This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

BAREFOOT – EVALUATION CRITERIA FOR SOFTWARE AND HARDWARE SELECTION

WP N° 2	DISSEMINATION
WP LEADER	ONE FAMILY
ANNEXES	1 COMPLETED EXAMPLE USE OF MATRIX
LAST MODIFIED	28/01/2012

Summary

1. Introduction

One of the general goals in the Barefoot project, is to stimulate VET trainers to improve and enhance their use of teaching/learning ICT skills, in particular for those with low ICT skills or working "in the filed" or in socio-economic disadvantaged training contexts where ICT facilities are not widely known and/or used. In order to realise that general objective, the project intends to create a guide about software categories highlighting particular features enabling them to be used by these trainers. For instance free or low cost software, easy-to-use software, accessible software, etc. The key point of the guide should be the definition of the trainers' needs in respect of their ICT skills. This means that the guide we provide should not be a general "yellow-page list of existing software", but a tool supporting trainers in software selection, driving them through the software categories according to their "low-profile ICT skill" or any other specific need. Therefore, this document represents a preliminary work on the software evaluation criteria and checklist.

2. Software categories

Before defining the criteria for software evaluation and selection it is necessary to specify how this guide should be used. The aims of that guide isn't the advertisement or underpinning of any software house or brand or specific software solution/application; rather the idea is to provide a classification of software into categories that could be useful for trainers when they arrange and deliver their training sessions. Then, for each category a short list of most popular and used software according to the evaluation criteria we defined will be provided. For instance most popular free or open source (criteria) software for presentation by means of slides (category).

We considered the following main categories:

- Multimedia content production (video, picture, screencast, etc.)
- Textual content production (word processor, spreadsheet, slide, web site, etc.)
- Content management (document management system, databases, repositories, etc.)
- Online interaction (social networks, forums, mailing lists, blogs, etc.)

In addition to the above main categories, we introduce two ones which can be considered across the others[1]:

- Basic/System-level utilities (antivirus, search engine, backup, data compression, etc.)
- Online services & utilities (encyclopedia, file sharing, video sharing, web site / forum / blog hosting, etc.)

[1] The point here is that if we consider those people (trainers) have no habits to work using computer we should also suppose they are not so familiar with basic software tools as well. Our guide should not just list software categories and related entries, but should also encompass guidelines on how and when use each type of software in the VET context.

These cross-categories are intended as collection of tools and utilities, basically delivered as online services or small piece of software to be installed locally, that are very easy to use, most of the times are free or very inexpensive, and the kind of service they provide are somehow related to one or more of the above categories (for that are across). For instance in the first category we can include antivirus, firewall, search engines and many other very common tools needed by a trainer if he/she start to use the computer in a very intensive way for his job. In the second category we can consider the online free hosting service for blog, forum or web site, the online sharing of slides for presentation, the music or video sharing channels, etc. There is no overlapping between these items and those in the first category (Multimedia) because the first category will list software useful to *create* or *edit* videos, meanwhile the cross-category will list online service useful to *share* and *deploy* videos.

Follows a detailed description of each category.

a) Multimedia content production

Nowadays, many e-learning platforms and tools provide interactive and multimedia features so to enable a plurality of media to coexist and contribute to an original lesson, rich of text, sounds, video or any other media of "electronic communication". For that there have been developed thousands of tools enabling users to produce their own multimedia pieces without the need to be an expert of production technology. Just to make an example, high quality movies can be done with a low cost camera and then saved into the computer just using a USB cable between the camera and the computer. No further complex and expensive technology is required. On the other hand, the digital manipulation of such multimedia material can be done at a basic level with easy-to-use and low cost or free software we are going to list in this category. The subcategories here are:

- Video recording and editing
- Picture editing
- Sound recording and editing
- Screen recording and video production
- Icon and image production/editing
- Animation production
- Concept/Mind mapping software (knowledge representation)

b) Textual content production

This is probably the most popular software category, that includes the widely and well-known tools used to create and editing documents in some common shapes like document, spreadsheet and so on. Even if the modern e-learning platforms and technologies make a wide use of different media the main part of educational content still are in a textual format. The subcategories here are:

- Word processing
- Spreadsheet
- Presentation by slides
- HTML editor

- PDF - related
- Wiki software
- Assessment (for instance survey and quiz creators)

c) Content management

When a trainer starts to use e-learning methodologies, they will discover that one the main issue is the increasing number of material produced, lesson after lesson. This is one of the nice aspects of a learning community based using modern technology: most of the time there is no need to print lessons so you can produce a large amount of~~dozen o~~f e-learning material~~stuff~~. As a natural consequence, there is the need to keep order in that increasing amount of material, and for that content management systems became a must. The subcategories here are:

- Document repository and management
- Versioning systems
- Database systems

d) Online interaction

The web provides many online interaction tools that can be used jointly with other software, by each member of a~~n~~ VET community, either trainers or students, in order to support the training activities and the social aspect of the community itself (for instance the relationship among learners and between learners and trainers). The subcategories here are:

- E-learning platform (with classroom utilities like chat, forum, emails, etc.)
- Social networks
- Blogs
- Forums
- VOIP phone calls
- Live chat
- Web meeting and conferencing

e) Basic/System-level utilities

Elevating the use of a computer from personal purpose to a professional activity implies that the machine should be kept in the right order more than before. In other words, when the computer becomes the tool by mean of which the trainer designs, develops and deploys his/her lessons, it is ~~better if the~~more important that the machine has no virus, ~~keeps~~ files are kept in the right places, has the best performance, and so on. Moreover, the aim of this document is also to provide some guidelines for trainers about how to use the computer and more in general the modern ICT in their VET activities and this means they should have a good knowledge of how to maintain a computer too. The subcategories here are:

- Security (antivirus, anti-spam, firewall, etc.)
- Storage (backup, data recovery, compression, disk analysis, disk optimization, etc.,)
- Search engine

- System utilities (performance optimization, driver recovery, etc.)

f) Online services & utilities

The evolution of the ICT and in particular the web has introduced many new ways to deal with issues of the everyday life. Sharing a picture with friends all around the word, making a video conference with colleagues on the other side of the Country, sharing a document with remote learners are all actions unthinkable just few years ago. However, even if these services seem ~~s~~ to be a well-known realty, they are just a pipe-dream for many professionals. So, this category intends to provide useful tricks for trainers in order to improve the quality of the training they provide by using the computer. The subcategories here are:

- Encyclopedia
- File sharing
- Video sharing
- Hosting (web site, forum, blog, etc.)
- Survey and quiz
- Online guides, models/templates and training material (for instance guide on how to work on HTML, how to create an impressive slide presentation, spreadsheet models, and so on.)

3. Evaluation criteria

About the software quality criteria there are many documents and standards at international level, main of which deal with technical aspects related to the production of software (software engineering). But, here we are dealing with evaluation criteria from the users' point of view, and ~~from a specific users~~from specific users, e.g. trainers as intended with the Barefoot's targets group. This is sometimes called "quality in use". For that, we consider useful a preliminary clarification about the evaluation criteria described in the literature of ~~the~~ software engineering. Th~~is ey~~ basically refers~~s~~ to the matching of design features and the final product's characteristics, and BAREFOOT is not considering this~~we are not interested into that~~. In other words, structural quality and functional quality are two lines of evaluation aimed to measure how much the software matches its design features or the user's outlooks. This has nothing to do with how much the software could be useful for a trainer in his/her training activity and how much effort~~s~~ is required by the trainer in order to use that software. Therefore, the most important evaluation criteria is the balance between cost and benefit where cost means not just money. Evaluation criteria like reliability (an attribute of resiliency and structural solidity), efficiency (attribute that ensures high performance once the application is in run-time mode), security (the risk of encountering critical vulnerabilities that damage the business), maintainability (also includes the notion of adaptability, portability and transferability, all features strongly related to the development issues) and some others, rightly considered in the software engineering, are here not considered at all or, in some cases, are examined from a different perspective. Moreover, considering the specific Barefoot's targets, that are VET trainers having "low-profile ICT skill", we should **encourage** the use of the technology and not **frighten** them with complex things they will consider too far from their

own knowledge and skills. That means that another important factor we should put in evidence in the evaluation criteria is the impact that the use of software can have on the trainer's activity.

Having in mind all the preliminary aspects mentioned above, we suggest to break the evaluation criteria down in two groups: **options**, those elements that might have multiple or subjective interpretation (from the user's point of view); **attributes**, those elements that are more a fact than an option, and the user can accept or not but cannot modify. In both cases we will include a list of possible impact on the user (the trainer) according to some initial conditions we can suppose. A first draft of evaluation criteria is shown below[2], with a brief description of each one.

- **Options**:
  - Usability. How easy is to learn and operate with the software? An extended set of attributes that bear on the effort needed for use could be: understandability, learnability, operability, attractiveness. In other words a software is usable if the end user can invest an *adequate* amount of time and efforts in order to be comfortable in using that software for his/her needs.
  - Productivity. Even considering an expert user, how many efforts are needed in order to produce outputs. Even measured as the level of effectiveness achieved against the efforts provided.
  - Level of integration/interoperability[3]. What is the standard of the output and its level on integration into other software application. For instance, a slide presentation tool can save a file format recognized by the most common e-learning platforms? A video format can be viewed with the most common player?
  - Accessibility. Does the software provide different environment setting so to be accessible to people with different kind of special needs? For instance high contrast color schema, dimensions of characters, and so on.
  - Level of portability[4]. Is the software available for different operating systems and/or platforms? Does it require any surrounding technology or software?
  - Level of documentation/basic support . What's the level of documentation/support available? Is there just an "Help" section into the interface, or are there many document or even books available in the marketplace? Is there an international community of users, a forum, mailing list or any other group of people having experience and keen to provide any help even for free?
  - Level of advanced support/training. Is there an appropriate diffusion of the software so to ensure the availability of advanced support services like training or consulting?

[2] The list has no order, that means the importance of a criterion is not related to its position in the list itself. A further research could also try to define a scale of priorities for criteria, useful for trainers in order to quick check the level of matching between the software features and his/her training needs.

[3] Sometimes in literature also referred as portability.

[4] Here the term is intended as equivalent to transportability or transferability

- Extent of user community. How much the software is known and used? Possible variations could be: extension at geographical level (how many in my Country/Region); extension based on peculiarity of user profile (how many trainers, engineers, high school students, and so on); how many public organizations.

- **Attributes**:
  - Interface language. Is the software interface available in different language or just in one language (most common English and Spanish)?
  - Documentation/support language. Is the software's documentation and/or support (as above described) available in more languages?
  - Type of license (costs). What is the license schema available (freeware, shareware, charityware, etc.)?
  - Type of source code. Proprietary or open source.

4. How to use the evaluation criteria

Once we have a list of evaluation criteria, it is important also to provide guidelines about how to use it. Here we want to propose a possible way to exploit the list of evaluation criteria, that is by means of a matrix that should simplify the overall evaluation, after that the user has analysed each criteria separately. The approach by matrix can be done in many different ways, we here considered two of them, and we suggest applying consecutively during the process of software selection. In more detail the selection process should follow these steps:

1) Define your needs and your constraints
2) Define the software category most appropriate to your needs
3) From the category move on to items: make a selection of candidate software from that category
4) Define the evaluation criteria applicable to your situation
5) Analyse carefully and separately each criteria (for instance by using the first matrix) for each candidate software. Form that analysis in all likelihood some candidates will be rejected
6) Make a benchmark, rating all remained candidate in comparison each to others (for instance by using the second matrix).
7) Make your decision

Following there are two possible implementation for the matrices named above

Matrix 1:

Software name						
Evaluation Criteria	Note	Impact	Remedy	Alternative	Added value	Rating (1-5)
Name of the evaluation criteria	Short note about the software vs. criteria. Normally it should be taken from an external source of information, like Barefoot web site for Barefoot's targets	What (negative) impact could have on my specific situation. If the Note field shows a positive evaluation, this field can be left empty or just "None"	Is there anything I can do to reduce negative impact? Can be left empty or "none" if empty the field Impact	Is there any other software with a lower impact? Can be left empty or "none" if empty the field Remedy	Is there any added value (not yet included in other evaluation criteria). Can be left empty or "none"	A personal global rate according to perception about that criteria applied to that software. 5 is Excellent 4 is Good 3 is Fair 2 is Poor 1 is Bad

Matrix 2:

Software vs Evaluation criteria	EC1	EC2	…	ECn
SW1				
SW2				
…				
SWk				

In each entry put scores from the above columns "rating"