

Consortium for studying, evaluating, and supporting the introduction of Open Source software and Open Data Standards in the Public Administration

Project acronym: COSPA



Work Package 2

Collection of requirements for OS applications and ODS in the PA and creation of a catalogue of appropriate OS/ODS Solutions

Deliverable 2.8/2.9

Prioritisation of requirements for OS and ODS

Contract no.: IST-2002-2164



Project funded by the European Commission under the
“SIXTH FRAMEWORK PROGRAMME”

Project Acronym	COSPA
Project full title	A Consortium for studying, evaluating, and supporting the introduction of Open Source software and Open Data Standards in the Public Administration
Contract number	IST-2002-2164
Deliverable	2.8/9
Due date	25/05/04
Release date	10/06/05
Short description	<p>WP2 primarily aims to understand what OS tools are currently utilized in PAs, and to which ODS the currently produced data is compliant. At the same time this work package intends to gather and analyse user requirements from the partner PAs, in order to devise possible OS/ODS solutions. We want to identify legacy database applications as reference for subsequent elaboration in the subsequent work packages (WP5). Different countries and specialisation of PAs generate a variety of different environments and problems to be addressed. Business administration is a dynamic activity: new laws, procedures, and responsibilities are being placed on PAs every day. This means that the requirements are not static, that requirement gathering is a continuous process, that there is a need for flexibility, adaptability and interoperability. This document presents a method for helping decision makers to determine the importance of the requirements.</p>
Author(s)	IBM, Free University of Bozen/Bolzano
Contributor(s)	University of Sheffield
Project Officer	<p>Tiziana Arcarese Trond Arne Undheim European Commission Directorate-General Information Society Directorate C - Unit C6- eGovernment, BU 31 7/87 rue de la Loi 200 - B-1049 Brussels - Belgium</p>

Disclaimer

The views expressed in this document are purely those of the writers and may not, in any circumstances, be interpreted as stating an official position of the European Commission.

Neither the European Commission nor any person acting on its behalf is responsible for the use which might be made of the information contained in this document.

Reference herein to any specific products, specifications, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favouring by the European Commission.

Copyright

This work is licensed under the Creative Commons Attribution-non commercial-NoDerivs License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Table of Contents

Deliverables Navigator.....	7
Executive Summary.....	8
1 Introduction.....	10
1.1 Definition of Terms.....	11
1.2 Scope of the Document.....	12
2 Method Overview.....	13
3 Method Phase: Plan & Organize.....	16
3.1 Information Requirements Definition.....	16
3.2 Requirements from Survey.....	17
4 Method Phase: Develop.....	18
4.1 Documents Review.....	18
4.2 Quality Check.....	18
4.2.1 Requirement Workshop / Survey.....	18
4.2.2 Consolidation.....	19
4.2.3 Validation.....	19
4.3 Software Group Check.....	19
4.3.1 Software Grouping.....	19
4.4 Studies Analysis.....	19
4.5 Knowledge Bases Analysis.....	20
4.6 Criteria Collection / Categorisation.....	20
4.7 Information Consolidation / Criteria Selection.....	20
4.8 Quality Check.....	20
4.8.1 Criteria Workshop / Survey.....	21
4.9 Procedure Selection.....	21
4.10 Alternative Approach.....	21
5 Method Phase: Review.....	23
5.1 Procedure Review.....	23
5.2 Quality Check.....	23
5.2.1 Procedure Workshop / Survey.....	23
6 Method Phase Usage.....	24
6.1 Applying Procedure.....	24
7 COSPA Usage of the Method.....	25
8 Procedure used in the Example Application.....	27
8.1 Discrete Cost-Benefit-Analysis.....	27
9 Application of the Discrete Cost-Benefit-Analysis.....	30
10 Conclusions.....	35
Appendix A - Abbreviations.....	36
Appendix B - Glossary.....	37
Appendix C - Software Groups.....	38
Appendix D - Categorised List of Criteria.....	39
Appendix E - Fuzzy Cost-Benefit-Analysis.....	40
Appendix E.1 - Weights Variation.....	40
Appendix E.2 - Using Fuzzy Scores.....	43
Appendix F - Questionnaire for Requirements.....	46

Appendix G - References..... 47

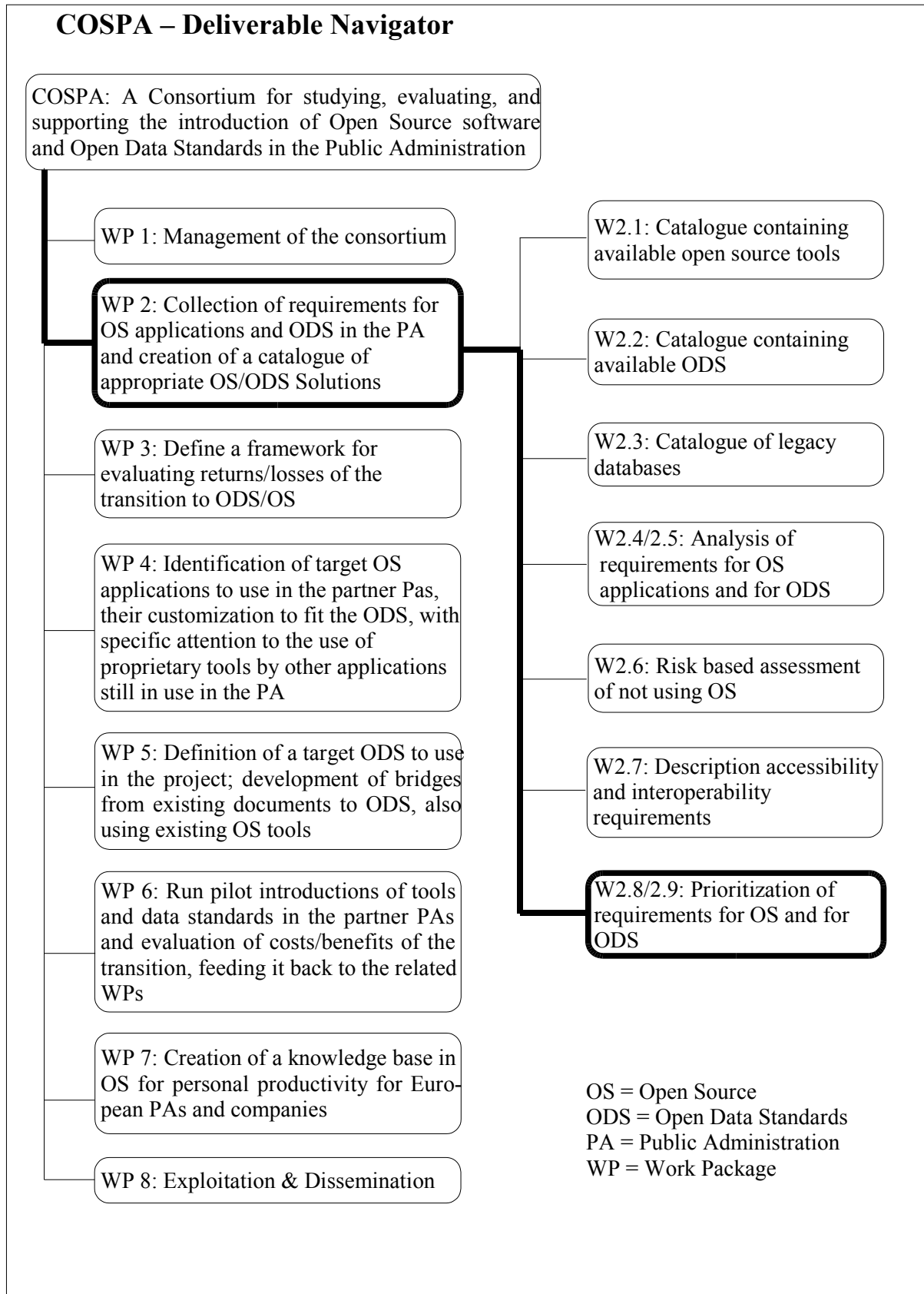
Index of Figures

Figure 1: Flow diagram of the Method for software requirements.....	14
Figure 2: Flow Diagram of the Method for data standards requirements.....	15
Figure 3: Decision graphs.....	41
Figure 4: Orness as a measure of the evaluators' optimism.....	42
Figure 5: Total score of a requirement.....	43
Figure 6: Fuzzy Numbers for "low", "medium" and "high".....	44
Figure 7: Decision Graph with fuzzy numbers.....	45

List of Tables

Table 1: Example Cost-Benefit-Analysis - smaller font refers to entries explaining the steps of the procedure, larger bold font entries show the final results.....	28
Table 2: Office programs requirements.....	31
Table 3: Prioritisation for progressive PA.....	32
Table 4: Prioritisation for preserving PA (red are changes to the progressive PA).....	33
Table 5: Prioritised List of Requirements.....	34
Table 6: Category and Criteria List.....	39
Table 7: Example variation of weights for the fuzzy Cost-Benefit-Analysis.....	43

Deliverables Navigator



Executive Summary

Objective

This document introduces a method for prioritising a given set of software or data standards requirements. The method has been applied on the data of Deliverable D2.4 for exemplifying its usage.

The goal of the method is to enable PAs managers to prioritise their own software or data standards requirements. Different requirements, analysis procedure, and context will produce different prioritised lists.

The method takes as input a set of requirements coming from the PAs and produces as output a list of requirements ordered by importance according to different level of requirements details.

The prioritised list is a powerful instrument used as baseline for the software acquisition process. In this process features of current or new software is checked against the requirements priorities retrieved by the method to verify usage against needs and to calibrate emphasis on existing requirements; this supports a software reorganization of a PA.

Requirements' priorities may also provide an instrument to evaluate software demand and address new market expansion. Software companies may take advantage of it developing new products or features and extending support of existing ones.

The prioritised output lists of this method may describe an overall state of the requirements related to the global context in which the method has been applied as well as a more detailed picture of the requirements' status of a single Public Administration. This depends on the procedure considered within the method.

In the COSPA project the method also provides an instrument supporting the feasibility analysis of a transition to a new solution (for example towards Open Source) as it identifies software or data standards requirements that are essential or required by the PA partners. Results of the derived analysis may be useful for the experimental phase of the project when Open Source Software and Open Data Standards are empirically tested in terms of the priority lists of software or data standards requirements.

The deliverable is in its final version and provides an example of application of the method on the COSPA data.

Methodology

The method has been created based on existing literature and similar approaches for various problems in Information Technology (e.g. Software Quality Models).

The work has been directed by IBM. FUB and IBM have collaborated for the definition and the implementation of all its parts. Working on Deliverable D2.4 the University of Sheffield has provided substantial information on input standards and formats. Criteria for the application of the method have been discussed within the whole Consortium.

Method's Value

The advantage of having prioritised requirements is a greater amount of objectivity in the requirements' categorisation. Objectivity in categorisation is useful for identification of the more important requirements by reproducible means. Up to now the prioritisation of requirements has only be done by subjective means resulting in possible neglect of externally imposed constraints or overestimation of needs.

On having an objectively prioritised list of requirements it is easier to choose negligible requirements and drop them in the following selection process.

Another advantage is the possibility of documenting in a reproducible way the decisions taking during software procurement or data format migrations.

1 Introduction

This document introduces a method to prioritise requirements of public authorities (PAs). It has been created within the COSPA project.

The method takes as input a set of requirements coming from the participating PAs and produces as output a list of requirements ordered by importance according to different levels of requirements' details. Therefore, the prioritised list is a powerful instrument for decision making at different levels of abstraction. For example, a manager of a PA may use it to decide whether to upgrade an existing or to acquire a new software on the base of its own current most important requirements. A software company may use the results of the method to enter or cope with a specific market sector on the base of the public known most important requirements. In the two cases two different levels of requirement details are needed. In the former case a ranking by importance of detailed local requirements, in the latter a ranking by importance across different PAs is expected. Therefore the method includes a section on quality assurance, where requirements coming from different sources are checked for compliance with standards. These standards are defined in terms of the different motivations.

In the COSPA project the method is applied to the requirements gathered from the PA partners. The method contributes to the project providing an instrument for the analysis of the current state of the software needs in the PA partners. The results may be then used for a feasibility analysis of a transition to a new solution, possibly an OSS. Results of this analysis may support the experimental phase of the project when Open Source software and Open Data Standards are empirically tested in terms of the priority list of software or data standards requirements.

We present the method detached from its use within the COSPA consortium to facilitate the adoption of the method in other PAs not directly related to COSPA. The results of the method in the COSPA project provides only an example of its usage; further adoptions may produce different results according to different context and analysis procedure.

The method consists of a set of sequential steps that guarantee:

- Quality of the requirements according to pre-defined standards
- Quality of the set of criteria assessing the requirements
- Validity of the procedure to process the requirements in term of the motivations
- Flexibility according to the major goal of its use

These steps and the related activities are described as overview in chapter 2 and in more detail in chapters 3 to 6.

The document consists of the following parts:

Chapter 2: Graphical display of the method to obtain a first overview together with a short description of the steps and the activities of the method.

Chapters 3 to 6: Description of the steps and activities of the method in detail. Chapters refer to steps, sections and subsections to activities.

Chapter 7: Description of the usage of the method within the COSPA project.

Chapter 8: Presentation of a process for prioritisation i.e. a Cost Benefit Analysis.

Chapter 9: Exemplary usage of the method for two different PAs using some data from D2.4

as input.

Appendix A: A list of abbreviations used in this document.

Appendix B: The list of terms and abbreviations used in this document.

Appendix C: The software groups used in the method as derived in D2.4.

Appendix D: The list of the criteria and their categories found for prioritising the requirements.

Appendix E: An explanation of methods to enhance objectivity in Cost Benefit Analyses through usage of fuzzy mathematics.

Appendix F: The questionnaire for software requirements from D2.4.

Appendix G: The list of references used in this document.

1.1 Definition of Terms

In the method different groups / teams participate. Their roles are:

- The Requesters: People who are interested in a prioritisation of software or data standards requirements. They initiate the processing of the method and define motivations for its use. The standards for the requirements have to be compliant with these motivations
- The Developers: People who conduct the method from the first to the last step. Their main tasks are: checking the input data for quality, selecting the analysis procedure and applying it to the software or data standards requirements
- The Suppliers: People who provide the software or data standards requirements. There are three different types of Suppliers: managers, system administrators and users. The input into the method differs depending on the type of Supplier

Though it is possible to have a single person in all three roles, this should be avoided to diversify the input and to get the best possible results.

In the COSPA project the three above mentioned roles are represented by the following groups:

- The consortium itself represents the Requesters.
- IBM and FUB are the Developers.
- The PAs and their shepherds are in the role of the Suppliers.

On using the method within a PA the roles would normally distribute as following:

- A manager to decide a software acquisition or reorganization may request requirements to be prioritised. This person would be the Requester.
- The Developer would be a member of the staff – most likely from the IT department – being charged with the prioritisation or an external consultant.
- The Suppliers are the future users – i.e. the colleagues of the Developer – as source of information.

1.2 Scope of the Document

This document is aimed mainly towards technical and academic readers. It is thought to support Small and Medium Enterprises (SMEs) assisting PAs through IT-consultancy by providing them with a method for more objectively prioritising software and data standard requirements.

PA decision makers should read for a description of the method's value and decide on the information given there.

The scope of this document cannot be the production of a universal prioritised list of requirements, since the order of requirements strongly depends on the social and political background of the PA applying the described method or the procedure used.

E.G. for a progressive PA a prioritised list of requirements might be: “Localised graphical user interface”, “Supports multiple languages”, “Localised spell-check”, “Localised hyphenation”, “Localised thesaurus”, “Supports user developed macros” where as for a preserving PA the same requirements might be ordered as: “Localised graphical user interface”, “Supports user developed macros”, “Provides good user help”, “Provides users customisable menu options”, “Supports multiple languages”, “Localised thesaurus” (see Tables 3, 4, and 5).

This documents describes a method which is a Model of Good Practice rather than trying to find a prioritised list of requirements applicable to all European PAs.

2 Method Overview

The two methods (one for software requirements, the other for data standards) have been developed to enable their users to prioritise requirements and identifying the most important ones.

A graphical representation of the method for prioritisation of software requirements is shown in Figure 1, the one of data standards requirements is shown in Figure 2.

The steps of the methods are explained in chapters 3 to 6. Each chapter refers to a phase of the method. A phase consists of a set of connected steps. The description of the method holds for both methods in parallel since they differ only in one step (a grouping of software is not sensible for data standards). Each step of the method (represented by an item in the graphical display) is described in one section. If the steps are optional, they are described in subsections following the section where the decision has been taken.

This approach for organising chapters and (sub)sections results in some chapters consisting of a single section. The chapter and (sub)section numbers have been displayed in the graphical representations of the methods for easy reference to the respective textual description.

The method is structured into the four phases “Plan & Organise”, “Develop”, “Review” and “Usage”.

The **Plan & Organise** phase is used to define the needed information from which to devise a prioritised list of requirements. This includes defining:

- the data format compliant to the context and motivation of the use of the method, and
- the quality standards the requirements have to fulfil to be acceptable input for the method.

The **Develop** phase includes all the tasks which are needed for the development of the prioritisation method itself. Once the software or data standards requirements comply with the quality standards they are categorized and grouped for a better understanding of their importance scope. Existing literature support this analysis. The inputs from two different sources (Suppliers information and Developers analysis) are merged. In the case of software or data standards requirements there might be a further step for grouping different software types if that seems necessary.

The **Review** phase is used to allow the Requesters to have a last check whether the method is capable of producing the desired output.

- The **Usage** phase is used in the method to carry out the actual prioritisation of the given requirements. The result of this phase is a prioritised list of requirements with the appropriate documentation how this list has been obtained.

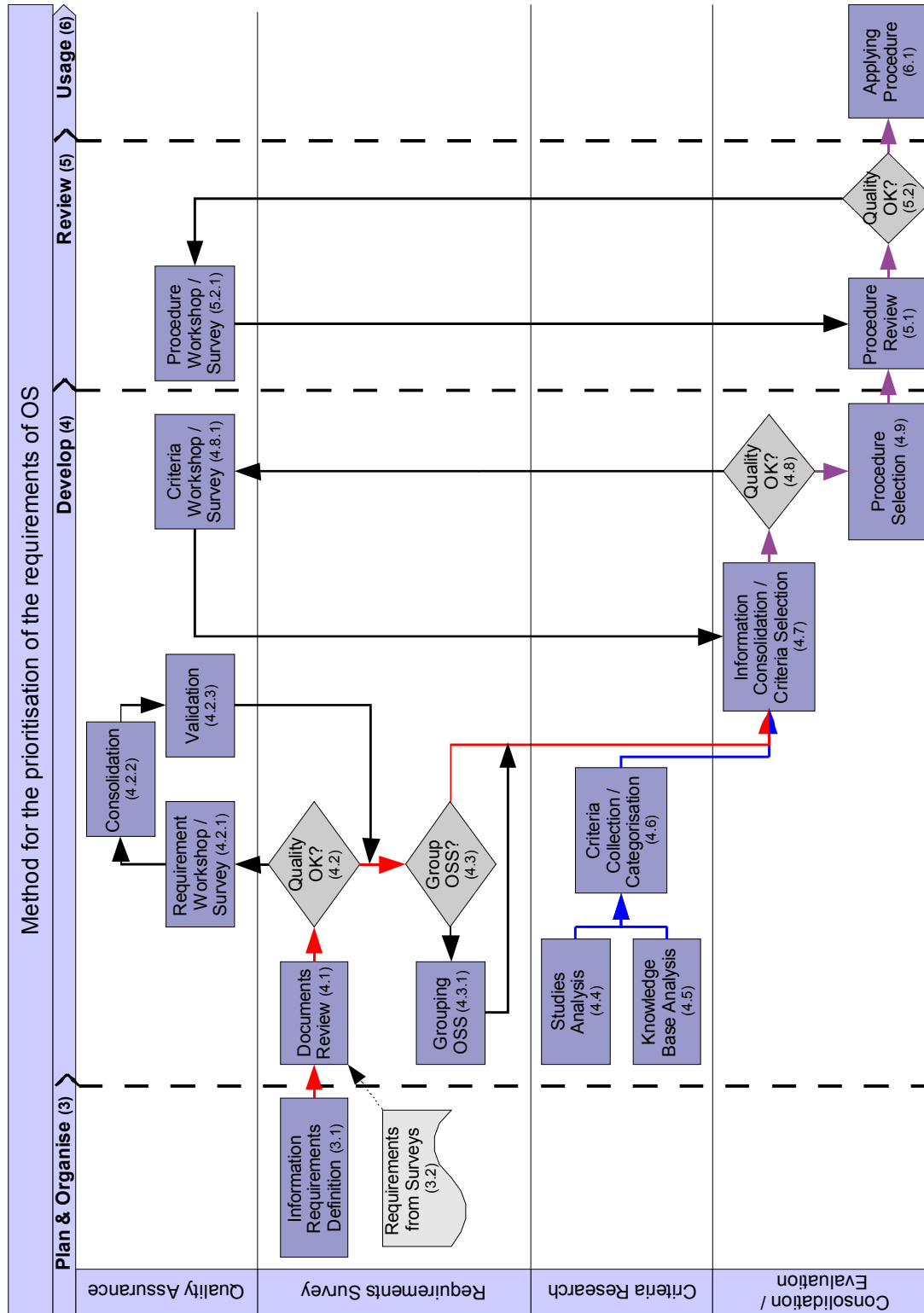


Figure 1: Flow diagram of the Method for software requirements

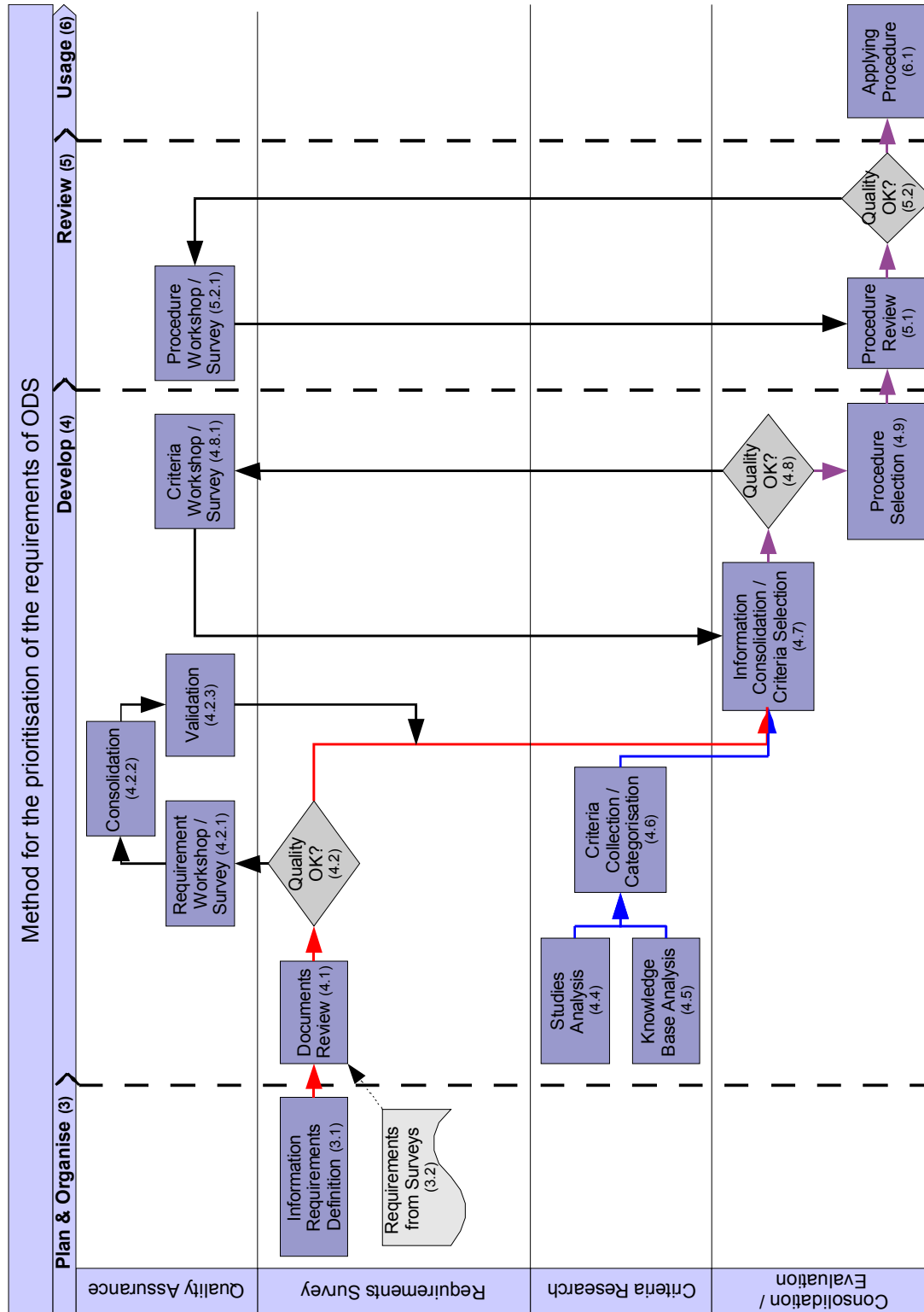


Figure 2: Flow Diagram of the Method for data standards requirements

3 Method Phase: Plan & Organize

This chapter explains the **Plan & Organise** phase of the method.

3.1 Information Requirements Definition

This means the definition of the informational input for the method.

This input for which the format and quality are defined here are requirements that a new software (e.g. OSS) or data standard (e.g. ODS) has to fulfil to be usable in the administrative business processes of a given PA. Namely, the requirements to be prioritised should arise from inspections of the administrative business processes in which the software or data standards are going to be used in. This inspection of administrative business processes also yields which requirements are mandatory and which the software or data standard therefore has to fulfil, otherwise the new product will be rejected.

Constraints may also be imposed by an existing infrastructure and by the motivations that drive the prioritisation need. For example if the goal is to understand an overall software or data standard status of a set of PAs then the requirements would not be so detailed as they were describing the peculiar needs of a single PA that wants to transit to a new software solution.

If a definition of such requirements is already available this can be taken and – if needed – adapted. If there is no requirements definition in place, one can use different approaches depending on:

- the required level of detail, and
- the possible involvement of Suppliers.

There are two major approaches: requirements inquiry and requirements inspection.

Requirements inquiry allows to proceed “from the native point of view”, obtaining information about requirements by either:

- talking to users,
- observing them using the system in a real working situation, or
- letting them answer questions verbally or in written form.

Requirement inquiry techniques include ethnographic studies, interviews and workshops, as well as surveys.

Requirement inspection is based on having the Developers inspecting the processes themselves and from there deducting the requirements. While this approach requires less involvement of Suppliers, using only requirement inspection cannot provide the same level of detail as obtained with requirement inquiry. Requirement inspection techniques include heuristic evaluation and cognitive walk through.

To be usable for the method, the list of requirements have to fulfil some quality standards, which include:

- Accessibility of the data (e.g. legible, known data format, ...),
- Completeness (e.g. no partially filled out questionnaires, several different answers, ...),

- Coherence (e.g. no contradictions within one questionnaire, level of detail in the data is sufficient, ...),
- Diversity (e.g. questionnaires filled out by different types of Suppliers, not different copies of a single answer, ...),
- Definiteness (e.g. each entry in the input refers only to a single requirement),
- Process coverage (i.e. all requirements imposed by the administrative business processes to be covered are listed), and
- Quantity (e.g. prioritising a too small set of requirements is not sensible).

Actors involved in this step are:

- Requesters to provide the information on the motivation to use the method and information on the software or data standards definition required,
- Developers to develop the quality standards and the relative input detail they need to apply the method, and
- Possibly the Suppliers since they are asked to provide the input.

3.2 Requirements from Survey

This represents the input from the Suppliers after they have compiled it or the requirements gathered by the Developers' inspection. Suppliers and Developers here work in strict collaboration.

4 Method Phase: Develop

This chapter is the core of the method: the gathered requirements are checked against standards; categories and criteria are selected on the base of existing literature and internal knowledge; an eventual grouping by type of software may be performed. The key actor here is the Developer. Requesters may enter this part to input the motivations of prioritisation.

4.1 Documents Review

The intention behind this part is to give the Developers the possibility to consolidate the input from different Suppliers into a data format they intend for internal use. This format is most likely either a spread sheet or a data base to facilitate further processing of the input.

Also the delivered data are checked against the defined quality standards for this data collection (Section 3.1). This check of the quality usually is a discussion between the Developers whether they think the requirements are sufficient and seldom done by more elaborated (and more objective) means.

This step purely involves the Developers.

4.2 Quality Check

If the input data does not meet the associated quality requirements, further steps (described in the following subsections) are inserted to enhance the data's quality prior to continuation with the main method.

4.2.1 Requirement Workshop / Survey

To improve the input data quality the first action to be taken is a refinement of the already available data. This can be done either by a workshop or by a survey. From these two possibilities the workshop is the better choice because here the quality can be improved better than by a survey carried out by the Developers since here a more direct communication between the Developers and Suppliers is established enabling an easier solving of problems in understanding.

The aim of the workshop should be to clarify all points which were left open by the original data. It also can be used to show the Suppliers how to provide the required input data in an appropriate form. The workshop not necessarily needs to be a physical meeting but also virtual meetings like conferences via the Internet (e.g. e-workshop) or by phone are possible.

A survey, where the Developers fill in the missing details based on assumptions and their knowledge of the situation is a poorer choice compared to the workshop since it does not provide the same quality of input as the workshop would produce. This should be avoided to guarantee the originality of the data and their conformance to the PA's domain.

A survey only needs the action of the Developers whereas the workshop involves both Developers and Suppliers.

4.2.2 Consolidation

The data obtained by the Requirement Workshop / Survey (Section 4.2.1) and the original data are merged here. Ideally this can be done by both groups (the Suppliers of the original data and the Developers) together.

It is possible though to shift this task to either group if necessary (e.g. one of the groups has no resource available to do this task).

4.2.3 Validation

This is the last check done by the Developers to see whether all the missing details have been provided. It is assumed in the flow of the methods that this check is only a last check and that there is no need for a second quality improving round. This holds true mainly since this validation usually already takes place during the workshop. In the case of a survey it is just a check if the work has been completed.

4.3 Software Group Check

Only in the case of requirement prioritisation for software, a check for grouping of the input is needed. This grouping can reveal either the possible shift of meaning for a requirement depending on its usage or the impossibility of applying all requirements to all types of software.

The most general categorisation of software types are server software and client software. Software requirements for the former concerns mainly software attributes like stability and availability whereas those for the latter regards more usability or accessibility. The difference is influenced by the target user. Another example of a possible software grouping is given in appendix C which originates from D2.4.

This inspection is carried out by the Developers.

4.3.1 Software Grouping

The input data of software requirements to be prioritised is grouped according to the software groups found while inspecting it (Section 4.3).

4.4 Studies Analysis

The input into the method not only consists of the data provided by the Suppliers (Section 3.2) but also of some criteria for prioritising the requirements. One way of finding these criteria is analysing external sources for similar projects and existing literature. The goal should be to obtain as many different criteria as possible. Chapter 7 includes a list of possible literature.

External sources are also fundamental for choosing the procedure (Chapter 6) going to be used for the prioritisation in accordance to the motivation and the goal driving the prioritisation.

Responsible for this action is the group of the Developers.

4.5 Knowledge Bases Analysis

The second possibility for obtaining criteria for the prioritisation of software and data standards requirements is searching internal data sources. This is the third source of input into the method. It relies on the invaluable collection of knowledge which is usually gathered in every departments or firm dealing with IT problems. It is advisable for the Developers to find as many potential criteria as possible.

4.6 Criteria Collection / Categorisation

The criteria found by Studies Analysis (Section 4.4) and Knowledge Bases Analysis (Section 4.5) have to be combined and grouped into categories for easier further processing.

The idea for the criteria categories is to keep together similar criteria which are concerned with different aspects of one concept. Like the two criteria “administrative efforts” and “security issues” could be grouped as “technical” criteria since they are imposed by technical staff or conditions.

This is a task performed by the Developers.

4.7 Information Consolidation / Criteria Selection

All informational inputs - requirements from survey (Section 3.2) and criteria from analysis (Sections 4.4, 4.5) - are merged by the Developers to form the resources for the further progress of the method. This merging allows selecting the meaningful criteria for the prioritisation. The resulting set of criteria not necessarily contains all findings.

4.8 Quality Check

This quality check determines whether the selected criteria (section 4.7) meet the required standards and if they are sufficient for the prioritisation task.

This quality check should be carried out by the Developers and the Requesters. If the resulting set of criteria is not suitable or the Requesters consider the requirements as incomplete, an action for improving the quality of the merged input is inserted to obtain a better set (Section 4.8.1). The actions of selecting criteria (Section 4.7), checking their quality and improving it if necessary are repeated until the quality of the criteria set is sufficient.

Reasons for rejecting the chosen set of criteria or the given input include:

- there are no requirements to be measured for criteria defined to be essential, or
- there are no criteria found to measure for requirements defined to be essential.

4.8.1 Criteria Workshop / Survey

Here the same actions as for requirements' quality (section 4.2.1) are performed.

After the survey or workshop the newly found criteria are consolidated by the Developers with the existing ones (Section 4.7) and another criteria selection and quality check is carried out.

4.9 Procedure Selection

After an appropriate set of criteria for the given requirements' prioritisation has been found, the prioritisation procedure itself is selected. The task of the procedure is to take as input the requirements and the determined criteria set and give as output a prioritised list of the requirements. The development of this procedure is carried out by the Developers.

The capability of the procedure to prioritise the software or data standards requirements is obtained by transforming the found criteria into a model which can be used to process the input requirements. The result of this processing is a prioritised list of the requirements.

The range of possible procedures starts with a simple count of criteria fulfilled by a requirement and goes up to elaborate computer programs, which use existing formulae or techniques for obtaining the prioritised list and even further. It is also possible to carry out different procedures consecutively and thus merging their different outputs into a single list.

The given range of procedures includes:

Ordering the requirements according to the number of criteria they fulfil or have positive influence on.

A cost-benefit-analysis. This is a procedure of weighted averaging of the number of fulfilled criteria per category.

A questionnaire asking different people to evaluate if a criterion is supported by a requirement or not and adding the obtained opinions.

Prioritising the requirements for each criterion separately and merging the thus obtained list into a single one by either sorting according to the count of first, second, third places the requirement obtained or by giving points according to place for the requirement and sorting according to the sum of these points¹.

As can be seen, the possibilities for procedures are countless. The main task for the Developers is to find and develop an appropriate procedure for the special case they undertake the prioritisation for.

4.10 Alternative Approach

Another approach for prioritising the requirements is a simple count of administrative business processes this requirement is needed for. It is also possible to group the processes

¹ The first is done – at least in Germany – for obtaining a ranking of the nations in the Olympic games, they are sorted according to the number of gold medals obtained, then by the number of silver medals and last by the number of bronze medals. The second is done in the Grand Prix d'Eurovision de la Chanson where a first place within a participating nation is worth 10 points, the second 8, the third 7 and so on.

by departments or other categories and then do a weighting for the categories similar as the one discussed for the prioritisation by criteria.

This approach is useful if the found criteria do not seem sufficient or the new product (software or data standard) is used in a decently large number of processes.

This approach does not need any criteria so the method's parts concerned with criteria research can be neglected.

5 Method Phase: Review

The phase **Review** contains the tasks related to finalising the method and assurance of the selected method's fulfilment of the desired requirements.

5.1 Procedure Review

In this part of the method the Requesters have the possibility to do another review of the developed procedure and request changes so that their requirements towards the method are fulfilled.

Possible procedures for this review of the Developers' work by the Requesters include:

- A presentation of the developed procedure by the Developers explaining the chosen procedure and asking the Requesters if they agree with this approach.
- An application of the procedure towards a set of requirements defined by the Requesters and changing the procedures parameters until a previously defined order within those requirements has been obtained.

Once again the possibilities of progressing with the method are numerous and depend on the precise situation the procedure is developed for.

5.2 Quality Check

If in the course of the procedure's review there are remarks of the Requesters or the Requesters do not think the developed procedure a suitable means for the prioritisation, the procedure will be refined or reworked respectively. If the suggested procedure is accepted by the Reviewers, the procedure will be applied and the results evaluated.

5.2.1 Procedure Workshop / Survey

If the procedure is not accepted by the Reviewers or they had additions or comments towards the procedure, the Developers have to modify the procedure accordingly.

Usually the modifications will be done by the Developers on their own. It might be helpful though, if there is a immediate need of Requesters' feedback, to arrange a workshop between Requesters and Developers.

The procedure is refined and reviewed by the Requesters until it is accepted by them.

6 Method Phase Usage

In the phase **Usage** the developed procedure is applied to the input data and the output is produced.

6.1 Applying Procedure

The method is finalised by applying the developed prioritisation procedure (Section 4.9) to the input data. The result of that application is the desired list of prioritised requirements to be used in the task the prioritisation has been carried out for. Another output of the method's application should be a set of documents describing the way the method has been applied and also its input and output. These documents are usable in reviews and further applications of this method.

7 COSPA Usage of the Method

The method described in chapters 2 to 6 in this document has been put to use within the COSPA project in the following way:

The data gathering (Section 3.2) has been carried out by interviewing the partner PAs. For this interview a questionnaire was developed to take up requirements towards OSS. This questionnaire is part of Deliverable D2.4. The questionnaire is given in appendix F.

Since we were not satisfied with the data (i.e. many answers were just copies of the examples and only a small number of answers were returned) there has been a second questionnaire (this time on-line), which was focussing more on administrative business processes than on technical requirements. The data collected by the web site is the base input for Deliverable D2.4. To obtain a better data input for Deliverable D2.4 shepherds assisted the PAs in filling in the questionnaire by telephone or visits to the PA.

To exemplify the method in this deliverable we use the data obtained from the first questionnaire, because the method is designed to prioritise technical requirements, which are not directly gathered by the second questionnaire any more. To extract requirements from the administrative business processes an extra inspection process would be needed. The inspection of administrative business processes is more suitable for obtaining requirements towards new software. Here a high level of detail in the process descriptions is required.

Moreover in the first round quality assurance was taking place in parallel to the collection of that data (Section 4.1) and was carried out in cooperation by direct communication (phone and e-mail) between IBM and USFD. This has guaranteed the good quality of the requirements we use here for the exemplification, though it also reduced the requirements' number.

This cooperation between IBM and USFD in the quality management for D2.4 covers all parts of this method up to Software Grouping excluding (Section 4.3). The standards applied for quality assurance have been:

- at least three answers from each participating PA,
- answers compiled by the PAs on their own accord, and
- answers stating concrete requirements.

For the search for criteria in external sources (Section 4.4) the following ones have been consulted (among others):

- LiMux – The migration of desktops from WindowsNT4.0 to Linux in Munich. Here especially the client study has been taken into account [1].
- FLOSS Report – A study to publicate information on the work of the OpenSource Community. Here especially the parts II and IIB of the final report are of interest [2].
- Programma OSOSS – A study on OSS usage in the PAs in the Netherlands [3].
- IDA - Study into the use of open source software in the public sector. Here the report on OSS usage is of interest [4].
- Berlecon Research – A basic report on “Linux and Open Source Strategies for CIOs” [5].
- Forrester – A report on a survey of managers concerning the introduction of Linux [6].

For the internal research for criteria (Section 4.5) IBM as a partner in the COSPA project relied on its yearlong experience in consulting. These criteria have been submitted to discussion within the COSPA mailing list and the criteria list was thus determined.

As result of the criteria research (Sections 4.4 and 4.5) a list of over two dozen criteria has been found. These criteria have been grouped into six categories. This complete list of categorised criteria is given in appendix D.

For consolidating the criteria and the data input (Section 4.7) the found criteria have been checked against the input from D2.4 and the appropriate criteria have been determined for further usage. The used criteria can be seen in the example application of the method in chapter 9. In chapter 8 the used procedure to be used in the method's application is presented and discussed.

8 Procedure used in the Example Application

In section 4.9 we listed different approaches to prioritise a set of items according to given standards. In this chapter we present a possible model that we have selected for the prioritisation of the requirements of Deliverable D2.4.

The technique we have selected for use in the example application (chapter 9) and discuss in this chapter is suitable to prioritise the requirements towards different objectives. Among these are the possibility of ranking specific requirements one PA might have or to rank general requirements PAs from a certain area or country might have towards a type of software or data standard.

8.1 Discrete² Cost-Benefit-Analysis

For an easy and straightforward approach to prioritise the requirements we decided to use a discrete cost-benefit-analysis. This approach is well known since the late 70ies when it has been adopted as a quality model [7]. It is also a classical procedure in multivariate statistical analysis.

First each category of criteria and each criterion gets assigned a weight identifying the importance it has for the Requesters. Preferably the criteria weights within each category and the ones of the categories sum up to 100 percent or 1.

As a reminder, the found criteria (e.g. license costs, training requirements, support availability, ...) have been grouped into categories of criteria with similar scope (e.g. legal and political criteria, technical criteria,). The complete list of criteria and categories used in the COSPA project are given in appendix D.

As next step in the cost-benefit-analysis each requirement is checked against each criterion and assigned a score in a pre-defined ordinal scale representing whether this requirement fulfils and / or supports this criterion or not.

As third step all given point-values are multiplied by the weight of this criterion, summed up within a category and divided by the sum of all weights within this category.

Finally all sums of the categories are multiplied with the appropriate category weight, these products are summed and divided by the sum of category weights.

The prioritisation is now given by sorting the requirements according to the thus obtained values.

Table 1 gives an example of the above described procedure. Here we took three categories with three, four and five criteria. For the numbered part we took discrete values from 0 to 2.

2 The cost-benefit-analysis is called discrete, since the weights are discrete values as opposed to the fuzzy values discussed in an appendix. So the term discrete is used to differentiate between fuzzy and normal (integer) weights.

	Category 1	Criterion 1.1	Criterion 1.2	Criterion 1.3	Score Cat 1	Category 2	Criterion 2.1	Criterion 2.2	Criterion 2.3	Criterion 2.4	Score Cat 2	Category 3	Criterion 3.1	Criterion 3.2	Criterion 3.3	Criterion 3.4	Criterion 3.5	Score Cat 3	Total Score
Weights	30	20	50	30		45	25	25	25	25		25	15	15	30	30	10		
Formula Symbols	w_1	w_{11}	w_{12}	w_{13}	s_1	w_2	w_{21}	w_{22}	w_{23}	w_{24}	s_2	w_3	w_{31}	w_{32}	w_{33}	w_{34}	w_{35}	s_3	s
Req. 1		1	2	0			1	0	0	2			0	2	2	1	1		
$s_{ij} * w_{ij} / \sum w_{ij}$		0,2	1	0			0,25	0	0	0,5			0	0,3	0,6	0,3	0,1		
Σ					1,2						0,75							1,3	
$s_i * w_i / \sum w_i$					0,36						0,34							0,33	
Σ																			1,03
Req. 1		1	2	0	1,2		1	0	0	2	0,75		0	2	2	1	1	1,3	1,03

Table 1: Example Cost-Benefit-Analysis - smaller font refers to entries explaining the steps of the procedure, larger bold font entries show the final results.

Here are the formulae which have been described above in the application of the cost-benefit-analysis:

$$s_i = \frac{\sum_j w_{ij} s_{ij}}{\sum_j w_{ij}} \quad (8.1)$$

with w_{ij} representing the weight for criterion j in category i and s_{ij} the score given for the requirement with respect to this criterion.

The total score for the requirement is:

$$s = \frac{\sum_i w_i s_i}{\sum_i w_i} \quad (8.2)$$

with w_i the weight of category i .

If the categories are independent of each other – this should be the case for a carefully chosen set of categories – a weighted average does not carry all the information (e.g. adding apples and oranges). Here a multiplication of the categories scores introduce a different interesting meaning: extreme scores are more emphasized. In analogy to a weighted average follows:

$$s = \prod_i w_i s_i = c \prod_i s_i \quad (8.3)$$

If the weights are kept constant the product of the weights is a constant itself and can therefore be neglected. In appendix E an extending technique is introduced, where the weight becomes needed again.

Though the product emphasises positive values for single categories interpretation of the results might be difficult since the results are on a highly potential scale. If one wants to assign qualitative terms to the different numbers used in the procedure the following formula can be used to obtain “legible” results³:

$$s = \sqrt[n]{\prod_i s_i} \quad (8.4)$$

where n is the number categories.

The difference between these two approaches is that the Sum is acting like a boolean OR and the Product like a boolean AND. The score obtained by the summing formula (8.2) takes all categories as equals (if all weights are equal) whereas the score obtained by the multiplication formula (8.3) emphasises the ones with the higher scores.

The main advantages of this procedure (Cost-Benefit-Analysis) are:

- easy to handle,
- on using categories and subcategories the influence of a single score-value towards the complete score is obscured, and
- accepted procedure.

The main disadvantage of this procedure is:

- It is strongly dependent on the Requesters and Developers. Weights are given subjectively. Different evaluators may outcome with different priority values and lists.

This disadvantage can be overcome by using a fuzzy cost-benefit-analysis as described in appendix E.

3 The unusual normalisation $\sqrt[n]{x}$ comes from the following consideration: If we assume all scores used in the Product are equal, we would like to have the final score of the same value. The normalisation factor arises from the product of scores coming up to: $\prod_i s = s^n$.

9 Application of the Discrete Cost-Benefit-Analysis

In the example below we describe how to fill the Table 1 using part of knowledge gathered from the the COSPA requirements (Deliverable D2.4). Furthermore we show the influence of the social and political background on the prioritisation results.

The following are the weights and categories defined for the COSPA project:

- 20% Legal and Political:
- 15% Technical
- 15% Strategic
- 20% Documentation / Training / Support
- 20% Usability / Accessibility
- 10% Other

These figures are based on a suggestion by IBM and have been adopted after discussion in the COSPA mailing list. They are specific for the COSPA project and may guide as basic values in possible later applications of the method presented in this document. There is no general method for obtaining these values. A good approach for obtaining them is to start with an equal split between all categories and then take away some points from categories which are deemed less important and distribute these among the ones deemed more important. The category weights are just mathematical remedy and still contain some degree of subjectiveness. This subjectiveness can be reduced by using variation techniques as described in appendix E.

We then suppose that the weights for the criteria are equally distributed. So for the category “Legal and Political” the criteria “Legal criteria,” “Reducing Licensing Costs,” and “Improving Data Protection” have the same weight 33%.

From inspection of the in D2.4 obtained data we found several requirements towards text processing programs which were stated by the PAs. These requirements are collected in Table 2. They have been collected from all PAs' answers. Requirements for text processing programs have been chosen since from administrative business process inspection this software type is involved in a vast majority of processes.

Localisation Requirements	
Localised spell-check	Localised graphical user interface
Localised thesaurus	Supports multiple languages
Localised hyphenation	
Layout and Composition Requirements	
Provides document layout style guides	Supports document composition
Enables document/data source linkage	Enables quality presentations to be produced
File Format Requirements	
Supports Multiple file formats	Supports XML Data import/export
Enables data conversion from one file format to another	Import/export from/to open and legacy formats
Export to PDF format	
Functionality Requirements	
Functions defined in ECDL Syllabus 4.0	Provides good user help
To provide word processing, presentation, spreadsheet & database functionality	Supports document editing with revision marks
Supports user developed macros	Graphically editing HTML documents
Provides users customisable menu options	
Technical Requirements	
Operates in a thin client environment	Integrates with a single LDAP sign on server
Automation possibility	

Table 2: Office programs requirements

To illustrate the influence of the PA's environment on the prioritised list of requirements we will assume that we have two different environments which are as much opposed to each other as possible. For each of these environments a prioritisation is carried out and the result presented.

Let one PA (Table 3) be in a very permissive and progressive environment, where legislation is not restrictive but encourages interoperability and documents have to be written in several different languages. Let the other PA (Table 4) be in a preserving and nationally focussed environment, where openness and interoperability are not considered necessary and only the national language is needed for documents.

In the prioritisation tables for these two exemplified PAs (Tables 3 and 4) the scores are used as follows: "0" does not apply (left usually blank), "1" might apply, "2" does apply. The score for a category is the average of the scores for the criteria within this category. The scores of each single requirements for a category are only indicative and they do not reflect the true values. The requirements "Enables quality presentations to be produced", "To provide word processing, presentation, spreadsheet & database functionality", Operates in a thin client environment", "Automation possibility" and "Integrates with a single LDAP sign

on server” have not been considered in the prioritisation since these usually are very specific to special PAs or are inherent to a text processing program.

	Weights	Localised spell-check	Localised thesaurus	Localised hyphenation	Localised graphical user interface	Supports multiple languages	Provides document layout style guides	Enables document/data source linkage	Supports document composition	Supports Multiple file formats	Enables data conversion from one file format to another	Export to PDF format	Supports XML Data import/export	Import/export from/to open and legacy formats	Functions defined in ECDL Syllabus 4.	Supports user developed macros	Provides users customisable menu options	Provides good user help	Supports document editing with revision marks	Graphically editing HTML documents	
Legal and Political	2																				
Legal criteria	33	2	2	2	2	2		1		2	1	2	2	2	1				1		
Reducing Licenses Costs	33					1				2	2	2	2	2		1			1		
Improving Data protection	33									1	1	1		1							
		0,67	0,67	0,67	0,67	1,00	0,00	0,33	0,00	1,67	1,33	1,67	1,33	1,67	0,33	0,33	0,00	0,00	0,67	0,00	
Technical	2																				
Improving Security	14																				
Improving Portability / Compatibility	14							1		2	2	2	2	2							
Reduce Life cycle costs	14									1	1	1	1	1					2		
Functionality / Features	14	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1	2	2
Improving Integration	14							1		2	2	2	2	2							
Stability / Availability	14																				
Scalability	14																				
		0,33	0,33	0,33	0,33	0,33	0,17	0,67	0,33	1,00	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,17	0,33	0,33	
Strategic	2																				
Improving Openess	14									2	2	1	2	2							
Usage of existing Standards	14									2	2	2	2	2	1		1			2	
Reducing Manufacturer Dependency	14									2	2	2	2	2	2		1	1			
Future existence	14									1	1	1	1	1	1						
Accessibility of used standards	14									2	2	2	2	2	2						
Strategic Plans	14	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2			1	1	
Improving Competition	14						1	1	1	2	2	2	2	2	2	2	1	1		2	
		0,33	0,33	0,33	0,33	0,33	0,49	0,33	0,33	1,95	1,95	1,79	1,95	1,95	1,47	0,16	0,49	0,16	0,16	0,81	
Documentation / Training / Support	2																				
Availability of Support	33																				
Availability of Documentation	33																		2		
Availability of Training	33														1						
		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,00	0,00	0,67	0,00	0,00	
Usability / Accessibility	1																				
Possibility of Localisation	16	2	2	2	2	2															
Access Features	16																				
Usability of GUI	16				2										1	2	2				
User Assistance	16																	2			
Overview over features	16																				
Resistance to erroneous input	16																				
		0,33	0,33	0,33	0,67	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,17	0,33	0,33	0,33	0,00	0,00	
Other	1																				
Avoiding New development	25									1	1		1	2		1	1				
Low Internal Support Effort	25				1													2			
Versioning / Change Management	25											1								2	
Importing of different Standards	25									2			2	2						1	
		0,00	0,00	0,00	0,00	0,31	0,00	0,00	0,00	1,31	0,84	0,84	1,31	1,54	0,00	0,54	0,23	0,00	0,77	0,23	
Total		0,33	0,33	0,33	0,59	0,35	0,03	0,06	0,03	0,25	0,22	0,23	0,24	0,26	0,25	0,31	0,31	0,31	0,07	0,06	

Table 3: Prioritisation for progressive PA

	Weights	Localised spell-check	Localised thesaurus	Localised hyphenation	Localised graphical user interface	Supports multiple languages	Provides document layout style guides	Enables document/data source linkage	Supports document composition	Supports Multiple file formats	Enables data conversion from one file format to another	Export to PDF format	Supports XML Data import/export	Import/export from/to open and legacy formats	Functions defined in ECDL Syllabus 4.	Supports user developed macros	Provides users customisable menu options	Provides good user help	Supports document editing with revision marks	Graphically editing HTML documents	
Legal and Political	2																				
Legal criteria	33	0	0	0	0	0		1		1	1	1	1	1	1					1	
Reducing Licenses Costs	33					1				2	2	2	2	2	2		1			1	
Improving Data protection	33									1	1	1		1							
		0,00	0,00	0,00	0,00	0,33	0,00	0,33	0,00	1,33	1,33	1,33	1,00	1,33	0,33	0,33	0,00	0,00	0,67	0,00	
Technical	2																				
Improving Security	14																				
Improving Portability / Compatibility	14							1		2	2	2	2	2							
Reduce Life cycle costs	14									1	1	1	1	1					2		
Functionality / Features	14	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1	2	2
Improving Integration	14							1		2	2	2	2	2	2						
Stability / Availability	14																				
Scalability	14																				
		0,33	0,33	0,33	0,33	0,33	0,17	0,67	0,33	1,00	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,17	0,33	0,33	
Strategic	2																				
Improving Openness	14									2	2	1	2	2							
Usage of existing Standards	14									2	2	2	2	2	1		1			2	
Reducing Manufacturer Dependency	14									2	2	2	2	2	2		1	1			
Future existence	14									1	1	1	1	1	1	1					
Accessibility of used standards	14									2	2	2	2	2	2						
Strategic Plans	14	0	0	0	0	0	1	1	1	0	0	1	0	0	1				1	1	
Improving Competition	14						1	1	1	2	2	2	2	2	2	1	1			2	
		0,00	0,00	0,00	0,00	0,00	0,33	0,33	0,33	1,63	1,63	1,63	1,63	1,63	1,30	0,16	0,49	0,16	0,16	0,81	
Documentation / Training / Support	2																				
Availability of Support	33																				
Availability of Documentation	33																		2		
Availability of Training	33														1						
		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,00	0,00	0,67	0,00	0,00	
Usability / Accessibility	1																				
Possibility of Localisation	16	2	2	2	2	2															
Access Features	16																				
Usability of GUI	16				2										1	2	2				
User Assistance	16																		2		
Overview over features	16																				
Resistance to erroneous input	16																				
		0,33	0,33	0,33	0,67	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,17	0,33	0,33	0,33	0,00	0,00	
Other	1																				
Avoiding New development	25									1	1		1	2		1	1				
Low Internal Support Effort	25				1														2		
Versioning / Change Management	25											1								2	
Importing of different Standards	25									2			2	2						1	
		0,00	0,00	0,00	0,00	0,31	0,00	0,00	0,00	1,31	0,84	0,84	1,31	1,54	0,00	0,54	0,23	0,00	0,77	0,23	
Total		0,28	0,28	0,28	0,54	0,30	0,02	0,06	0,03	0,22	0,21	0,21	0,20	0,23	0,24	0,31	0,31	0,31	0,07	0,06	

Table 4: Prioritisation for preserving PA (red are changes to the progressive PA)

The values in the table for the preserving PA (Table 4) is using the same values as the one for the progressive (Table 3) with some changes, which are marked in red. For space reasons, the columns and rows are exchanged compared to the example table Table 1.

As last note it is important to say that not all requirements score in each category. Best example is the category “Documentation / Training / Support”. Here only one requirement (“ECDL⁴ syllabus”) concerns this topic.

4 European Computer Driving License [8]

The prioritised list of the requirements are given in Table 5. As can be easily seen, the background used in the prioritisation has strong influence on the order of the requirements.

progressive PA		preserving PA	
Localised graphical user interface	0,59	0,54	Localised graphical user interface
Supports multiple languages	0,35	0,31	Supports user developed macros
Localised spell-check	0,33	0,31	Provides good user help
Localised hyphenation	0,33	0,31	Provides users customisable menu options
Localised thesaurus	0,33	0,3	Supports multiple languages
Supports user developed macros	0,31	0,28	Localised thesaurus
Provides good user help	0,31	0,28	Localised spell-check
Provides users customisable menu options	0,31	0,28	Localised hyphenation
Import/export from/to open and legacy formats	0,26	0,24	Functions defined in ECDL Syllabus 4.
Supports Multiple file formats	0,25	0,23	Import/export from/to open and legacy formats
Functions defined in ECDL Syllabus 4.	0,25	0,22	Supports Multiple file formats
Supports XML Data import/export	0,24	0,21	Export to PDF format
Export to PDF format	0,23	0,21	Enables data conversion from one file format to another
Enables data conversion from one file format to another	0,22	0,2	Supports XML Data import/export
Supports document editing with revision marks	0,07	0,07	Supports document editing with revision marks
Enables document/data source linkage	0,06	0,06	Enables document/data source linkage
Graphically editing HTML documents	0,06	0,06	Graphically editing HTML documents
Supports document composition	0,03	0,03	Supports document composition
Provides document layout style guides	0,03	0,02	Provides document layout style guides

Table 5: *Prioritised List of Requirements*

10 Conclusions

The method we have presented is a means to prioritise software or data standards requirements by importance. The requirements are collected from the PAs and verified for quality and standards compliance. These standards have been chosen in accordance to the motivation for the prioritisation of the software or data standards requirements. If, for example, the goal is to obtain an overall picture of the needs of a set of PAs the software or data standards requirements format should not be detailed too much.

The PAs are then asked to score each requirement by accordance in terms of specified categories and criteria. A mathematical procedure then provides the means to compute a total score from the single ones defined for each criterion. The procedure may vary with the motivations of the prioritisation. In the case above a Cost-Benefit-Analysis may be the most appropriate.

The requirements are then ranked by the total score.

The list of prioritised software or data standards requirements is then the base for a decision making process for software purchase or update. In the COSPA project the method may be adopted in the experimental phase for a feasibility analysis of a transition to a new software solution (for example toward an Open Source solution).

This document also shows the importance of the environment of the PA resulting in different prioritised lists.

The deliverable provides theory and examples supporting the method description. In the appendix E the method provides a description of a possible approach to reduce subjectivity of the scoring procedure.

Appendix A - Abbreviations

AAU	Ålborg University, Danmark
CO	Connecta, Italy
Dx.y	Deliverable x.y
ECDL	European Computer Driving License
FLOSS	Free/Libre/OpenSource Software
FUB	Free University of Bolzano, Italy
HTML	HyperText Markup Language
IDA	Interchange of Data between Administrations
LDAP	Lightweight Directory Access Protocol
ODS	Open Data Standard
OFNWA	ordered Fuzzy Number Weighted Analysis
OS	Open Source
OSOSS	Open Standards in Open Source Software
OSS	Open Source Software
OWA	Ordered Weighted Analysis
PA	Public Administration
PDF	Portable Document Format
PP	Province of Pisa, Italy
SWRA	South West Regional Authority, Ireland
USFD	University of Sheffield, United Kingdom
WP	Work Package
XML	eXtensible Markup Language

Appendix B - Glossary

A list of words used in a specific meaning throughout this document:

Method	Referring to a technique of solving a general task by a structured approach.
Procedure	The technique (mathematical) for carrying out the prioritisation
Technique	An established process / formula for solving a small task.
Criterion	A simple term to compare the requirements to.
Category	Discerning a group of similar criteria.
Requirement	A term describing a feature desired within a software.
Score	A value assigned to a requirement with respect to a criterion.

Appendix C - Software Groups

The deliverable D2.4 has determined the following software groups:

Software Type	Example Software
PA Management	software for internal administration, like payroll, purchase, accounting, etc.
Administration and Services	software for offering the administrative tasks of the PA to the citizen like electoral registration, tax management, social services
Culture, Education, Tourism and Leisure	software to publicise public facilities inside the PAs area
Environment, Development and Planning Control	software for managing the administration of infrastructural elements like roads, electricity, water, parks
Information Technology	Software for managing the IT infrastructure of the PA

Appendix D - Categorical List of Criteria

Table 6 shows the categories and associated criteria which have been found.

Categories	Criteria
Legal and Political (licenses, regulations, ...)	<ul style="list-style-type: none"> • Legal criteria • Licenses (gif) / Costs • Political reasons • Data protection • Register of software in use (no. of copies / licenses, users, ...)
Technical (security, ...)	<ul style="list-style-type: none"> • Security • Portability / Compatibility • Life cycle / Life cycle costs⁵ • Functionality / Features • Integration • Stability / Availability • Scalability
Strategic (vendor dependency, development plans, ...)	<ul style="list-style-type: none"> • Openness (closed / open software) • Usage of existing standards • Manufacturer Dependency • Future existence / accessibility of used standards • Strategic Plans • Competition
Documentation, Training, Support	<ul style="list-style-type: none"> • Support • Documentation • Training (Cost, Available Material)
Usability and Accessibility (localisation, ...)	<ul style="list-style-type: none"> • Localisation • Access features for handicapped • Usability of GUI • Appropriate (unneeded features, menus) • User Assistance / Guidance by the software • Overview over features • Resistance to erroneous input
Other	<ul style="list-style-type: none"> • New development / existing version (debugged) • New product is used in several administrative business processes • Men Power (internal support, productivity) • Versioning / Change Management • Import of other Standards

Table 6: Category and Criteria List

⁵ Life cycle costs consist not only of the costs to acquire a software but also for costs for its maintenance. These maintenance costs are split according to a Gartner Group study from 1996 into: 43% end user operations (education, personal support, data management), 27% technical support (client support - installation, configuration, removal -, 1st and 2nd level support, systems management, disk management, security and virus management, software distribution), 21% capital costs (hardware, software, infrastructure), 9% administration (asset management, audits/review, security, policies and methods). Further reading in Deliverable 3.1

Appendix E - Fuzzy Cost-Benefit-Analysis

We have seen that a traditional cost-benefit-analysis has a certain degree of subjectivity due to the Developers assigning the weights to different categories and / or criteria. This subjectivity acting on the total score s from (8.2), can lead to different results of prioritisation with different priority orders within the requirement set. In this appendix we introduce two techniques to control this subjectivity: the variation of the weights and the fuzzy logic. The former controls the subjectivity due to the Developers' assignment of the weights, the latter the subjectivity in those measures that have nominal scale – like the Likert scale {very low, low, medium, high, very high}.

Appendix E.1 -Weights Variation

The total and the partial scores (8.1 and 8.2) depend each of one set of weights assigned subjectively by the Developer. Varying the corresponding weights, the scores varies. For simplicity we focus our discussion of the total score, but everything in the following may be translated for any partial score.

Comparing total score trends is comparing the score independently of the weights. In this section we analyse the score trend due to the variation of the weights.

To graphically represent and discuss such trends, it is typical to aggregate the weights into a unique variable. The most common aggregation formula is the orness:

$$\sigma = \frac{1}{(n-1)} \sum_{i=1}^{n-1} ((n-i)w_i) \quad (D.1)$$

Assuming the weights $w_i \in [0,1]$, the orness is defined in the interval of values $[0,1]$ and is a function of the weights. The total score s is a function of the weights and the judgements w_i (see also Chapter 8). We want to express it in terms of σ and the judgements. Had it been immediate if to one value of orness corresponds only one set of weights. Unfortunately this does not hold in general: for example, $[1/2, 1/4, 1/20, 3/20, 1/20]$ and $[0, 1, 0, 0, 0]$ have the same orness value $3/4$. To get this one to one correspondence we proceed in the following way:

- We define the score at the k -th level $s_k = s(w_1, \dots, w_k)$, which considers only the first k categories
- We define the orness at the k -th level of aggregation

$$\sigma_k = \frac{1}{(k-1)} \cdot \sum_{i=1}^{k-1} (k-i) \cdot w_i \quad k=2, \dots, n \quad (D.2)$$

Note that $\sigma_n = \sigma$

- Forming a Basic Rule of Linear Algebra⁶ we prove that the weights w_i are in one to one correspondence with σ_k
- We set $\sigma_k = c_i \cdot \kappa$ with $0 \leq \kappa \leq 1$ for $j=2, \dots, n$
- Computing the w_i in terms of σ_k we express each $w_i \quad i=1, \dots, n-1$ as a multiple

⁶ Basic Rule of Algebra: A check whether the determinant of the transformation matrix is zero, i.e. the transformation from w_i to σ_k .

of κ . With $w_n = 1 - \sum_{i=1}^{n-1} w_i$ the total score s becomes a function of κ (not linear in κ) and the judgements w_i . As $\sigma_{n-1} = \sigma \equiv \kappa$ s is a function of σ ⁷

- For fixed values of the judgements varying κ , the total orness σ and the total score s vary continuously and plot in a graph of a two dimensional plane: the decision graph

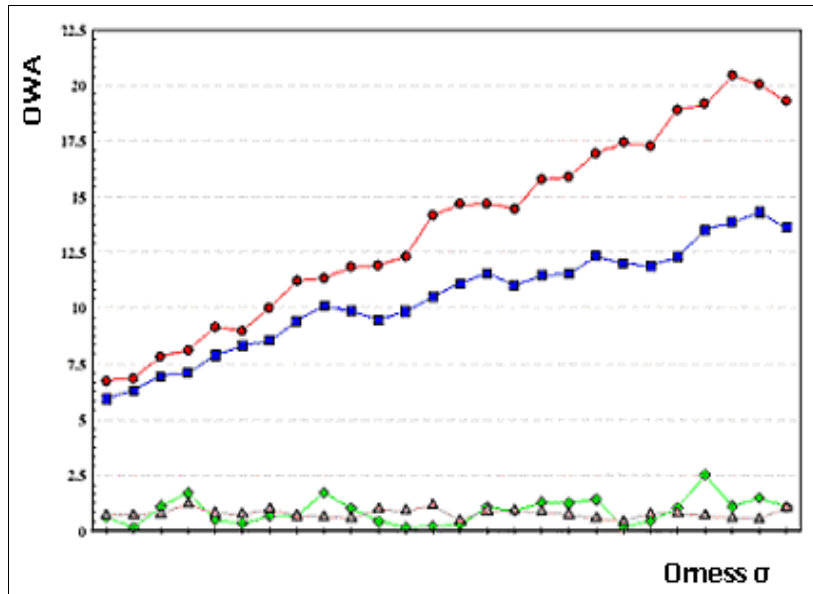


Figure 3: Decision graphs

Figure 3 displays four decision graphs each corresponding to the trend of a total score of a requirement. We can compare the four requirements in the following way:

The requirement represented by the line dotted with circles (top line) has for any value of orness the highest value of OWA among the four decision graphs. Therefore the requirement has always the highest priority, for each choice of orness level.

The requirement represented by the line dotted with squares is always second in priority for any level of orness

The requirements represented by the line dotted with triangles and dotted with diamonds have no strong difference in priority. Though they both have always less priority than the top ones.

The orness is not only a sole trick to display the total score in a two dimensional plane, rather is a critical index expressing the optimism of the Developer.

7 There may be some constraints due to the formula $\sum_{i=1}^n w_i = 1$ see footnote 7 in the following example

To understand how orness expresses the Developer's optimism we need to modify slightly the regular formula (8.2 and 8.3) for the total (and partial) score: this is the Ordered Weighted Averaging (OWA) [9]. OWA is computed according to a modification of the formula (8.2):

$$s = \frac{\sum_i w_i s_{\phi_i}}{\sum_i w_i} \quad (D.3)$$

where $[\dots, s_{\phi_i}, \dots]$ is the permutation of the scores $[s_1, s_2, \dots, s_n]$ defined by ordering them by their decreasing value.

By definition (D.1) the orness emphasizes scores with low index rather than scores with high index. Therefore the values of the weights in an OWA determine the relevance of the partial scores in terms of their values. For example:

1. To get the most pessimistic attitude take $\sigma=0$, then the weight combination is $w_*=[0, \dots, 0, 1]$ and therefore $s=s_{\text{MIN}}$ where s_{MIN} is the partial score in $[s_1, s_2, \dots, s_n]$ with lowest value.
2. If we weight all the categories evenly we have a neutral attitude towards the categories, meaning $\sigma=1/2$. Therefore the weight combination is $w_{\text{ave}}=[1/n, \dots, 1/n, 1/n]$ and all the categories have the same relevance to the Developer.
3. To get the most optimistic attitude take $\sigma=1$. Then the weight combination is $w^*=[1, 0, \dots, 0]$ and therefore $s=s_{\text{MAX}}$ where s_{MAX} is the partial score in $[s_1, s_2, \dots, s_n]$ with highest value.

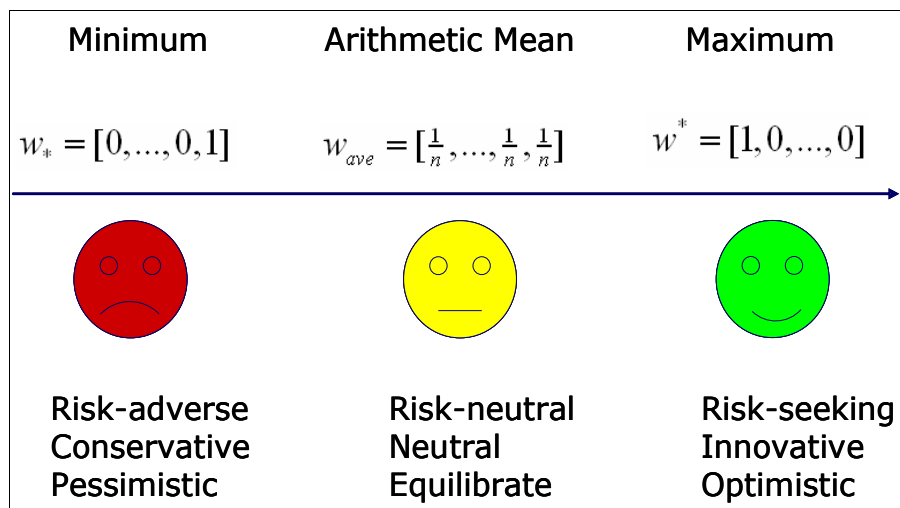


Figure 4: Orness as a measure of the evaluators' optimism

Interpreting the orness as measurement of optimism allows us to extend the interpretation of the mutual position of the decision graphs in the plane (Figure 3). Here in the right hand side, the line dotted with diamonds is always above the line dotted with triangles. This means that when the Developer is optimistic the requirement defined by the line dotted with diamonds has higher priority than the requirement defined by the line dotted with triangles.

Example: Let $2/3, 0, 1/3$ be a set of partial scores. The ordered triplet of scores s_{ϕ} is

$2/3, 1/3, 0$. Furthermore, from (D.1) $\sigma = w_1 + \frac{1}{2}w_2$ and from (D.2) $\sigma_1 = w_1$. If we set $\sigma = \kappa$ and from (D.2) $\sigma_1 = \frac{\kappa}{2}$, since $w_1 + w_2 + w_3 = 1$ by a simple rule of linear algebra, we have

$$w_1 = \frac{\kappa}{2}, \quad w_2 = \kappa, \quad w_3 = 1 - \frac{3}{2} \cdot \kappa$$

Now we vary κ from 0 to 1. To display a line we choose 8 values of κ , namely $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 1\}$.

The resulting total scores are given in Table 7 and plotted in Figure 5

Orness $\sigma \equiv \kappa$	Intermediate Orness σ_1	Weights	Total Score
0	0	[0, 0, 1]	0
0,1	0,05	[0.05, 0.1, 0.85]	0,07
0,2	0,1	[0.1, 0.2, 0.7]	0,13
0,3	0,15	[0.15, 0.3, 0.55]	0,2
0,4	0,2	[0.2, 0.4, 0.4]	0,27
0,5	0,25	[0.25, 0.5, 0.25]	0,33
1 ⁸	0	[0, 1, 0]	0,33

Table 7: Example variation of weights for the fuzzy Cost-Benefit-Analysis.

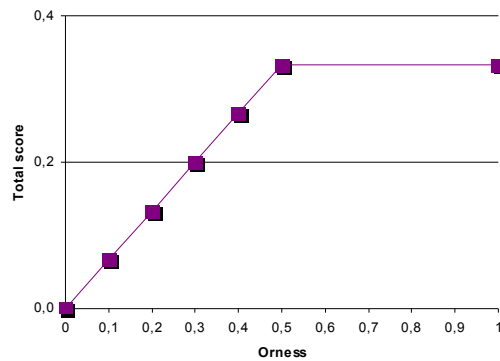


Figure 5: Total score of a requirement

Appendix E.2 -Using Fuzzy Scores

The theory of fuzzy sets has been introduced to reduce the subjectivity of measures with nominal scale.

⁸ Note: since $w_3 = 1 - \frac{3}{2} \cdot \kappa > 0$ follows $\kappa < \frac{2}{3}$, for $\kappa = 1$ we need to choose $[0, 1, 0]$

Assigning discrete numbers for the different criteria has two main disadvantages. The first is – as already mentioned – the problem of the subjectiveness of the assignment of the scores toward each criterion. The second is that the amount of accordance of a criterion towards a requirement might not be expressible by the given set of discrete numbers (e.g. allowed values are 0 (no) and 1 (yes) but the answer has to be “maybe”).

Note since $w_3 = 1 - \frac{3}{2} \cdot k > 0$ then $k < \frac{2}{3}$, for $k = 1$ we need to choose $[0, 1, 0]$ To overcome these problems one replaces crisp numbers with fuzzy numbers as values of a measure.

A fuzzy number is not a number in the usual sense but can be represented by a function, called membership function. An example for fuzzy numbers defining nominal values such as “low”, “medium” and “high” is shown in Figure 6.

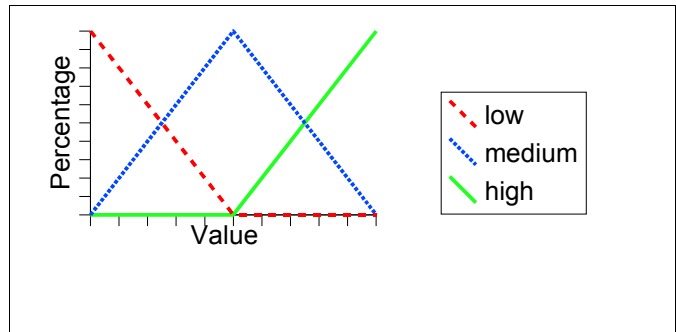


Figure 6: Fuzzy Numbers for "low", "medium" and "high".

As can be seen from Figure 6 triangular fuzzy numbers are represented by a lower boundary, i.e. the smallest value associated with the figure, the centre value, and an upper boundary. So each fuzzy number can be represented by a triple of numbers, denoting these points of the representing function. $[s_{lb}, s_c, s_{ub}] \equiv \hat{s}()$

E.G. if we say, a person is tall, this is unlikely to be smaller than 1.70 m, probably around 1.90 m and perhaps up to 2.10 m. So our fuzzy number for “tall” would be a triangular function starting at 1.70 m, with its maximum at 1.90 m and ending at 2.10 m.

The formulas for the cost-benefit-analysis now become using fuzzy numbers:

$$\hat{s}_i() = \frac{\sum_j w_{ij} \hat{s}_{ij}()}{\sum_j w_{ij}} \quad (8.1')$$

$$\hat{s}() = \frac{\sum_i w_i \hat{s}_i()}{\sum_i w_i} \quad (8.2')$$

$$\hat{s}() = \prod_i \hat{s}_i() \quad (8.3')$$

The sum and the product of fuzzy numbers are themselves fuzzy numbers, i.e. they are represented by a function. Sum and multiplication are defined similarly to the set operation AND (intersection) and OR (sum) [10] taking respectively the minimum value and the maximum value of the membership function.

The usage of fuzzy numbers and the variation of weights from the previous chapter together is described by the Ordered Fuzzy Number Weighted Averaging (OFNWA) theory [11].

The result of the combination is not a simple line as for pure OWA any more, rather an area. These can be represented by their lower and upper border and their mean value, as shown in Figure 7.

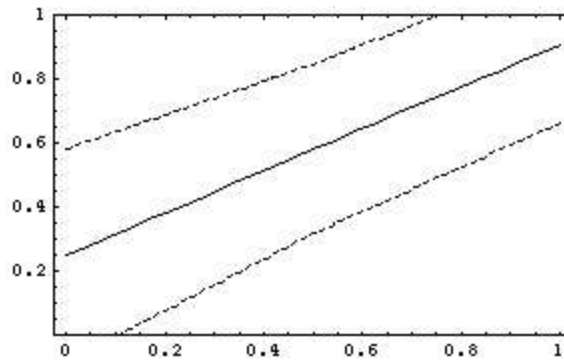


Figure 7: *Decision Graph with fuzzy numbers.*

Requirements are then compared through their respective area or border.

Appendix F - Questionnaire for Requirements

(excerpt from Deliverable D2.4)

Name of Organisation: USFD		Date: 21th May 2004
Data standard for: Curriculum vitae		
Criticality (1 – 5 ; 1 is unessential and 5 is critical): 1		
Brief description of the standard: the standard needs to provide a mark-up for curricula vitae.		
Constraints, assumptions and dependencies: Must be available for any operating system. The standard must be based on XML.		
Functional requirements:		
Description	Priority:	
Allows to define name(s) and surname	Mandatory	
Allows to define contact information	Mandatory	
Allows to define nationalities	Mandatory	
Allows to define the countries/regions where the subject is allowed to work	Desirable	
Allows to define date of birth	Mandatory	
Allows to define gender	Mandatory	
Allows to define work experience	Mandatory	
Allows to define education and training	Mandatory	
Allows to define mother tongue(s)	Mandatory	
Allows to define knowledge of foreign languages	Mandatory	
Allows to define social skills and competences	Optional	
Allows to define organisational skills and competences	Optional	
Allows to define technical skills and competences	Optional	
Allows to define artistic skills and competences	Optional	
Allows to define special skills and competences	Optional	
Allows to define driving licence type and status	Optional	
Allows additional information to be provided	Optional	

Appendix G - References

- 1 muenchen.de (2004). *LiMux – Die IT Evolution*. Checked on 08/03/2005 from <http://www.muenchen.de/Rathaus/referate/dir/limux/89256/index.html>
- 2 Infonomics (2002). *FLOSS Report*. Checked on 08/03/2005 from <http://www.infonomics.nl/FLOSS/report>
- 3 OSOSS (2004). *Programma Open Standaarden en Open Source Software voor de overheid*. Checked on 08/03/2005 from <http://www.ososs.nl>
- 4 IDABC (2001). *Available documentation on Open Source Software*. Checked on 08/03/2005 from <http://europa.eu.int/idabc/en/document/2623#study>
- 5 Berlecon Research (2004). *Linux- und Open-Source-Strategien für CIOs*. Checked on 08/03/2005 from http://www.berlecon.de/output/en/studien.php?we_objectID=157
- 6 Forrester (2004). *The Linux Tipping Point*. Checked on 08/03/2005 from <http://www.forrester.com/ER/Research/Report/Summary/0,1338,16288,FF.html>
- 7 N. Fenton, S. L. Pfleeger (1996). *Software Metrics: A Rigorous and Practical Approach*. International Thomson Computer Press, London
- 8 The ECDL Foundation Limited (1997). *European Computer Driving License Foundation Home Page*. Checked on 08/03/2005 from <http://www.ecdl.com/>
- 9 R. R. Yager (1988) On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transaction on Systems, Man and Cybernetics*, 18 (1), 183. Springer-Verlag Berlin
- 10 W. Pedrycz and F. Gomide (1998). *An introduction to fuzzy sets: Analysis and Design*. The MIT Press
- 11 G. Canfora, L. Cerulo, R. Preziosi, L. Troiano (2003). A tool for Decision Support implementing OFNWA approach: a case study. *Proceeding of Software Engineering and Knowledge Engineering Conference*, San Francisco