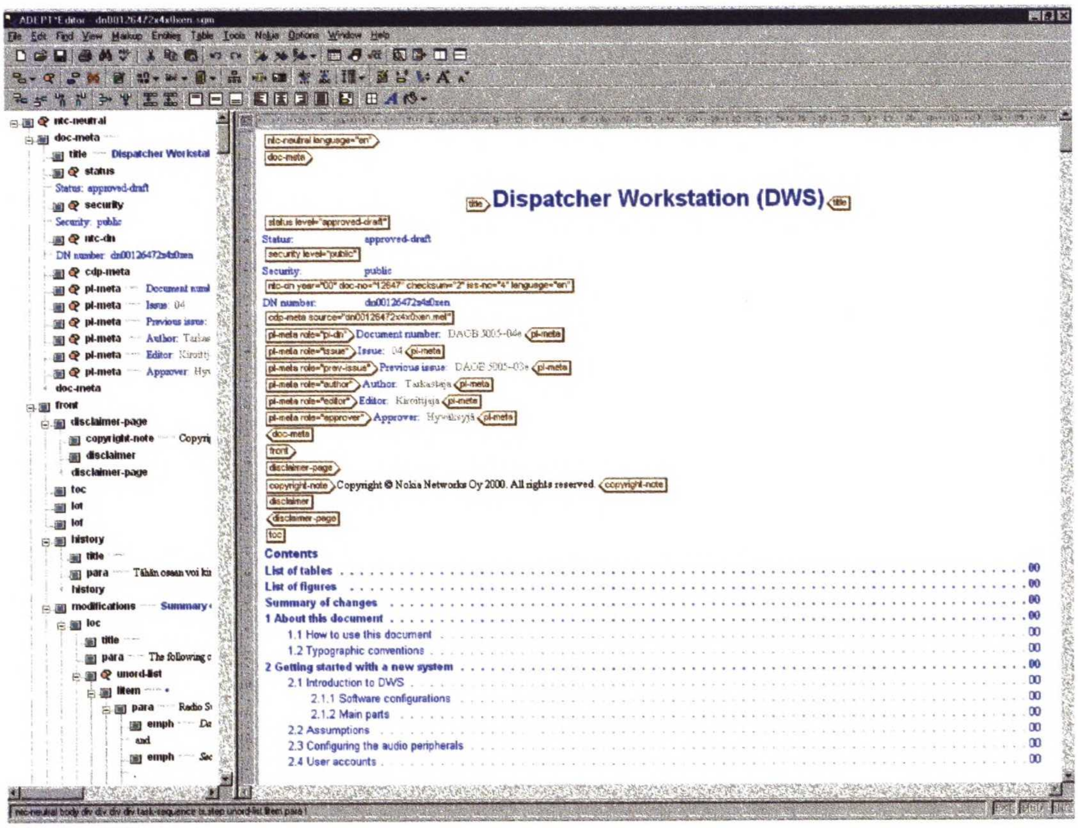


Figure 23. *A View of the Adept 8.2 Workbench*

DOMAS (Document Management System) is used for defining and managing the contents of the release, SISU, in question. The SISU ensures that correct versions of customer documents are included in a release. The DOMAS database has basic information on the documents and has information as to which configurations each document belongs to. One document can belong to several of them. SISU management is usually done by the customer documentation project manager.

5.3.3 Applying the Layout of the Document and Archiving the Document

Document layout rules for paper formats are defined so that uniform layout can be achieved. The technical implementation in Adept has been done with the FOSI standard. The SGML validation of documents and production of print-ready PDF documents is done with a centralised and automated validation and printing service (VPS). The VPS service produces the layout with the native Framemaker method. The validation phase may be carried out either by the author or a documentation assistant.

The output PDF file of VPS system is used for the approval as is. The approval is to be done as soon as possible. It requires an extra effort from a technical writer to get the response from the checker and approver quickly. When the archivist receives the notification that the document is approved, the document and all the related files are moved to the local server-based archive. However, a documentation assistant always checks that the document meets all the requirements of the archiving system. Mainly this applies to the layout rules and naming conventions to be followed.

5.3.4 Freezing the SISU and Publishing the Documentation Release

Freezing the SISU means freezing the documentation configuration of a release in question. Usually this happens in the last piloting milestone of the product process. The freezing is carried by the customer documentation project manager of that particular release. The status of the updated documents are changed right before the freezing in the "approved" state for the release by an archivist or a project manager and the final documents are copied to a shared file server. A notification of the documentation status and processing orders is sent automatically to the customer documentation production unit.

This chapter explains the three phases of the delivery process that are not locally taken care of. Thus, they belong to the delivery process. Possible delivery process improvements are not considered in the context of the current project. The two other phases, checking and approving the document library and changing the status of the sales item, occur back again locally in the PMR unit. Checking and approving the document library is done as a team effort by local technical communicators and the project manager and changing the status of the sales item is done by a person representing the PMR logistics organisation. If some deficiencies, such as missing figures or an inappropriate layout, are found from the document library, they must be corrected back in the authoring phase.

The document production system, also known as the Customer Documentation Production (CDP), gets an extensive printout of the document library of the release from a separate Configuration Management System (COMAS). According to this information, the document production staff get the original and finalised SGML documents from the PMR's local archive and archives them to the official CDP system. After that, PMR receives a Dynatext book and paper library of the release. These publishing products still have to be approved by a checker and approver in the PMR customer documentation department before the documentation is finally ready for the delivery. If there are some deficiencies, mainly with the layout of figures, tables or references, there might be a need for going back to the early phases of the process to make the corrections. The delivery is done by the CDP organisation according to the sales item orders of the PMR logistics department. The mass delivery means, in this case, the mass production of the CD-ROM deliveries.

5.3.5 Documentation Tools

Besides Adept SGML editor, there is also another important new tool taken into use. It is called Documentation Workbench (DWB). It is meant to be the main platform for SGML authoring and file management environment but many of its tools such as Adept and Visio can be run as stand-alone. DWB includes also some project management properties but it does not really offer any help in the documentation process or workflow. DWB is under development all the time and some progress in usability and writer-related issues may be expected soon. Laakso /30/ has more closely described the development work done with the DWB in his thesis.

Drawing and image processing tools remained unchanged after the migration. Previously, only TIFF-format was used for storing images. SGML enables also the usage of CGM (Computer Graphics Metafile) vector graphics format. Consequently, a DWB-based tool called CGMfix was developed for correcting deficiencies caused by the CGM filters of the current drawing tools.

5.3.6 Maintaining SGML Documents

Model versions of every single approved document in every supported language are archived in the physical archival depository. There is a new naming and versioning system taken into used for both documents and figures. Also an extensive use of metafiles and document history tracking has been started. Every SGML document has the possibility to include a history metadata element, which helps the document to carry information about itself. Metadata may include information such as release info, contact people or information about upcoming changes. All the figures used in documents are now stored in the common file system. There is no document or version management system being used at the present.

6 MEASURES OF THE DELIVERY CYCLE OF DOCUMENTS

6.1 Experimental Design for the Measures

6.1.1 Experiment Goal

The goal of the experimental study of this thesis is to find out the rapidity and simplicity of a structured documentation process. These two factors, rapidity and simplicity, are considered as characteristic of a structured documentation process from the engineering approach. Measures in the old and the new environment were done, respectively, to be able to analyse the changes which occurred in the migration. Experiments were made both in the authoring phase and the postprocessing phase. The delivery cycle of documents can be calculated by adding up the total time of these two phases together.

In order to reach the experiment goal, the authoring measures of different documents should somehow be comparable. Because the exactly same updates cannot be done to all of the documents, the comparison was done by calculating the time spent on authoring a page of the document in question. The ration is not exactly truthful, since updates may not be distributed evenly all over the document but there might be changes or additions in one of many chapters. Thus, the ratio (h/page) is a compromise and considered as a precise enough comparison value for this case environment.

6.1.2 Experiment Circumstances

The chosen process measure method was time measures of the process phases. The reasons for selecting this metric were that time is a relatively precise unit and easy to measure, and thus results can be easily compared to parallel measure results. In addition, the technical writers have become used to hour reporting in their work to some extent already, so the method is familiar.

The authoring measures were conducted in three parts within half a year, a period which included two document releases. Those three parts consisted of one Interleaf document and two consecutive SGML documents. Five voluntary technical writers were chosen to be part of the experiment for all the consecutive phases.

In the both two phases, authoring and postprocessing, time was considered the most relevant metric. In a project-oriented environment, the time spent on the documentation reflects well the needs set to meet a deadline. The authoring process in both environments was divided into three categories: writing at the computer, background research, and tools and management. Tools and management category includes the time spent on tool issues and file management. Also total writing time was calculated from the measured values. The writing at the computer function itself was divided also into three subcategories depending on what kind of update the particular document needed. Those categories were: writing new material, copy-pasting new ready-made material and updating material. Updating material includes correcting, replacing and removing material or restructuring of a document. The metric in this case was percentage. Presentation of the test points is given in Table 4.

Table 4. *Test Points of Measuring Writing Efficiency*

Old process	**Test points**	**Number of people**
Duration of authoring process	Total writing time (h), writing at the computer (h), background research (h), tools and management (h)	5
Classification of pure writing	Writing new material (%), copy-pasting new ready-made material (%), updating material (%)	5
New process, sample 1	**Test points**	**Number of people**
Duration of writing process	Total writing time (h), writing at the computer (h), background research (h), tools and management (h)	5
Classification of pure writing	Writing new material (%), copy-pasting new ready-made material (%), updating material (%)	5
New process, sample 2	**Test points**	**Number of people**
Duration of writing process	Total writing time (h), writing at the computer (h), background research (h), tools and management (h)	5
Classification of pure writing	Writing new material (%), copy-pasting new ready-made material (%), updating material (%)	5

The postprocessing process of the legacy Interleaf environment was divided into eight categories: approving the document, moving to the archive, finalizing layout and making PDF files, making a document list, printing paper documents for product line use, creating the electronic book, approving the electronic book and CDP production. The latter three categories are common for all the documents in the release, so only one measurement of these parts could be made. Total postprocessing time was calculated from the individually measured values. Illustration of the test series is given in Table 5.

The postprocessing process of the new SGML environment was divided into seven categories: applying the document layout, approving the document, moving the document to the archive, freezing SISU, CDP production, checking and approving the document library and sales item, and CDP mass production. The latter four categories are common for all the documents in the release so only one measurement of these parts could be made. CDP production covers all the delivery process phases described in Chapter 5.3.4 Publishing the Documentation Release, except that only CD-ROM production is included in the measures because it is the main product, as already in the legacy system. Total postprocessing time was calculated from the individually measured values. Presentation of the test points is given in Table 5.

Table 5. *Test Points of Measuring Postprocessing Efficiency in Interleaf Environment*

Old process	**Test points**	**Number of documents**
Duration of archiving	Total time (h)	Release
Classification of archiving	Approving the document (h), moving to the archive (h), finalizing layout and making PDFs (h), making a document list (h), printing paper documents for the product line (h), creating electronic book (h), approving electronic book (h), CDP production (h)	Release
New process	**Test points**	**Number of documents**
Duration of archiving	Total time (h)	Release
Classification of archiving	Applying the document layout (h), approving the document (h), moving the document to the archive (h), freezing SISU (h), CDP production (h), checking and approving the document library and sales item (h), CDP mass production (h)	Release

6.1.3 The Methods and Tools of Measuring the Data

The writing phase was measured with the co-operation of five writers who all donated one of their documents into measuring, so the total number of documents was five for each of the three authoring measures. The writing process was measured by the writers themselves. They were given an electronic Excel spreadsheet for writing down their work hours for the chosen document on a daily basis. The sheets were personal for everyone so that the test results would be objective in the sense that measures were unbiased. The spreadsheet is given in Appendix A.

The postprocessing phase was measured with the co-operation of two documentation assistants that measured their work with five documents. In most cases, not including two exceptions, the actual number of work hours for that phase was measured. In the approving the document -phase, hours were measured from the beginning to the end (24 h/5 days a week) in spite of the actual work hours and duration of the daily work. In both CDP production phases it was assumed that they work 7.5 h each day. The assistants were given one electronic Excel spreadsheet for writing down their work time for the chosen five documents and another one for writing down their work time for the document set of a release. In other words, both of these sheets cover document-specific and common subcategories mentioned earlier. The sheets for postprocessing a single document also had the following information: assistant's name, document type (Interleaf or SGML), document code and document length. Respectively, the sheets for postprocessing a document set had the following information: assistant's name, document type in the release (Interleaf or SGML) and the number of documents in the release. All the postprocessing spreadsheets are given in Appendices B and C.

6.2 Results and Result Analysis

6.2.1 Interleaf-based Process

The complete measuring results can be found from Appendices D and E and the calculation results in question can be found from Appendix F. Measuring the Interleaf based process raised some instant remarks. Most of the tasks of technical writers are updating existing documents and not authoring new material. The amount of background work varies a lot between different writers and documents, which is quite natural since every authoring task is a unique one. In most cases, the background work also concentrates on the early half of the authoring phase, which also can be seen as a quite obvious result. Also the more work a document demands from a writer, the less time is proportionally spent on the tools and management category. Thus, it seems that tool problems are quite minor at the present and file management has become a somewhat routine task that is well-understood and does not cause major difficulties.

When the time spent on authoring a page of a document (h/page) in Interleaf environment is compared to the proportion of new material authored for the document update (%), the coefficient of determination R^2=0.9634. The calculations show that those two factors have correlation with the 99% confidence interval. Figure 24 illustrates the relationship. Thus, the duration of the authoring phase in the well-established Interleaf environment is highly dependent on the amount of new material added into the document.

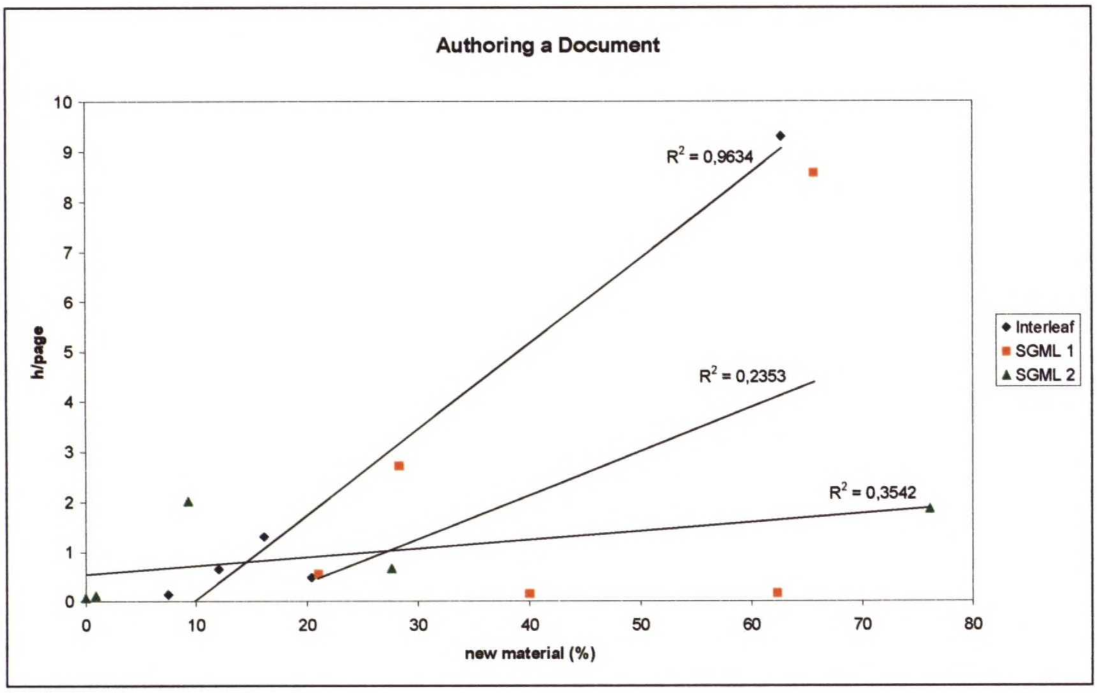


Figure 24. *Comparison of hours spent on authoring a page of a document with the proportion of new material authored for the document update. Data for five writers and documents. The five writers all updated different documents, with different kind of updates, for these two releases so that the same documents could not be measured consecutively.*

The amount of background research and planning (R^2=0.9632) and tools and management seem to have a correlation to the amount of new material. The first correlation is illustrated in Figure 25 The calculations also show that those two factors have correlation with the 99% confidence interval. This is quite natural, because both of those categories require substantially the most effort when new material is produced. If the data point with h/page value of 5.7 is removed from the calculations, the R^2 value is still 0.9632.

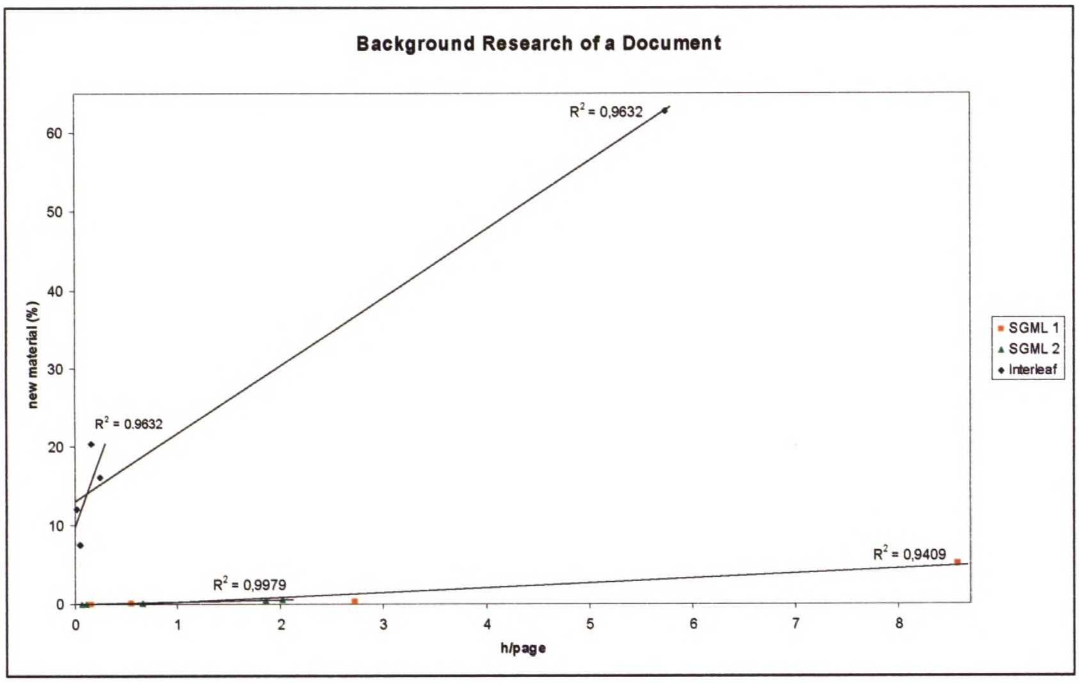


Figure 25. *Comparison of hours spent on background research per page of a document with the proportion of new material authored for the document update*

When examining the postprocessing phase measure results, the first remark is that the approval takes an unacceptably long period of time. Since the deadlines are tight and all the phases so far have been accomplished within the schedule, it feels quite irrational to keep the process suspended for such a long time. The average time spent on checking and approving five documents was in this case 29.8 hours, which shows that the approval phase took 98.2 % of the total postprocessing time. The standard deviation of the measured values was huge, 30 hours, which means that the approval time clearly depends on the checker's and approver's time schedule and the complexity of the document under inspection. Naturally, the product release includes mostly many other, and more critical, matters than customer documents that also have to meet the deadline; the approval may not be the first priority task. However, there might be a solution to the bottleneck in better communication between the employees concerned.

When leaving the document approval phase out of the calculations, the longest time period, 52.6 % with 0.40 hours on the average, was spent on finalising the document layout and making a PDF file out of it. The second most time-consuming phase, with 21.3% or 10 minutes, was printing out the paper documents for the use of the production line. However, all the document-specific postprocessing phases, without approval, lasted an hour as an average per one document, which is not relatively a large part of the whole customer documentation process. Complete results are presented in Appendix F.

Postprocessing a document set for the release seemed to be quite time-consuming and laborious. The reasons for the hard work needed was already described in Chapter 5.2.5

Producing the Electronic Book. The editing and creation of the electronic PDF book out of the document set took 145 hours. When authoring a document took 168.1 hours as an average and postprocessing a document took 30.3 hours as an average, the postprocessing per single document of a complete document set took only 1.50 hours. The total time postprocessing the whole document library set was 223 hours.

The average efficient delivery cycle for one document in the Interleaf based process was calculated to be 199.9 hours. Of this value, 168.1 hours (84%) was spent on authoring and 31.8 hours (16%) was spent on postprocessing. The average value is quite imprecise, because the sample size was only five documents. Still some trendsetting results can be seen.

6.2.2 SGML-based Process

The SGML-based process measures are based on the second SGML authoring measures since the user experience corresponds more with the Interleaf environment. The complete measuring results can be found from Appendices D and E and the calculation results in question can be found from Appendix F. In the SGML environment, most of the authoring consists also of updating material but in this release, also some new documents were introduced, which raised the amount of new material also in the case of measured documents. Tools and management seemed to take about twelve percent of the total authoring time when we leave one exceptionally large, possibly erroneous value out of the calculations. The reason for neglecting that value out of the results is that it is based on a measure that was done to very minor update of which 14 out of 16 hours was spent on tools and management. The more work a document requires the less time is proportionally spent on the tools and management.

When the time spent on authoring a page of a document (h/page) in SGML environment is compared to the proportion of new material authored for the document update (%), the coefficient of determination is $R^2 = 0.3542$. Figure 24 illustrates this relationship as well. The calculations show that those two factors do not have correlation with even the 95% confidence interval. The reason for the low correlation may be that with new tools, methods and environment there is more unpredictable variation in the time needed to produce new material. The amount of background research and planning seem to have a strong correlation, $R^2 = 0.9979$, to the amount of new material. As the calculations show, these two factors have correlation with the 99% confidence interval. The reason for this is that the amount of background research has a direct effect on the amount of new material in the document. In addition to this, the measured documents had a relatively large amount of new material in them.

In the postprocessing phase, the first remarks are that the approval still takes an unacceptably long period of time and applying the layout with the VPS system is relatively time-consuming as well, since the average is about 50 minutes for one document. As a rule of thumb, it often takes even 2-3 hours because of long queues. This can be critical to the already tight schedules and writers feel very uncomfortable when they just have to wait for the PDF document production. The average time spent on checking and approving five documents, in this case, was 86.7 hours, which shows that the approval phase took 98.8 % of the total postprocessing time. The standard deviation of the measured values was large: 101 hours, which means that the approval time must be heavily dependent on the checker's and approver's time schedule. Of course, completely new documents might be laborious to check, but most of the time this is not the case.

When leaving the document approval phase out of the calculations, the longest time period, 84.0 % with 50 minutes on the average, was spent on making a PDF file out of the SGML source document. The second-longest phase, with 10 minutes, was moving a final document to the archive. Document-specific postprocessing phases, without approving included, lasted an hour as an average per document which, relatively, is not a large part of the whole customer documentation process. The problem is mostly the VPS system, used for validating and printing, which makes the process unpredictable.

Postprocessing a document set for the release was a completely different subprocess than before in the legacy environment. The two local phases, namely freezing the SISU and checking and approving the document library and approving the sales item, took 8 and 16 hours to accomplish. Checking and approving documents depends highly on the size of the document set and on the amount of completely new parts in the set, which need to be checked more carefully. When authoring a document took 40.8 hours on average and postprocessing a document took 87.6 hours on average, the postprocessing of a single document of a complete document set took only 0.9 hours.

The average efficient delivery cycle for one document in the process was calculated to be 128.5 hours. Of this value, 40.8 hours (32%) was spent on authoring and 87.7 hours (68%) was spent on postprocessing.

6.2.3 Comparison of the Process Measures

The comparison of the process measures includes only the latter SGML authoring measures, since they reflect more a stable authoring environment and the inexperience of the new tools and process did not bias the results so much. The SGML measures are compared to each other in Chapter 6.2.3 Comparison of the Process Measures.

The statistical difference of the authoring environments can be calculated by comparing the means of the populations. Thus, the interest is in testing the null hypothesis that the population means are equal. Thus, we are testing

$$H_0\!: \mu_i = \mu_j \qquad H_1\!: \mu_i \neq \mu_j \tag{2}$$

where, H_0 is the null hypothesis

H_1 is the alternative hypothesis and

$\mu_{i,j}$ is the average number of hours spent on writing a page of a document.

The results can be seen in Appendix F. As the F value is 0.49 and the P value is 0.51, the H_0 cannot be rejected in the risk level $\alpha = 0.05$. Thus, since higher risk level is statistically not accepted, we have statistical evidence that these two authoring environments do not significantly differ from each other. The reasons for this can be the small sample size and writers' good previous knowledge of word processing applications. The h/page averages (Interleaf 0.97, SGML 0.56) may give a slight impression that writing became a little faster in the SGML environment. However, the small sample size and the chosen documents may also bias the impression.

When examining the authoring measures of individual writers, an overall impression is that a larger part of the total time is in spent on tools and management in the SGML measures (27.4%) than in the Interleaf measures (8.6%). That might indicate that SGML tool issues still take some time and also that document management is slightly more laborious in the SGML environment. On the other hand, the amount of new material, 25.1%, used in the documents was larger in the SGML release than 23.7% in the Interleaf release. The difference is so small that no conclusion can be drawn from that.

As can be seen from Figure 26, the total delivery time is shorter in the SGML environment. However, the Interleaf and SGML delivery cycles can not be compared to each other directly, since authoring and approving a document are such case-specific issues. When the standard deviations of the authoring phases are compared, it is found that the Interleaf value 193.7 is noticeably higher than the SGML value 27.4. The most probable reason for the difference is the nature of releases. In the SGML release changes were smaller and all changes were quite similar of size. In the Interleaf release, the changes in some documents were quite extensive. The comparison of standard deviations of document postprocessing times shows that SGML documents had higher mutual differences in postprocessing times. The reasons for the differences were longer approval time and the unpredictable duration of applying the layout. Still, the positive result is the postprocessing a document of a document set -phase, which shows that automation in the publishing end of the SGML process speeds up the total process as desired. The standard deviation cannot be calculated for document sets since they occur only once in a release.

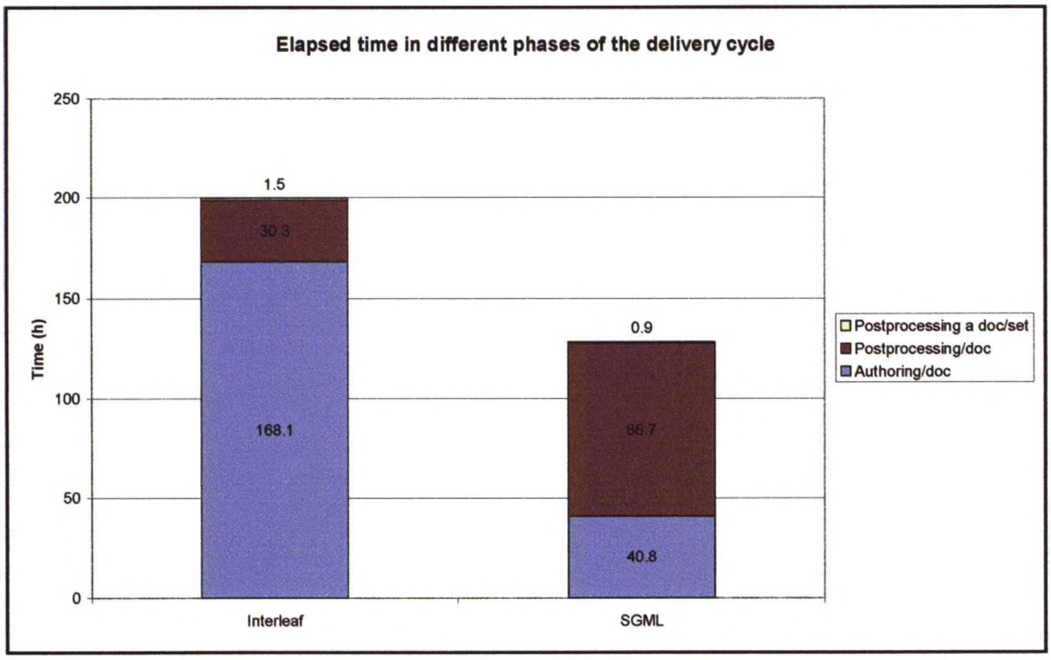


Figure 26. *The comparison of the elapsed time in different phases of the document delivery cycle in Interleaf and SGML environments. The times are calculated per single document in consecutive document releases.*

As the number of local process phases has decreased from previous eight to current seven, there have been some changes in the relative duration of the phases. Although the number of phases changed only a little, the local process as a whole is less resource-consuming. In this case, less time was spent on writing documents in SGML but the documents and the required changes are not comparable in any way. The same applies to approval of the documents, in which the Interleaf environment gave smaller results. Applying the layout was more time-consuming in the SGML environment. This was clearly because of the inflexible VPS system. Archiving documents had no difference between these two environments. The total delivery time per one document of the local phases, without authoring and approving, was 1.6 hours in the Interleaf environment and 1.2 hours in the SGML environment. This result shows that the new process is somewhat faster than the old Interleaf process. However, it obviously seems that it is not yet working as effectively as possible and also that because of the relatively small size of the product line, the local document production is quite flexible and quick, although it reserved a lot of local resources.

All in all, the documentation process became simpler and more straightforward in the postprocessing phases, mainly because the production phases were automated and centralised away from the local staff. At the same time, the duration of the local phases decreased as well.

6.2.4 Comparison of SGML Authoring Measures

The complete measuring results can be found from Appendices D and E and the calculation results in question can be found from Appendix F. When comparing the averages of time spent on tools and management, the second SGML document round gave higher result, 27.4%. Thus, the time spent on tools and management section can be seen mostly as a part of everyday life rather than beginners' difficulties.

When the time spent on authoring a page of a document (h/page) in SGML environment is compared to the proportion of new material authored for the document update (%), the coefficient of determination $R^2 = 0.2353$ for the writers' first SGML documents and $R^2 = 0.3542$ for the writers' second SGML documents. Figure 24 illustrates these relationships as well. It can be seen from both the figure and the regression analysis that the relationship is statistically insignificant in both cases, but it is a bit stronger within the latter documents. This may be a sign of learning when less time is used for wrestling with practical word processing difficulties.

The comparison of background research vs. new material (Figure 25) shows that the correlation becomes stronger in the second authoring round, which is natural when new material requires the most background research. The overall impression of the time used for tools and management is that most of it is spent in the first and last days of the authoring process. This seems quite natural, since naming and archiving conventions require some additional work just then.

The time spent on authoring a page of a document (h/page) next to computer was also tested with the analysis of variance (ANOVA) technique. The hypothesis for testing was

$$H_0: \mu_i = \mu_j \qquad (3)$$
$$H_1: \mu_i \neq \mu_j$$

where, H_0 is the null hypothesis

H_1 is the alternative hypothesis and

$\mu_{i,j}$ is the average number of hours spent on writing a page of an SGML document.

The results of the analysis are shown in Appendix F. When comparing the P-value 0.51 with the level of significance ($\alpha = 0.05$), it can been seen that the P-value is considerably larger, so that H_0 cannot be rejected. In other words, we have statistical evidence that these two authoring cases do not significantly differ from each other. The most important reason is that every technical writer knew about the basic concepts of a word processor already, which made them able to start "a fair bit up on the learning curve", as stated in the literature /41/. However, the h/page averages (first 0.98, second 0.56) may give a slight impression that writing became a little faster in the second round.

6.3 Sources of Error

The writing phase measures can be biased by slight misunderstandings of the categorisation and careless filling of the hour sheet. Thus, the result can be noisy data. Although the instructions were given, the authors were given freedom to choose their method of recording the data. For example, some writers gave only the total amount of hours spent on a document instead of daily work hours. Also the approval time may vary quite a lot if an approver has a busy schedule, very strict demands, or if the document is approved without actually checking the contents at that phase.

The sample size of five was quite small because of the restrictions of the case environment. Thus, the confidence level was not as high as it could have been in a more ideal environment. The chosen writers were considered as a representative set of technical writers but the most objective sampling would have been, of course, a completely random selection.

The postprocessing phase measures were not very error prone, because they were quite automated and straightforward to run. The people involved in the postprocessing phases were the only ones possible in the case environment. Inexperience with the new process could have caused some sort of error.

6.4 Summary of Results

A summary of the results of this Chapter is represented in Table 6.

Table 6. *Summary of Process Measures*

	Interleaf	SGML	
		SGML 1	SGML 2
Authoring, R^2	0.9634	0.2353	0.3542
Confidence intervals of R^2, authoring (first lower bound, then upper bound)	0.0587 (99%) 0.2839 (99%)	-0.2020 (95%) 0.3767 (95%)	-1.0443 (95%) 2.1317 (95%)
Background research, R^2	0.9632	0.9409	0.9979
Confidence intervals of R^2, background research (first lower bound, then upper bound)	2.9613 (99%) 14.4349 (99%)	0.0957 (99%) 1.1400 (99%)	0.2270 (99%) 0.3102 (99%)

Checking & approving per one document (h)	29.8	86.7
Standard deviation of checking & approving (h)	30	101
Authoring per one document (h)	168.1	40.8
Authoring per one page of one document (h/page)	0.97	0.56
Standard deviation of authoring documents (h)	193.7	27.4
Postprocessing per one document (h)	30.3	86.7
Standard deviation of postprocessing documents (h)	29.9	102.2
Postprocessing per one document of a complete document set (h)	1.5	0.7
Average delivery cycle of one document (h)	199.9	128.5
Total local delivery time without authoring & approving (h)	1.6	1.2
Percentage of authoring time spent on tools and management (%)	27.4	8.6
Percentage of the total writing time spent on writing at the computer (%)	23.7	25.1

7 USABILITY EVALUATION

7.1 Experimental Design of Usability Evaluation

7.1.1 The Evaluation Method and Evaluation Criteria

The chosen evaluation method was observation. The reasons for selecting observation were that it is easy to conduct, it does not consume very much time or other resources and it fits well to evaluating a the small-scale environment like the case environment is. The most relevant drawback, which is randomness of results, should have been avoided by telling the users about the minimum tasks they should complete and by using a questionnaire as an additional method.

The evaluation method for the test was chosen from the usability attributes introduced in Chapter 4.1 Definition of Usability. The chosen evaluation criteria are efficiency, memorability, lack of errors and satisfaction.

Efficiency is an essential characteristic of any kind of a process. Memorability can be noticed by observing how well users remember how to perform certain tasks needed during the customer documentation process. Errors may be very critical to the process that needs to be fluent and produce best possible quality for the customers. Subjective satisfaction is also important so that the system is pleasant to use and the good experiences encourage other technical writers to adapt to the new environment without fear. Because of the theoretical reputation of the term SGML among many technical communicators, the tools and processes should be easy enough to use. Otherwise they might want to go back to the old-fashioned word-processing environment that was more controllable by the users themselves. Thus, appealing interfaces are very much related to the subjective satisfaction.

7.1.2 The Selection of Test Users

The test users were technical writers with a university background, mostly with language degrees. They all were experienced writers, who knew well the conditions set to good-quality customer documentation. They represent well the average user of the system.

The selected five users were technical writers of whom two have approximately five and three have two months' experience authoring documents in the SGML environment. The writers had not grown accustomed to the shortcomings of the SGML process but were not total novices either. In the beginning of their SGML career, the writers had taken a one-day Adept editor course and a more informal two hour course of process and other SGML-related tools training. Also a variety of instructions were written and made available for the writers. They already had some sort of fluency in their work so that they knew what they should be doing and why. All users had a basic knowledge of SGML theory and, most importantly, of practical issues. The test was not supposed to measure how well the operating system or the tools are known but how well the usability of the tools and environment support the fluidity of the customer documentation process.

Nielsen /41/ has showed that in practical development five test users is often enough. When expert-user performance is measured, five users give a 70% probability of getting ±15% of the true mean. The number of test users was reasonable because there was no need to find out every single usability problem, only the major ones. Besides, the amount of users of the system in this case environment was more than five because the SGML migration was still ongoing. Despite all these restrictions, the test has been considered to have a practical value, although the confidence level of 95% used for research was not reached.

7.1.3 Test Environment

The test environment was the regular working environment of the test users and the system. This way the situation was natural, unlike in a test conducted in a test lab. The testing was done with the computer equipment and access rights that originally were owned by the users to maintain a normal working environment.

The test was conducted with PC workstations with all the necessary applications installed. The programs users needed in their work process were Adept editor 8.2, Documentation Workbench 1.2.75 and Lotus Notes 4.6. The last one was accessing the VPS database and server producing PDF documents out of SGML project folder. All the users worked as they would on a regular working day. All the necessary material and help was available. However, all use of this supplementary material was reported by the evaluator.

7.1.4 Evaluation Metrics

The most obvious usability metric is success rate when performing a representative task. The other two usability measures used here are task performance and subjective satisfaction. Task performance is measured by the amount of time it takes for a user to perform a task. Subjective satisfaction is measured by the user's answers to a questionnaire with free form statements.

7.1.5 Evaluation Situation

The evaluation was performed in Helsinki and Tampere during four days in two weeks' time. The new SGML process was just fully implemented into every technical writer's use. The test lasted three hours for each user. The meeting was started with a short briefing on what the test was about and what were the tasks that the user was asked to perform. The briefings were kept simple and uniform each time so that the opinions of the evaluator were not revealed and thus would not have affected to the user's operation.

The evaluation was performed by observing directly the regular daily tasks of the technical writers. However, the writers were asked, at any given time during the observation, to carry out the most important tasks: authoring an SGML document, running the VPS service and perhaps using the Documentation Workbench. The evaluator was not allowed to give instructions in problem situations unless no one else was able to solve them within a moderate period of time.

After the observation, also an inquiry was made by interviewing users about their tasks, the ways they use their SGML tools and what is the actual work process like. This was done to make sure that everyone encounters the same questions and can express their opinions impartially. In addition to this, open questions may reveal issues and opinions that were not considered while composing the closed questions. The questionnaire was performed with both closed and open questions shown in Table 7 and Table 8.

Table 7. *The closed questions for the usability study. Arguments and their counter-arguments are divided into categories based on the evaluation criteria.*

Category	Argument	Counter-argument
Efficiency	• The SGML process is efficient and eases routine tasks	• The Interleaf-based process was faster and smoother than the SGML process
Memorability	• The SGML process has few enough phases to be usable	• The SGML process is hard to use and it is easy to forget
Few errors	• The SGML tools are easy to use	• I have personal experience that the SGML tools include fatal bugs and features
Satisfaction	• The instructions for the SGML environment are sufficient	• SGML tools are complicated and the interfaces are confusing and visually poor

Table 8. *The Open Questions for the Usability Study*

1. Are the instructions for producing SGML documents sufficient? Are the terms used in the system familiar for the users?	6. Are any phases of the SGML process easing working substantially?
2. Is authoring documents easy enough?	7. Is the printing of documents troublesome?

3. Are the interfaces of different SGML related tools clear and visually appealing? Are the icons and figures used in the applications appropriate?	8. Are you honestly able to recommend the SGML environment to other technical writers?
4. Which functions are necessary and should be included in the SGML related tools?	9. Is there something essential that you feel that the SGML environment is missing compared to the Interleaf environment or else?
5. Are there any phases in the SGML process that are troublesome?	10. Any suggestions and comments?

For the closed questions, the Likert scale /46/ was used to measure the user opinions. The scale consists of five alternatives. The argument gets the better points the better the usability of the process in that particular area is. In each category, there is a counter-argument from different subject. Thus, half of the closed questions were counter-arguments. This was done because the questions were wanted as objective as possible and as non-leading as possible. The Likert scale used in the evaluation is illustrated in Figure 27.

5	4	3	2	1	N/o
Strongly agree	Slightly agree	Neutral	Slightly disagree	Strongly disagree	No opinion

Figure 27. *The Likert Scale Used in the Evaluation*

The questions were tested on a single person with the think-aloud method, which is considered a good way for identifying users' major misconceptions. On the basis of the pilot test, the questions in a misleading or confusing form were re-phrased and preconceptions were minimised as closely as possible. The complete usability evaluation questionnaire can be found from Appendix G.

7.2 Results and Result Analysis

7.2.1 Results of the Observation

After the results of the observation were collected and organised, it was clear that the observation findings could be classified into four categories. The categories are: findings requiring further training, process-related and tools-related findings and positive remarks. Findings requiring further training can be solved with right instructions and training, process-related problems need technical changes in the work environment, tool-related problems can be solved only by checking the settings of applications, informing the product vendors about the serious deficiencies and informing the writers with roundabouts to the problems. The first three categories have their most important findings italized. They are expanded upon in Chapter 7.2.4 Evaluation of the Results. Basic information on the tools of the old environment was given in Chapter 5.2.6 Documentation Tools and of the new environment was given in Chapters 5.3.2 Authoring SGML Documents and 5.3.5 Documentation Tools.

Findings requiring further training:

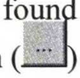
- The recent documents list in Adept's file-menu was not used by all writers, which caused extra searching when opening a document for editing
- Document map could be used more for copy-paste operations and navigation by most of the writers
- Copy-paste operations could be done by the drag-and-drop method as well
- The position of the cursor is more meaningful than some writers had understood
- The quality of figures seen on the screen is not the same as on a print-out
- *Graphic declarations are easier when copy-paste and browse options are taken advantage of*
- *TIFF figures should be cropped to as small as possible to avoid huge file sizes*
- The error message on the bottom line of the Adept editor was often neglected in error situations
- Document should be saved more often during authoring since it is simple and may save from a lot of grief in case of an accident. Automatic save was relied on too much
- The update for autonumbered fields was often forgotten, which caused confusing situations when checking the structure of that part of the document
- *Check completeness was not run by every participating writer*
- When running the check completeness option, the links showing the problem area were not even known to exist
- The search options of the SGML authoring tool were not fully known or used by all users
- *The Nokia layout rules, mostly considering figure and table sizes, were not fully understood yet*
- Different views in the VPS service were not fully understood by everyone, which caused less effective browsing when making a PDF

Process-related findings:

- *Adding a new DIV -element in the middle of a structure caused heavy problems*
- *The error-proneness of naming documents and figures with long and random-looking document naming codes*
- Recognising similar-looking and long file names coded with the special document naming convention is often difficult from a long file list

- When inserting a graphic, the figure-graphic-caption element combination caused some confusion
- Using and removing elements that were automatically generated as a required substructure of the document was considered a bit complex
- Sem-element was considered confusing, especially because the attribute changes are not shown on the screen as font changes
- Often a PDF document was produced just for proof-reading the contents
- The quality of figures had to be checked from paper prints
- Planning of a document structure was done mostly on paper
- *The meaning and editing of the metafile was considered confusing by some writers*
- *VPS validation messages were often neglected completely if the PDF transformation was ok, which may cause harm in the latter phases of the documentation process*
- Tables and lists without any attribute value cause fatal errors in the VPS even though the layout is fine without them
- When running an SGML document through the VPS system, the status "approved-draft" is unclear as a term
- Tables with frame=none attribute cause a needless VPS error, because the attribute is nowadays accepted as such

<u>Tool-related findings:</u>

- *When inserting attributes to a list element, a bug related to a selection menu was noticed*
- When a row is inserted after the last row of a table without borders, an extra border is inserted between these two rows
- *Inserting links is hard, because the list of key values given with the link tool is so messy*
- Links are not WYSIWYG, which makes it more difficult to see where they actually are targeted
- *Keywords that are wanted as index items in between text have to be re-written*
- Inserting captions to figures is cumbersome, because the cursor has to be in a too precise position to be able to accomplish the task
- *Figures are not WYSIWYG in Adept, which makes using them often error-prone*
- Figures sometimes have extra hair lines
- CGMfix should produce more informative and less technical messages
- The postscript check box of the VPS request form was not found necessary
- When retrieving a PDF file from the VPS system, the icon () for browsing the save folder is unclear
- Capital letters in document codes are not accepted by the VPS system

Positive remarks:

- Printing a selection of a document in Adept editor is well known
- One of the writers had a good way of working: PDF document was checked electronically and only the pages needing corrections were printed on paper
- Other documents were well used as examples when authoring more complex parts of a document
- Full tags vs. no tags views were used actively depending on the situation
- History element containing useful information about e.g. the latest update and contact persons of the document was made good use of
- Adept's spell checker was found useful and was thus used often
- Help material, such as print-outs, Lotus Notes stored guides and technical support person guidance were widely used by some users

7.2.2 Results of the Closed Questions

The results of the closed questions of the questionnaire are summarised in the Table 9.

Table 9. *The results of the closed questions*

Questions	**Average**	**Standard deviation**	**Median**
The SGML process is efficient and eases routine tasks	3.25	1.50	3
The Interleaf-based process was faster and smoother than the SGML process	3.40	1.34	4
The SGML tools are easy to use	3.60	1.14	4
The SGML process has few enough phases to be usable	3.00	1.41	2
I have personal experience that the SGML tools include fatal bugs and features	3.40	0.89	4
The instructions for the SGML environment are sufficient	3.60	1.34	3
SGML tools are complicated and the interfaces are confusing and visually poor	2.80	1.30	3

The SGML process is hard to use and it is easy to forget	2.40	1.14	2

7.2.3 Results of the Open Questions

The many available instructions were considered good and sufficient but a document describing the whole authoring process with all the details was required by one writer. Also the terms were considered somewhat confusing compared to the legacy environment. An example is the term 'element' in SGML environment, which was previously known as a 'component' is the Interleaf environment. Authoring itself was considered fairly easy when one first gets used to the view with full tags. However, one writer requested more training with authoring details.

The attitude towards the clarity and visual appeal of the SGML tools was quite neutral. They seemed to be suitable for their purpose of use. Documentation Workbench (DWB) was commented as being slightly confusing because of the many icons and windows. One writer commented that the Interleaf user-interface is quite depressing.

Restructuring the hierarchy of documents was considered hard and complicated to accomplish in the Adept editor. The visibility of index tags and entries was considered very confusing. Also the ability to collapse and expand particular element types in a document was required.

The manual version control of documents and figures was considered troublesome and confusing. The version number of the SGML metafile has to be changed in too many places. The authoring tool should have a capability of sizing figures instead of doing everything with drawing tools. With regards to a PDF document, the writers find it difficult to judge figure sizes on the screen. Also the VPS system for creating PDF files was considered complicated and time-consuming. However, one writer commented that the VPS system is a relief, because pagination and layout do not have to be worried about anymore.

Any phases of the SGML that help to ease the work process were not identified by very many of the writers at this point. One of the writers pointed that updating figures is simple because they are just linked to the document outside. Also Adept as a tool was found much quicker than Interleaf to work with because the navigation is fast, especially when some of the chapters are hidden.

Printing of documents was found troublesome by some writers, because Adept print-outs do not show page numbers and the final layout requires running the document through the VPS which may often be queued. VPS itself as an application, however, was not considered difficult to use. Making PDF files locally in the Interleaf environment was much quicker and reliable.

The user-friendliness of the SGML environment was not found very good at the moment, in the sense that SGML tools could be fully recommended to totally new users. The opinions may change when more practice and experiences of SGML authoring have been gained but some writers told that there are still too many shortcomings in the process. However, important benefits of using structured documentation, such as applicability to managing large document masses and necessity for online documentation, were understood. Interleaf was found more usable for paper-based documentation. Interleaf was also mentioned as having a more illustrative interface what comes to planning of document structure.

The possibility to have forced page breaks was required when compared SGML authoring with Interleaf authoring. Also the principle of WYSIWYG was said to be missing. Linking was considered much more complicated than with Interleaf. The link tool was considered visually messy and some kind of sorting should be implemented to it. When constructing links, the amount of spaces within the link element may be a little confusing, especially because it differs from the one needed with the index element. For example, list elements should have automatic attributes because they are almost always the same and should be used in a certain way forced by the layout rules. Some elements have too many attributes, of which only a few are supported. This situation makes it too easy to enter illegal attribute values. A strong comment was that SGML environment has been designed from the publishing point of view but for a writer it is far from an ideal tool.

A tool for producing graphics lists derived from the SGML document's figure and caption elements was desired for keeping track of figure use. The SGML migration project was thanked as being quite successful and no negative effects on authoring convenience of the first SGML-based documentation release was reported.

7.2.4 Evaluation of the Results

The training problems can mostly be solved with instructions and guidance of technical support persons. The emerged issues have to be gone through with the technical writers so that they know about possible solutions. Graphic declarations could be made easier by a software upgrade or by handling graphics some other way than using file entities. TIFF figures should also be used as little as possible because of the large file sizes and often complicated resolution attributes. Check completeness could also be run automatically by the authoring software when the document is saved, just to make sure it has been done and the most common mistakes can thus be avoided. Every writer should be familiar with the Nokia layout rules and their reasoning. In that way, many futile technical mistakes can be avoided.

Most of the process related problems are taken into consideration when the documentation processes, DTDs and tools are developed further. Beside getting a better understanding of the DIV concept, adding a new DIV-element in the middle of a structure should be made easier by having a more clear view of the document structure available in the authoring software. The error-proneness of naming documents and figures is a known problem besides version control and document management. For solving those problems, an adequate graphics database or DMS (document management system) system has been investigated. Also, the responsible interest group is informed with the VPS related problems. The confusion with the metafiles could be avoided also by letting document assistants take care of filling them out. However, the problem often is that the writers are willing to do everything concerning their documents by themselves, which requires them to be able to deal with technical issues as well. The meaning of VPS validation messages has to be explained thoroughly to the writers but the messages should also be more outstanding in the VPS service so that they would not be neglected so easily.

Tool problems are being investigated to find out what can be done to solve them inside PMR or Nokia Networks. Some of the shortcomings may be solved with software updates and some may be given as feedback for the software developers. Especially, the problems with the attribute selection menu and link management should be fixed by the software developers. The double text required by the index items is a DTD issue, which should probably be considered in the context of the DTD improvement work. The finding associated with the visual appearance of figures in the SGML editor depends on many issues, most importantly on the different resolution of a monitor screen compared to paper media and resolution attributes of the graphic element. The best way of avoiding these problems is to use consistently the same resolution with all the figures in the document, print proofs of the figures directly from the editor and use CGM figures in as many cases as possible.

The reported positive remarks show that writers have motivation to learn to work in the SGML environment and are highly trained for finding information on their own. Also some signs of higher level of understanding of the on-line documentation paradigm could be seen.

Closed questions revealed many interesting opinions, although very strong ones were not expressed in any question. The average answer as to finding the SGML process efficient was very slightly positive but the median was neutral. The reason for this kind of result is that many test users did not have very long experience of using the SGML environment so that they could have pointed out inefficiencies. Another reason may be that some phases are found more efficient and some others less efficient than the Interleaf-based process. Support for the latter reason is provided also by the median of four for the second statement: "The Interleaf-based process was faster and smoother than the SGML process". The SGML tools were considered quite easy to use, as the average was 3.6. The answers for the phase count of SGML process were controversial, since the standard deviation was 1.41. Still, the median was two, which indicates that less work phases could increase satisfaction. However, the SGML process was still not found very hard to use or easy to forget, since the average of that question was 2.4. A positive remark was that the test users were relatively happy with the instructions available.

Open questions revealed that a clear understanding and an extensive description of the practical steps of working in SGML environment are missing. This means that SGML-related guidelines have to be improved all the time. Another matter that emerged especially in the open questions was the need for fast paper printing with appropriate page breaks, page numbers and real-time layout. This need probably diminishes when hypermedia-type online documentation is becoming more commonly used by both customers and writers. At the moment writers are still too much tied with the paper-oriented publishing paradigm. User-friendliness of the tools is getting better all the time so that hopefully more and more writers feel it a convenient environment very soon. The navigation was already found quite easy. However, the fact is that SGML/XML authoring tools are not as user-friendly as traditional word-processing applications but the advantages are more in the later phases of the documentation life cycle. The ability of planning the structure of a document is an important feature of an authoring environment. Thus, the ways of improving that side of the environment are being investigated.

7.2.5 Evaluator's Remarks

The observation turned out to be a very fruitful method for finding out the shortcomings of the system and the early documentation production process steps. The same major problems kept coming up with every test user. As stated in the literature, the first two test users revealed most of the problems already. Most of the major problems were most likely revealed with the observation and questionnaire with the five tested users, as it was planned beforehand. All the problems that technical writers and tools support persons had been complaining about during the SGML migration came up also in the usability evaluation.

The biggest difficulty was that the evaluator himself has been working as an SGML support person with the particular writers and in some situations it was hard to find instructions for solving a problem without asking the evaluator. The temptation to help test users was, especially at the beginning of the evaluation sessions, also quite definite.

The test users, who had only two months' experience authoring documents in the SGML, commented afterwards that the observation and the questionnaire could have been conducted somewhat later so that they could have performed more efficiently and could have given more comprehensive answers to the questionnaire.

7.3 Sources of Error

The chosen test users were a representative sample of the writer population part of the study. They were the same as in the authoring measure. The observations encountered some interruptions and difficulties but so does a regular working day most often. Based on the comments of the writers and other experience, the majority of the possible authoring situations were covered but, of course, the test situations were not perfect.

The scope of the evaluation was ensured with a questionnaire. The questionnaire was problematic because users may answer questions negligently and interpret questions in different ways, although the questions were made as readable as possible. However, writers seemed to have taken the questions seriously and the answers were complete. The validity of questions was completely based on the judgement of the evaluator. The target was to compile a list of questions so that they would cover all the evaluation criteria.

8 CONCLUSIONS AND FUTURE WORK

As the total delivery time of the documentation process was measured in the legacy and new SGML environment, it was found shortened in the latter, mainly because of the more efficient and centralised postproduction. Earlier the postprocessing of a document library of a release was done completely locally but along the SGML migration a centralised postproduction system was affiliated. The SGML migration also resulted in a decreased number of process phases and a simplified local documentation process. Actually, the Interleaf delivery time should rather be compared with the local SGML delivery time, since only this factor affects the resources and time schedules of the business unit. The centralised publishing work is hard to influence at all from within a business unit, but seems like it still works quite efficiently. What comes to individual phases, it can be said that approving a document takes an unacceptably long period of time. However, this concerns both of the processes and the problem is actually not technical. The most problematic phase seems to be applying the paper layout to the SGML documents, which is time-consuming and its duration unforeseen to a writer.

Observation and open questions revealed many weaknesses in the tools and the documentation process, which need to be solved by developing the methods, processes and DTDs further. Some of the tool-related issues can be solved by the local technical support persons but some require software updates. Also some improvements, such as quick navigation within a document, were noticed in the SGML environment. The usability evaluation was conducted maybe in a little bit too early phase of the SGML migration. With some more experience, some of the test users could have avoided beginner's mistakes and would have analysed the open questions thoroughly. Thus, some of the emerged problems can be solved with more training and practice. Closed questions revealed that the SGML tools are found usable but practical matters in the documentation process are still quite difficult and some of them are not fully internalised. All in all, the SGML-based documentation process already satisfied most of the participant writers quite well as such. The chosen usability evaluation method, observation, was found a fruitful practice for testing a system. The emerged problems seem to correlate quite well with the preconceived idea and user comments in other departments. Still, many new important ideas came up.

On the basis of the experiences gained during this study, it can be said that hypermedia process models can and eventually should be applied to the technical online documentation development life cycle with small adjustments. The level of interactivity and multimedia in technical documentation will not reach the current level of interactivity and multimedia in hypermedia products very soon, in many cases they do not even fit into the purpose at all. It is also very much dependent on resources and competent creators of useful media elements, such as animations and hotspots. The custom of rapid prototyping will most probably become also a part of technical documentation processes, because of the navigational and structural planning, which are more complex tasks when compared to paper-oriented publishing. Object-oriented hypermedia documentation models seem to be too heavy and itemised for the case environment type of organisations but when they become more mature and more practical experiences are collected, the situation can change.

In the future, possible improvements such as XML based authoring environment, new multimedia technologies and documentation based on smaller document units would slightly change the documentation process. However, the transfer from SGML to XML would practically be more of a mechanical than conceptual change. Electronic on-line publishing enables completely new kind of documents that take advantage of modern audio, video and animation formats. Modularity would bring up new challenges to documentation designers when documents have to be designed as independent information units for reusability purposes. More mature and advanced XML authoring tools are expected to become available so that the user-friendliness of tools will increase and the prices hopefully will drop due to competition. Also the need for a professional DMS (Document Management System) will be constantly monitored.

If a new, just introduced project and resource management tool was taken into use, it would be a big help in work hour tracking, resource and workload management and project and program planning. Among other valuable reforms, also documentation project planning would be expected to become much more controllable compared to the current situation.

New terminals can be used independently of time and location and, therefore, will enable a completely new range of applications for documentation. These terminals, such as 3G mobile phones, PDAs (Personal Digital Assistant) or e-books, will utilise structural languages such as XML, HTML, XHTML and WML (Wireless Markup Language). Thus, SGML or XML-based documentation is an asset, which enables a quick and flexible adaptation to a variety of publishing media.

A question that rises along with the possible hypermedia process exploitation is how to the hypermedia-based customer documentation within a product development process changes the interaction between these two processes. Also, a better understanding and control of the early phases of documentation processes occurring before authoring should be reached. These two dilemmas require future work.

9 SUMMARY

Two of the most important objectives of starting to use SGML in technical documentation are the speeding-up and simplification of the documentation process. The purpose of this master's thesis was to study whether these objectives are actually met while migrating from word processing -based documentation to SGML-based documentation.

The first step before evaluating these two documentation processes was that the author familiarized himself with the case study environment. In practice, it meant that all the process phases and the persons involved in them had to be determined.

The delivery cycle of documents of the legacy and SGML processes were measured. Both processes were defined in the practical workflow level before the process measures were conducted. The delivery cycle of documents was measured in two parts: the authoring and postprocessing phases were separated from each others. The authoring measures were based on hour reports of technical writers and postprocessing measures on hour reports of documentation assistants and other related persons. In addition to this, authoring phase measures, which at both instances included five documents, included also five extra documents in the SGML-based process so that the learning issues could be taken into consideration.

The process measure results showed that there was no statistical difference between the authoring environments, although the SGML measures were slightly better on average. Some learning can also be seen between the two SGML results of authoring measures, although there was no statistical certainty either. The amount of new material written into documents was seen to be well related with the authoring time spent per page in the Interleaf environment, yet not significantly at all in the SGML environment. This probably indicates that SGML authoring is not so straightforward yet and tool issues occupy some of the authoring time. Time spent on background research seemed to correlate well with the amount of new material in a document in both environments. The total delivery time of the documentation process shortened, mainly because of the more efficient and centralised postproduction. It must be remembered that the Interleaf and SGML delivery cycles can not be compared to each other directly, since authoring and approving a document are such case-specific issues. As the total delivery cycle shortened in the SGML-based process, the number of local phases decreased and local delivery cycle shortened naturally as well.

The weaknesses of the new SGML-based process were identified for further development by conducting a usability study. The selection of the evaluation method was based on usability literature and existing knowledge on usability. The usability study was done using observation and a questionnaire. Observation and open questions revealed some advantages but also many weaknesses in the tools and the documentation process, which need to be solved in a way or another. Also quite many problematic areas emerged that can, fortunately, be solved with more training and practice. Closed questions revealed that the SGML tools are found usable but practical matters in the documentation process are still quite difficult and some of them are not fully internalised. All in all, the SGML-based documentation process already satisfied most of the participant writers quite well as such.

BIBLIOGRAPHY

/1/ **Albin, B.** *Process Constraints in the Management of Technical Documentation.* Proceedings of the 14th annual international conference on Marshaling new technological forces: building a corporate, academic, and user-oriented triangle. 19 – 22 October, 1996, Research Triangle, United States, pp. 67-74.

/2/ **Anon.** *Getting Started with SGML.* ArborText, Inc.,1995. Updated 19 January 2000 [referenced 6 June 2000]. Available at <URL: http://www.oasis-open.org/html/getstart.htm>.

/3/ **Boehm, B. W.** *A Spiral Model of Software Development and Enhancement.* IEEE Computer 21(1988)5, pp.61-72.

/4/ **Bryan, M., Li, M-S.** *OII Guide to Workflow Management and Collaborative Authoring.* European Commission's Open Information Interchange (OII) service. Updated 13 January 2000 [referenced 6 June 2000]. Available at <URL: http://guagua.echo.lu/oii/en/workflow.html>.

/5/ **Conrad, K.** *SGML: It's Not Just for Documents Anymore.* The Sagebrush Group, 1995, Revision of work published in the proceedings of SGML '94 and was presented at SGML '94, Vienna, Virginia, November 6-11, 1994. Available at <URL: http://www.sagebrushgroup.com/nondoc.htm>.

/6/ **Carlshamre, P., Tumminello, J. L.** *Technical Communicators' Current Views on Usability and Collaboration.* Proceedings of the 13th annual conference on Emerging from chaos: solutions for the growing complexity of our jobs, 1995, pp. 11 – 19.

/7/ **Elovainio, K.** *SGML-pohjainen dokumentointiprosessi.* Technical Research Centre of Finland (VTT). Espoo, Finland 1995. 129 p.

/8/ **Florac, W. A., Park, R. E., Carleton, A. D.** *Practical Software Measurement: Measuring for Process Management and Improvement.* CMU/SEI, 1997. 246 p.

/9/ **Garshol, L. M.** *Introduction to XML.* Updated 20 August 1999 [referenced 5 June 2000]. Available at <URL: http://www.stud.ifi.uio.no/~lmariusg/download/xml/xml_eng.html>.

/10/ **Garzotto, F., Paolini, P. Schwabe, D.** *HDM - a model based approach to hypermedia application design.* ACM Transactions on Information Systems, 11(1993)1, pp. 1-26.

/11/ **Goldfarb, C.** *The SGML Handbook.* Clarendon Press, Oxford, Great Britain 1990. 663 p.

/12/ **Goldie, P.** *Using SGML to Create Complex Interactive Documents for Electronic Publishing.* IEEE Transactions on Professional Communication, 40(1997)2, pp. 129–138.

/13/ **Grosso, P.** *Concepts Behind SGML-aware Authoring.* ArborText, Inc. Updated 19 January 2000 [referenced 5 June 2000]. Available at <URL: http://www.oasis-open.org/html/sgml-aware-authoring.html>.

/14/ **Hackos, J. T.** *Telecommunications Industry Benchmark Study - Final Report.* The Center for Information-Development Management, Denver, Colorado, USA 2000. 89 p.

/15/ **Heimbürger, A.** *Teknisen dokumentaation kehittäminen hypermedia- ja CD-ROM –tekniikan avulla.* Technical Research Centre of Finland (VTT). Espoo, Finland 1993. 104 p.

/16/ **Heimbürger, A.** *Applying SGML to publishing process.* Esbo Nordinfo Rapport 1995:1. 34 p.

/17/ **Herwijnen van, E.** *Practical SGML*. Kluwer Academic Publishers, Boston 1994. 288 p.

/18/ **Hollingsworth, D.** *Workflow Management Coalition: The Workflow Reference Model.* The Workflow Management Coalition Specification, Belgium 1994. Updated 29 November 1994 [referenced 2 June 2000]. Available at <URL: http://www.aiim.org/wfmc/standards/docs/rmv1-16.pdf>.

/19/ **Hom, J.** *The Usability Methods Toolbox.* Updated 10 June 1998 [referenced 26 June 2000]. Available at <URL: http://www.best.com/~jthom/usability/>.

/20/ **Hubbard, S.E.** *A practical approach to evaluating test results*. Professional Communication, IEEE Transactions on 32(1989)4, pp. 283 –288.

/21/ **Humphrey, W. S.** *A Process or a Plan?* Software Engineering Institute (SEI) 1999. Updated 1 June 2000 [referenced 2 June 2000]. Available at <URL: http://www.sei.cmu.edu/publications/articles/watts-humphrey/process-or-plan.html>.

/22/ **Ibrahim, R. L., Hirmanpour, I.** *The Subject Matter of Process Improvement: A Topic and Reference Source for Software Engineering Educators and Trainers.* CMU/SEI Technical Report, 1995. 160 p.

/23/ **Juntunen, T.** *Managing the Documentation Process in a Global Customer Training Organisation.* Department of Computer Science, Helsinki University of Technology, Finland 1995. 75 p.

/24/ **Kaminsky, S. K.** *Test Early, Test Often: A Formative Usability KH for Writers.* Proceedings of the 10th annual international conference on Systems documentation , 1992, pp. 47-55.

/25/ **Kanerva, A. Rojek, J.** *A data-collection strategy for usability tests.* Professional Communication, IEEE Transactions 37(1994)3, pp.149 –156.

/26/ **Kemppainen, P.** *SGML Implementation: Case Nokia Telecommunications.* Master's Thesis. Department of Computer Science, University of Tampere, Finland 1998. 104 p.

/27/ **Kontio, J.** *Process Management in Software Development*. Lecture slides of the course Tik-76.601 Introduction to Software Engineering at Helsinki University of Technology. Held at 13 October 1999. 47 p. Available at: <URL: http:// www.cs.hut.fi/Studies/Tik-76.601/1999/Slides/ProcessMgmt.PDF>.

/28/ **Korpela, J.** *Merkkauskielten idea.* Helsinki University of Technology, 2000. Updated 8 March 2000 [referenced 2 June 2000]. Available at: <URL: http:// www.hut.fi/~jkorpela/html/merkkaus.html>.

/29/ **Kuronen, T.** *Hajautettu dokumenttien hallinta - Johdatus tekstin ja dokumenttien käsittelyyn tietoverkoissa.* Oulun Yliopisto, Oulu 1999. 71 p.

/30/ **Laakso, P.** *Käyttäjäkeskeiset suunnittelumenetelmät SGML-kirjoitusympäristön suunnittelussa.* Master's Thesis. Department of Computer Science, University of Helsinki, Finland 2000. 74 p.

/31/ **Laakso, S. A., Laakso, K.-P.** *Käyttöliittymien suunnittelu ja arviointi ohjelmistoprojekteissa.* Interacta Design Oy. Presented at the Global Documentation Meeting, 8 June 2000, Helsinki, Finland.

/32/ **Laru, M.** *Measurement of Business Processes. Customer Documentation within Software Development Process as Case.* Faculty of Mechanical Engineering, Helsinki University of Technology, Finland 1996. 97 p.

/33/ **Lowe, D., Hall, W.** *Hypermedia and the Web – An Engineering Approach.* John Wiley & Sons Ltd., 1999. 626 p.

/34/ **Lowe, D., Webby, R.** *Utilisation of process modeling in improving the hypermedia development process.* The New Review of Hypermedia and Multimedia 5(1999)1, p. 133-150.

/35/ **Lowe, D., Webby, R.** *Web Development Process Modelling and Project Scoping: Work in Progress.* In: Proceedings of the First International Workshop on Web Engineering (WebE 98). Brisbane, Australia, 14 April 1998. Updated 28 May 1999 [referenced 9 June 2000]. Available at <URL: http:// btwebsh.macarthur.uws.edu.au/san/webe98/Proceedings/Lowe/WebE98-LoweWebby.htm>.

/36/ **Lynch, R., Cross, K.** *Measure Up! Yardsticks for Continuous Improvement.* Blackwell Business, Cambridge, MA, 1995. 250 p.

/37/ **Lyytikäinen, V.** *Rakenteisuuden hyödyntäminen elektronisissa dokumenteissa: SGML-pohjaisen dokumentaation tutkimus ja käyttö Suomessa 1997.* Updated 17 April 1998 [referenced 12 June 2000]. Available at: <URL: http://www.cs.jyu.fi/ ~raske/kartoitus97.html>.

/38/ **Mehlenbacher, B.** *Software Usability: Choosing Appropriate Methods for Evaluating Online Systems and Documentation.* Proceedings of the 11th annual international conference on Systems documentation , 1993, pp. 209 – 222.

/39/ **Mendes, M. E. X., Hall, W., Harrison, R.** *Applying measurement principles to improve hypermedia authoring.* The New Review of Hypermedia and Multimedia 15(1999)1, p. 105-132.

/40/ **Nielsen J., Landauer, T. K.** *A mathematical model of the finding of usability problems.* Proceedings of ACM INTERCHI'93 Conference. Amsterdam, The Netherlands, 24-29 April 1993. pp. 206-213.

/41/ **Nielsen, J.** *Usability Engineering.* Academic Press, Inc., San Diego, CA, USA 1993. 358 p.

/42/ **Nielsen, J.** *Users First: Usability Metrics - How Good Are You?* Column on ZDNet, August 1999. Updated 25 August 2000 [referenced 26 June 2000]. Available at <URL: http://www.zdnet.com/devhead/stories/articles/ 0,4413,2321306,00.html>.

/43/ **Nielsen, J.** *Why You Only Need to Test With 5 Users.* Jakob Nielsen's Alertbox, March 19, 2000. Updated 3 April 2000 [referenced 26 June 2000]. Available at <URL: http://www.useit.com/alertbox/20000319.html>.

/44/ **Olsina, L.** *Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy.* In: Proceedings of the First International Workshop on Hypermedia Development, Processes, Methods and Models. Pittsburgh, PA, USA, 20-24 June, 1998. Updated 12 November 1998 [referenced 12 June 2000]. Available at <URL: http://www.eng.uts.edu.au/~dbl/HypDev/ ht98w/Olsina/AISWork_HT98.html>.

/45/ **Olsina, L.** *Functional View of the Hypermedia Process Model.* Faculty of Engineering, National University of La Pampa Engineering School, Argentina 1998. Updated 12 November 1998 [referenced 12 June 2000]. Available at <URL: http://gidis.ing.unlpam.edu.ar/downloads/htf5hfpm.html>.

/46/ **Preece, J.** *Human-Computer Interaction.* Wokingham, Addison-Wesley, 1994. 775 p.

/47/ **Price, J.** *Structuring Complex Interactive Information.* IEEE Transactions on Professional Communication, July 1997 pp. 69-77. Available at <URL: http:// www.theprices.com/4artTW7.htm>.

/48/ **Riepponen, T.** *Production of technical documentation – the case of NTC.* InSGML project, Department of Computer Science and Information Systems, University of Jyväskylä. Jyväskylä, Finland 1999. 133 p.

/49/ **Rinta-Piirto, M.** *Teknisen asiakasdokumentaation esittäminen online-dokumentaationa.* Master's Thesis. Laboratory of Media Technology, Helsinki University of Technology. Espoo, Finland 1999. 170 p.

/50/ **Rosenbaum, S.** *Usability evaluations versus usability testing: when and why?* Professional Communication, IEEE Transactions 32(1989)4, pp. 210 –216.

/51/ **Rozum, J. A.** *Defining and Understanding Software Measurement Data.* Carnegie Mellon University, Pittsburgh, PA, USA 1999. Available at <URL: http:// www.sei.cmu.edu/pub/documents/articles/pdf/using.data.asm.pdf>.

/52/ **Ryökkynen, S.** *Verkkolomakepalvelu ja sen evaluointi.* Master's Thesis. Laboratory of Media Technology, Helsinki University of Technology. Espoo, Finland 1998. 101 p.

/53 **Saarela, J.** *The role of metadata in electronic publishing.* Doctoral thesis. Finnish Academy of Technology, Espoo 1999. 614 p.

/54/ **Schöch, V.** *HCI Space: Definition of Usability (ISO 9241).* Updated 9 June 1999 [referenced 10 July 2000]. Available at <URL: http://www.tau-web.de/hci/space/ i7.html>.

/55/ **Smith, J. M.** *SGML and Related Standards – Document Description and Processing Languages.* Ellis Horwood Limited, England 1992. 151 p.

/56/ **Sommerville, I.** *Software Engineering.* 5th edition. Addison-Wesley Pub Co, 1995. 742 p.

/57/ **Tanskanen, J.** *Guide to writing in SGML environment (Neutral DTD for Adept*Editor).* Issue 2-0. Nokia Networks, 2000. 72 p.

/58/ **Theng, Y. L., Marsden, G.** *Authoring tools: Towards continuous usability testing of web documents.* Proceedings of the HT98 Workshop on Hypermedia Development Processes, Methods and Models. Pittsburgh, PA, USA, 20-24 June, 1998. Updated 12 November 1998 [referenced 4 July 2000]. Available at <URL: http://www.eng.uts.edu.au/~dbl/HypDev/ht98w/YinLeng/HT98_YinLeng.html>.

/59/ **Theng, Y.L., Thimbleby, H. and Jones, M.** *Designer tools for hypertext authoring.* IEE Colloquium on "The Authoring and Application of Hypermedia-Based User Interfaces," Digest No: 95/202, U.K. Available at <URL: http:// www.cs.mdx.ac.uk/staffpages/yinleng/ieetools.html>.

/60/ **Thimbleby, H.** *Gentler: a tool for systematic web authoring.* Int. J. Human Computer Studies 13(1997)47, pp.139-168. Academic Press Limited, 1997. Available at <URL: http://ijhcs.open.ac.uk/thimbleby/thimbleby.pdf>.

/61/ **Travis B., Waldt D.** *The SGML Implementation Guide - A Blueprint for SGML Migration.* Springer-Verlag, Berlin, Germany 1995. 522 p.

/62/ **Treviranus, J., McCathie-Nevile, C., Jacobs, I., Richards, J.** *Authoring Tool Accessibility Guidelines 1.0.* W3C Recommendation 3 February 2000. Updated 24 February 2000 [referenced 27 June 2000]. Available at <URL: http:// www.w3.org/TR/2000/REC-ATAG10-20000203//>.

/63/ **Usdin, T., Graham, T.** *XML: Not a Silver Bullet, But a Great Pipe Wrench.* StandardView 6(1998)3, pp.125-132.

/64/ **Venna, J.** *Suuryrityksen Ohjeiston Hallintajärjestelmä Case: Nokia Telecommunications/Switching Platforms.* Master's Thesis. Helsinki School of Economics and Business Administration, Finland 1997. 104 p.

/65/ **Walker J. H.** *Review: Authoring Tools for Complex Document Sets.* The society of Text 1997, pp. 132-147. Available at <URL: http://www.scism.sbu.ac.uk/ inmandw/review/hypermedia/review/rev18424.html>.

/66/ **Watson, D. G.** *Brief History of Document Markup.* Updated 20 November 1992 [referenced 6 June 2000]. Available at <URL: http://edis.ifas.ufl.edu/ BODY_AE038>.

Appendix A. Writing Efficiency Measure Sheet

Hour measures of writing a document

Writer's name	**Interleaf or SGML (1,2)**	**Doc code**
Elton Example	*SGML 1*	*DGDA5007-02*

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* **Total time of writing a document for the release (h)**	**1) Writing at the computer (h)**	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material* [1] *(%)*	**2) Background research and planning (h)**	**3)Tools and management** [2]**(h)**
1	0	0	0	0	0	0	0
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						
8	0						
9	0						
10	0						
11	0						
12	0						
13	0						
14	0						
15	0						
16	0						
Σ	0	0				0	0

Appendix B. Efficiency Measure Sheets for Postprocessing a Document

Hour measures of postprocessing a document

Assistant's name	Interleaf or SGML	Doc code	Doc length (pages)
Elton Example	*Interleaf*	*DGDA5007-02*	*123*

Doc #	*Don't fill this column* Total time of postprocessing a document for the release (h)	1) Approving the document (h)	2) Moving to the archive (h)	3) Finalizing layout and making a pdf (h)	4) Making a document list (h)	5) Paper documents for the production line (h)
1	0	*0*	*0*	*0*	*0*	*0*
2	0					
3	0					
4	0					
5	0					
Σ	0	0	0	0	0	0

Hour measures of postprocessing a document

Assistant's name	Interleaf or SGML	Doc code	Doc length (pages)
Elton Example	*SGML*	*dn00126484*	*123*

Doc #	*Don't fill this column* Total time of postprocessing a document for the release (h)	1) Applying the document layout (h)	2) Approving the document (h)	3) Moving to the local archive (h)
1	0	0	0	0
2	0			
3	0			
4	0			
5	0			
Σ	0	0	0	0

Appendix C. Efficiency Measure Sheets for Postprocessing a Document Set

Hour measures of postprocessing a document #2

Assistant's name	Interleaf or SGML	Number of docs
Elton Example	*Interleaf*	*123*

[1] Starting from letting CDP to know we have our electronic book ready, lasting until it's available to the customer

* These values are measured only once since they are done for all the documents together

Release	*Don't fill this column* Total time of postprocessing a document set for the release (h)	6) Creating electronic book (h) *	7) Approving electronic book (h) *	8) CDP production[1] (h) *
2.1	*0*	*0*	*0*	*0*

Hour measures of postprocessing a document #2

Assistant's name	Interleaf or SGML	Number of docs
xxxxxxxxxxxxxx	*SGML*	*123*

Release	*Don't fill this column* Total time of postprocessing a document for the release (h)	4) Freezing the SISU (h)	5) CDP production (h)	6) Checking and approving the document library and sales item (h)	7) CDP mass production (h)
2.1	*0*	*0*	*0*	*0*	*0*

Appendix D. Results of Measuring Writing Efficiency

Interleaf environment:

Hour measures of writing a document

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	Interleaf	DEIA5063	49

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	Don't fill this column Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	52	35	12	15	8	12	5
Total	52	35				12	5

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	Interleaf	DGDA 5002-08e	54

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	Don't fill this column Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	460	140	70	5	25	300	20
2	33	18	20	40	40	10	5
3	7	3			100		4
4	2	1			100		1
Total	502	162				310	30

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	Interleaf	DEIF 5052-02e	122

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	Don't fill this column Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	1					1	
2	5	4			100	1	
3	2					2	
4	1						1
5	2					2	
6	3	3	25		75		
7	2	2			100		
8	1	1			100		
Total	17	10				6	1

Appendix D. Results of Measuring Writing Efficiency

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*Interleaf*	*DNDB5046-02e*	*200*

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	84	65.5	18	49	33	3.5	15
2	2						2
3	3						3
4	2.5	2.5			100		
5	6	3			100		3
6	9	7			100		2
7	5	5			100		
8	19	15			100		4
Total	130.5	98				3.5	29

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*Interleaf*	*DGIB 5015-04e*	*285*

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	6.75	5	75	15	10	1.5	0.25
2	7	1	100	0	0	6	
3	7.25	1	100	0	0	6	0.25
4	7	1	100	0	0	6	
5	7	1	100	0	0	6	
6	6	1	100	0	0	5	
7	7	1	100	0	0	6	
8	7	1	25	25	50	6	
9	7	7	25	25	50	0	
10	7	7	25	25	50	0	
11	7	7	25	50	25	0	
12	6	5	0	50	50	1	
13	8	7	0	50	50	1	
14	8	7	0	50	50	1	
15	7.25	7	0	50	50	0	0.25
16	7	7	50	25	25	0	
17	9	9	0	50	50	0	
18	9.25	9	0	50	50	0	0.25
19	8.25	8	0	0	100	0	0.25
Total	138.75	92				45.5	1.25

SGML (1) environment:

Hour measures of writing a document

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*SGML 1*	*DN00126418*	*58*

[1)] Correcting, replacing, removing material or renovating the structure of a document
[2)] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1)] (%)*	2) Background research and planning (h)	3)Tools and management [2)](h)
1	460	140	70	5	25	300	20
2	14	8	37.5	12.5	50	3	3
3	4	2		100			2
4	2						2
5	8	8	100				
6	8	7		86	14		1
7	1.5	1			100		0.5
Total	497.5	166				303	28.5

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*SGML 1*	*DN00126566*	*110*

[1)] Correcting, replacing, removing material or renovating the structure of a document
[2)] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1)] (%)*	2) Background research and planning (h)	3)Tools and management [2)](h)
1	61	34	21	50	29	10	17
Total	61	34				10	17

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*SGML 1*	*DN00126593*	*288*

[1)] Correcting, replacing, removing material or renovating the structure of a document
[2)] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1)] (%)*	2) Background research and planning (h)	3)Tools and management [2)](h)
1	7	1			100	5	1
2	6.5	3.5	25	30	45		3
3	7	5	60	10	30		2
4	3.5	3	100				0.5
5	2.5	2	100				0.5
6	6	6	100				
7	6	6			100		
8	7	6	100				1
9	1	1			100		0.5
Total	46.5	33.5				5	8

Hour measures of writing a document

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*SGML 1*	*DN00267266*	*34*

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* **Total time of writing a document for the release (h)**	**1) Writing at the computer (h)**	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	**2) Background research and planning (h)**	**3)Tools and management [2](h)**
1	8	4	60	40		2	2
2	6	2	60	40		3	1
3	8	4	50	50			4
4	6.5	3	40	60		2	1.5
5	7.5	3	65	35		1.5	3
6	7.5	5	60	40		1.5	1
7	5	2	40	60		1	2
8	5.5	4		80	20		1.5
9	7.5						7.5
10	3	3		50	50		
11	7	4	40	40	20		3
12	7	5		30	70		2
13	7	6			100		1
14	3	2			100		1
15	4	3		50	50		1
Total	92.5	50				11	31.5

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*SGML 1*	*dn00126581*	*170*

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* **Total time of writing a document for the release (h)**	**1) Writing at the computer (h)**	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	**2) Background research and planning (h)**	**3)Tools and management [2](h)**
1	2						2
2	3	2	100			1	
3	2.5	2.5	100				
4	1					1	
5	4	4	50		50		
6	3.5	3.5			100		
7	4	3	50	25	25		1
8	3	2			100		1
9	3.5	3			100		0.5
Total	26.5	20				2	4.5

SGML (2) environment:

Hour measures of writing a document

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*SGML 2*	*DN00252569*	*42*

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	7	2	50	50		3	2
2	7	2	100			5	
3	5	1	100			3	1
4	6	1	100			3	2
5	6	2	70	30		2	2
6	3	1	100			1	1
7	3	1	100			2	
8	2					2	
9	7	5	100				2
10	5	5	100				
11	2	2			100		
12	3	3	100				
13	5	2	100				3
14	6	3	100				3
15	4	1					3
16	5	4			100		1
17	2	1			100		1
Total	78	36				21	21

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	*SGML 2*	*DN00126515*	*48*

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1. Writing next to computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2. Background research and planning (h)	3.Tools and management [2](h)
1	7					6	1
2	6.5	6	30	40	30		0.5
3	5.5	4	100			1	0.5
4	2	1			100	0.5	0.5
5	5	5			100		
6	4.5	4			100		0.5
7	1.5	1			100		0.5
Total	32	21				7.5	3.5

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	SGML 2	DN00256986	30

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	3	2		25	75		1
2	4	4		100			
3	7.5					7.5	
4	4	4	25	75			
5	7.5	7.5	40	60			
6	4.5	4.5		100			
7	4.5	3.5		50	50		1
8	7	3			100	4	
9	7	3			100	4	
10	4.5	4.5			100		
11	4	4			100		
12	2	2			100		
13	1	1			100		
Total	60.5	43				15.5	2

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	SGML 2	DN00126721	156

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	1					1	
2	2					2	
3	3	2			100	1	
4	2.5	2			100		0.5
5	3.5	3	50	50			0.5
6	2	2			100		
7	3.5	3			100		0.5
Total	17.5	12				4	1.5

Writer's name	Interleaf or SGML (1,2)	Doc code	Doc length (pages)
xxxxxxxxxxx	SGML 2	DN00132306	238

[1] Correcting, replacing, removing material or renovating the structure of a document
[2] Everything that has to do with managing documents or figures plus tool problems

Day	*Don't fill this column* Total time of writing a document for the release (h)	1) Writing at the computer (h)	*1.1 New material (%)*	*1.2 Copy-pasting new ready-made material (%)*	*1.3 Updating material [1] (%)*	2) Background research and planning (h)	3)Tools and management [2](h)
1	9	2			100		7
2	2						2
3	5						5
Total	16	2				0	14

Appendix E. Results of Measuring Postprocessing Efficiency

Interleaf environment:

Hour measures of postprocessing a document

Assistant's name	Interleaf or SGML	Doc code	Doc length (pages)
xxxxxxxxxxxxxx	*Interleaf*	*DGDA5007-02*	*123*
	Interleaf	*CAG 80528-03e*	*43*
	Interleaf	*DNOA 5021-04e*	*184*
	Interleaf	*DNDB 5024-03e*	*90*
	Interleaf	*DGIB 5015-05e*	*285*

0.083=5min
0.16=10min

Doc #	*Don't fill this column* Total time of postprocessing a document for the release (h)	1) Approving the document (h)	2) Moving to the archive (h)	3) Finalizing layout and making a pdf (h)	4) Making a document list (h)	5) Paper documents for the production line (h)
DGDA5007-02	*72.343*	*72.1*	*0.083*	*0.16*	*0.083*	*0.16*
CAG 83528-03e	*2.903*	*2*	*0.16*	*0.5*	*0.083*	*0.16*
DNOA5021-04e	*24.16*	*23*	*0.16*	*1*	*0.083*	*0.16*
DNDB 5024-03e	*48.193*	*47.95*	*0.083*	*0.16*	*0.083*	*0.16*
DGIB 5015-05e	*3.963*	*3.72*	*0.083*	*0.16*	*0.083*	*0.16*
Total	*151.562*	*148.77*	*0.569*	*1.98*	*0.415*	*0.8*

Hour measures of postprocessing a document #2

Assistant's name	Interleaf or SGML	Number of docs
xxxxxxxxxxxxxx	*Interleaf*	*45*
	SGML (PMR)	*20*
	SGML (SWP)	*84*

1) Starting from letting CDP to know we have our electronic book ready, lasting until it's available to the customer

2) 2 weeks max., sometimes only 2 days

*) These values are measured only once since they are done for all the documents together

Release	*Don't fill this column* Total time of postprocessing a document set for the release (h)	6) Creating electronic book (h) *	7) Approving electronic book (h) *	8) CDP production[1] (h) *
2.1-0	*223*	*145*	*3*	*75*[2]

SGML environment:

Hour measures of postprocessing a document

Assistant's name	Interleaf or SGML	Doc code	Doc length (pages)
xxxxxxxxxxx	SGML	dn00126484	48
	SGML	dn00126515	48
	SGML	dn00126593	288
	SGML	dn00256986	30
	SGML	dn00126878	32

Doc #	Don't fill this column Total time of postprocessing a document for the release (h)	1) Applying the document layout (h)	2) Approving the document (h)	3) Moving to the local archive (h)
1	265.79	2.5	263.13	0.16
2	21.28	0.75	20.37	0.16
3	24.96	0.5	24.3	0.16
4	48.81	0.33	48.32	0.16
5	72.8966	0.12	72.62	0.16
Total	433.7366	4.1966	428.74	0.8

Hour measures of postprocessing a document #2

Assistant's name	Interleaf or SGML	Number of docs
xxxxxxxxxxxxxxx	SGML	221

Release	Don't fill this column Total time of postprocessing a document for the release (h)	4) Freezing the SISU (h)	5) CDP production (h)	6) Checking and approving the document library and sales item (h)	7) CDP mass production (h)
2.1	202.5	15	90	22.5	75

Appendix F. Calculations Based on the Efficiency Measures

Authoring Interleaf documents:

Document 1:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		9.296296296		
Writing at the computer	32.27091633	3		
New material			101.6	62.71604938
Copy-pasting new ready-made material			14.2	8.765432099
Updating material			46.2	28.51851852
Background research and planning	61.75298805	5.740740741		
Tools and management	5.976095618	0.555555556		

Document 2:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.139344262		
Writing at the computer	58.82352941	0.081967213		
New material			0.75	7.5
Copy-pasting new ready-made material			0	0
Updating material			9.25	92.5
Background research and planning	35.29411765	0.049180328		
Tools and management	5.882352941	0.008196721		

Document 3:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.6525		
Writing at the computer	75.09578544	0.49		
New material			11.79	12.03061224
Copy-pasting new ready-made material			32.095	32.75
Updating material			54.115	55.21938776
Background research and planning	2.681992337	0.0175		
Tools and management	22.22222222	0.145		

Document 4:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.486842105		
Writing at the computer	66.30630631	0.322807018		
New material			18.75	20.38043478
Copy-pasting new ready-made material			31.75	34.51086957
Updating material			41.5	45.10869565
Background research and planning	32.79279279	0.159649123		
Tools and management	0.900900901	0.004385965		

Document 5:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		1.306122449		
Writing at the computer	73.4375	0.959183673		
New material			5.64	16.11428571
Copy-pasting new ready-made material			7.05	20.14285714
Updating material			3.76	10.74285714
Background research and planning	18.75	0.244897959		
Tools and management	7.8125	0.102040816		

	Hours of authoring
	52
	502
	17
	138.75
	130.5
$\bar{x}$	168.05

Postprocessing Interleaf documents:

	Average (h)	% of total average	Average without approval (h)	% without approval	Standard deviation (h)
Total time of postprocessing a document for the release	30.3124	100	0.5584		
Approving the document	29.754	98.15784959			30.07469169
Moving to the archive	0.1138	0.375423919		15.11689692	
Finalizing layout and making a pdf	0.396	1.306396062		52.60361318	
Making a document list	0.083	0.273815336		11.02550478	
Paper documents for the production line	0.16	0.527836793		21.25398512	

Doc #	h/Page
1	0.588154472
2	0.067511628
3	0.131304348
4	0.535477778
5	0.013905263
Total	1.336353488
Average	0.267270698

	% of total average
Freezing the SISU	7.407407407
CDP Production	44.44444444
Checking and approving the document library and sales item	11.11111111
CDP mass production	37.03703704

Total authoring time	840,25
Average	168,05
Standard deviation	193,70
Total postprocessing time (part #1)	151,56
Average	30,31
Standard deviation	29,87
Total postprocessing time (part #2)	223,0
Average (all)	1,50
Total average	199,86

Authoring SGML (1) documents:

Document 1:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		8.577586207		
Writing at the computer	33.36683417	2.862068966		
New material			109	65.6626506
Copy-pasting new ready-made material			16.02	9.65060241
Updating material			40.98	24.68674699
Background research and planning	60.90452261	5.224137931		
Tools and management	5.728643216	0.49137931		

Document 2:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.554545455		
Writing at the computer	55.73770492	0.309090909		
New material			7.14	21
Copy-pasting new ready-made material			17	50
Updating material			9.86	29
Background research and planning	16.39344262	0.090909091		
Tools and management	15.45454545	0.154545455		

Document 3:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.161458333		
Writing at the computer	72.04301075	0.116319444		
New material			20.875	62.31343284
Copy-pasting new ready-made material			1.55	4.626865672
Updating material			11.075	33.05970149
Background research and planning	10.75268817	0.017361111		
Tools and management	17.20430108	0.027777778		

Document 4:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		2.720588235		
Writing at the computer	54.05405405	1.470588235		
New material			14.15	28.3
Copy-pasting new ready-made material			19.75	39.5
Updating material			16.1	32.2
Background research and planning	11.89189189	0.323529412		
Tools and management	34.05405405	0.926470588		

Document 5:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.155882353		
Writing at the computer	75.47169811	0.117647059		
New material			8	40
Copy-pasting new ready-made material			0.75	3.75
Updating material			11.25	56.25
Background research and planning	7.547169811	0.011764706		
Tools and management	16.98113208	0.026470588		

Authoring SGML (2) documents:

Document 1:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		1.857142857		
Writing at the computer	46.15384615	0.857142857		
New material			27.40	76.11111111
Copy-pasting new ready-made material			1.6	4.444444444
Updating material			7	19.44444444
Background research and planning	26.92307692	0.5		
Tools and management	26.92307692	0.5		

Document 2:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.666666667		
Writing at the computer	65.625	0.4375		
New material			5.8	27.61904762
Copy-pasting new ready-made material			2.4	11.42857143
Updating material			12.8	60.95238095
Background research and planning	23.4375	0.15625		
Tools and management	10.9375	0.072916667		

Document 3:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		2.016666667		
Writing at the computer	71.07438017	1.433333333		
New material			4	9.302325581
Copy-pasting new ready-made material			18.25	42.44186047
Updating material			20.75	48.25581395
Background research and planning	25.61983471	0.516666667		
Tools and management	3.305785124	0.066666667		

Document 4:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.112179487		
Writing at the computer	68.57142857	0.076923077		
New material			1.5	12.5
Copy-pasting new ready-made material			1.5	12.5
Updating material			9	75
Background research and planning	22.85714286	0.025641026		
Tools and management	8.571428571	0.009615385		

Document 5:

	%of total	h/page	h total	% of writing
Total time of writing a document for the release		0.067226891		
Writing at the computer	12.5	0.008403361		
New material			0	0
Copy-pasting new ready-made material			0	0
Updating material			2	100
Background research and planning	0	0		
Tools and management	87.5	0.058823529		

Hours of authoring
78
32
60.5
17.5
16

$\bar{x}$ 40.8

Postprocessing SGML documents:

	Average (h)	% of total average	Average without approval (h)	% without approval	Standard deviation (h)
Total time of postprocessing a document for the release	86.74732	100	0.99932		
Applying the document layout	0.83932	0.967545741		83.9891126	
Approving the document	85.748	98.84801052			101.3505218
Moving to the archive	0.16	0.184443738		16.0108874	

Doc #	h/Page
1	0.857142857
2	0.4375
3	1.433333333
4	0.076923077
5	0.008403361
Total	2.813302629
Average	0.562660526

	% of total average
Freezing the SISU	6.896551724
CDP Production	48.27586207
Checking and approving the document library and sales item	10.34482759
CDP mass production	34.48275862

Total authoring time	204
Average	40,8
Standard deviation	27,4
Total postprocessing time (part #1)	433,74
Average	86,75
Standard deviation	102,21
Total postprocessing time (part #2)	202,5
Average (all)	0,92
Total average	128,5

Regression analysis, authoring a document (Interleaf):

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0,98152212
R Square	0,963385673
Adjusted R Square	0,951180897
Standard Error	0,859843228
Observations	5

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	58,35914612	58,35914612	78,9351391	0,003006801
Residual	3	2,21799113	0,739330377		
Total	4	60,57713725			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 99,0%*	*Upper 99,0%*
Intercept	-1,691241663	0,597878814	-2,828736565	0,066258686	-3,593960671	0,21147735	-5,18336078	1,800877454
X Variable 1	0,171274016	0,019277748	8,884544957	0,003006801	0,109923562	0,23262447	0,058675627	0,283872406

Regression analysis, authoring a document (SGML 1):

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0,485118155
R Square	0,235339624
Adjusted R Square	-0,019547168
Standard Error	3,630538193
Observations	5

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	12,16997661	12,16997661	0,923310393	0,407492134
Residual	3	39,54242271	13,18080757		
Total	4	51,71239933			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 99,0%*	*Upper 99,0%*
Intercept	-1,362221858	4,271364596	-0,318919593	0,770702759	-14,95562309	12,23117937	-26,31061208	23,58616837
X Variable 1	0,087359683	0,090915345	0,960890416	0,407492134	-0,201973791	0,376693158	-0,443663002	0,618382369

Regression analysis, authoring a document (SGML 2):

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0,595183047
R Square	0,35424286
Adjusted R Square	0,138990479
Standard Error	0,870701941
Observations	5

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	1,247648269	1,247648269	1,645709373	0,28967452
Residual	3	2,274365612	0,758121871		
Total	4	3,522013881			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 99,0%*	*Upper 99,0%*
Intercept	0,54368619	0,498987042	1,089579777	0,355594022	-1,044314768	2,131687147	-2,37082114	3,458193522
X Variable 1	0,017561629	0,01368952	1,282852046	0,28967452	-0,026004574	0,061127831	-0,06239677	0,097520029

Regression analysis, background research of a document (Interleaf):

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0,981405404
R Square	0,963156567
Adjusted R Square	0,950875423
Standard Error	4,942906545
Observations	5

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	1916,120666	1916,120666	78,425637	0,003035281
Residual	3	73,29697534	24,43232511		
Total	4	1989,417641			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 99,0%*	*Upper 99,0%*
Intercept	12,94184383	2,52497643	5,125530551	0,0143789	4,906234384	20,97745328	-1,806159028	27,68984669
X Variable 1	8,698074692	0,982186826	8,855825045	0,0030353	5,572314924	11,82383446	2,961270995	14,43487839

Regression analysis, background research of a document (SGML 1):

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0,970002597
R Square	0,940905039
Adjusted R Square	0,921206718
Standard Error	0,642867866
Observations	5

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	19,74058564	19,7405856	47,7657494	0,006208626
Residual	3	1,239837281	0,41327909		
Total	4	20,98042292			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 99,0%*	*Upper 99,0%*
Intercept	-0,370312624	0,360559301	-1,0270505	0,3799675	-1,517774317	0,777149069	-2,476284603	1,735659355
X Variable 1	0,617849461	0,089397294	6,91127698	0,00620863	0,333347106	0,902351816	0,095693478	1,140005444

Regression analysis, background research of a document (SGML 2):

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0,99894748
R Square	0,99789607
Adjusted R Square	0,99719476
Standard Error	0,01336521
Observations	5

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	0,254171974	0,25417197	1422,9048	4,09835E-05
Residual	3	0,000535887	0,00017863		
Total	4	0,254707861			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 99,0%*	*Upper 99,0%*
Intercept	-0,01387692	0,008995556	-1,542642	0,220594	-0,042504824	0,014750979	-0,0664186	0,038664751
X Variable 1	0,26863853	0,007121646	37,7214108	4,098E-05	0,245974251	0,291302805	0,227042079	0,310234977

Comparing Interleaf vs. SGML authoring:

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Interleaf	5	4,853958	0,970792	1,389719
SGML	5	2,813303	0,562661	0,3509

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0,416427	1	0,416427	0,478482	0,508684	5,317645
Within Groups	6,962473	8	0,870309			
Total	7,3789	9				

Comparing the two SGML authoring processes:

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
SGML 1	5	4,875715	0,975143	1,430615
SGML 2	5	2,813303	0,562661	0,3509

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0,425354	1	0,425354	0,47752	0,509099	5,317645
Within Groups	7,126056	8	0,890757			
Total	7,55141	9				

Appendix G. Usability Evaluation Sheets

USABILITY EVALUATION

Please circle the best alternative. The meaning of the alternatives is explained below.

5	4	3	2	1	N/o
Strongly agree	Slightly agree	Neutral	Slightly disagree	Strongly disagree	No opinion

1. The SGML process is efficient and eases routine tasks	**5**	**4**	**3**	**2**	**1**	**N/o**
2. The SGML process is faster and smoother than the Interleaf process	**5**	**4**	**3**	**2**	**1**	**N/o**
3. The SGML tools are hard to use	**5**	**4**	**3**	**2**	**1**	**N/o**
4. The SGML process has few enough phases to be usable	**5**	**4**	**3**	**2**	**1**	**N/o**
5. I have personal experience that the SGML tools include fatal bugs and features	**5**	**4**	**3**	**2**	**1**	**N/o**
6. The instructions for the SGML environment are sufficient	**5**	**4**	**3**	**2**	**1**	**N/o**
7. SGML tools are complicated and the interfaces are confusing and visually poor	**5**	**4**	**3**	**2**	**1**	**N/o**
8. The SGML process is hard to use and it is easy to forget	**5**	**4**	**3**	**2**	**1**	**N/o**

Please express your opinions as detailed as possible.

9. Are the instructions for producing SGML documents sufficient? Are the terms used in the system familiar for the users?

10. Is authoring documents easy enough?

11. Are the interfaces of different SGML related tools clear and visually appealing? Are the icons and figures used in the applications appropriate?

12. Which functions are necessary and should be included in the SGML related tools?

13. Are there any phases in the SGML process that are troublesome?

14. Are any phases of the SGML process helping to ease the work process substantially?

15. Is the printing of documents troublesome?

____

____

16. Are you honestly able to recommend the SGML environment to other technical writers?

____

____

____

____

17. Is there something essential that you feel that the SGML environment is missing compared to the Interleaf environment or something else?

____

____

____

18. Any suggestions and comments?

____

____

____

____

____

____

____

____

____

____