[From: *Translating and the Computer 14.* Papers presented at a conference… 10-11 November 1992 (London: Aslib 1992)]

EVALUATION OF SOFTWARE

J E Rowley, Principal Lecturer
Crewe+Alsager College of Higher Education

ABSTRACT

Two aspects of the selection and evaluation of software packages are reviewed: the strategy for evaluation and selection, and criteria that might be applied in selection. The evaluation and selection of a software package should be approached as a project. Appropriate strategies for the selection and evaluation of software packages can be based on information systems methodologies. The main stages in the project are: the definition of objectives, evaluation of options, definition, selection and design, implementation and evaluation and maintenance. A system requirements specification is an important document in this process. Software selection must be guided by appropriate criteria. General criteria are cost, lifetime and life history, originator, supplier, support, maintenance, technical considerations and compatibility, ease of use, interfaces and integration. In addition specific criteria must be developed for specific categories of packages. A checklist of criteria for database packages is given.

INTRODUCTION

This paper focuses on the selection and evaluation of software packages. There are two main sections to the paper: the strategy or methodology that might be adapted during evaluation, and the criteria that might be applied and in the selection of the most appropriate software package. Both of these issues are addressed from a practical angle. A much more theoretical approach could be taken, especially in the area of information systems methodologies, but this is not felt to be appropriate.

SOFTWARE PACKAGES

The focus of this paper is the selection and evaluation of software packages. This process will normally take place as part of the selection of a complete computer system involving hardware and software, and the strategy for the complete system might follow a similar pattern to that outlined below. Nevertheless, this paper will concentrate on the selection of a software package.

A further assumption underlies the comments made here. This is that for most applications software should be acquired in the form of a pre-written, commercially available software package. It is possible to develop software in-house, although the design and implementation of in-house software definitely requires greater skills of systems analysis, communication, liaison and project management, than the acquisition and implementation of a pre-written package. A package, like all off-the-peg garments, is not tailor-made and cannot be expected to cater for every little idiosyncrasy in every application. Usually this is an advantage; it encourages users to adopt more standard practices and irons out some of their unnecessary habits. On occasions, an application really does demand a tailor-made solution, and here in-house software development may be appropriate. A tried and tested package is likely to be much easier to implement than an in-house package. The package should be robust, and supported by documentation, user groups and other users, training, help desks and maintenance arrangements. Such support is especially crucial for the newly computer literate information manager, but should also be welcomed by all who have to manage a computer system.

A STRATEGY FOR EVALUATION AND SELECTION

Given the range of options when considering a new computer system a strategy for evaluation and selection and for the management of the project is essential. Such a strategy can be exploited both to assist in choosing an appropriate software package, and to design the system that will be created with the software package. Analysis and data gathering associated with both of these activities can evolve in parallel. Any strategy that is adopted will identify a number of stages through which the project must pass. These stages will take time. Very often the time involved in the selection and implementation of a computer system is seriously underestimated leading to late implementation, unfulfilled expectations and other associated hazards. A small system should merit much less planning time than a larger system, but planning remains necessary.

The strategy which is proposed below assumes that the systems project evolves from application to software to hardware. In other words, the requirements of the application are identified, prior to the introduction of constraints that may be posed by hardware and software. In some environments this is an idealised model. Sometimes hardware has already been purchased, or it is desirable to adhere to externally imposed hardware or software standards and this may pose severe constraints on the available software, and, in turn on the applications. Where the opportunity exists to start by a full examination of the application it should not be squandered by premature narrowing of software and hardware choices.

Systems analysts have developed information systems methodologies to assist in the analysis and design of information systems. Whilst these methodologies are probably too detailed to be applied in the selection of a software package there are a number of characteristics of such methodologies that might usefully be integrated into any strategy for systems analysis and design. These are:

1. The methodology specifies a detailed and clear series of phases, sub-phases and steps through which the project must pass.

2. A top-down approach is taken which starts with the development of a broad overview of the system, and then progressively more detail is added.

3. One step in the methodology leads to the next, with increasing refinement.

4. Quality assurance steps that must be completed are positioned throughout the project.

5. Heavy reliance is placed on the systems specification which is drafted early in the project and revised as work proceeds and additional insights are gathered.

6. User consultation is emphasised throughout the project.

7. Checking on perspective by user consultation and the use of a range of tools and techniques.

8. The use of specialist tools and techniques, many of which are graphical to aid in communication and to act as part of the record of the system.

9. Early emphasis on what the system will do, or logical analysis, before the analyst moves on to consider how this should be achieved.

A model for a strategy for the selection and evaluation of software packages is proposed below. Other models may be developed, to suit local circumstances. The specific stages in the strategy are not important, provided all key issues are addressed. The existence of a strategy is crucial to effective communication and management. The strategy proposed comprises the following stages.

Definition of objectives

The first stage must be to define the objectives of the project. This may commence with an analysis of the existing system if this is felt to be an appropriate basis for development. This phase probes why procedures are performed, what improvements are possible and what lends itself to computerisation. A new system must not, however, be hampered by previous practices and it may be more fruitful to perform a needs analysis and to define the objectives of the new system without reference to any earlier systems. A cost benefit analysis may be appropriate, but costs of both the existing and the new system may be difficult to identify in any real sense, and benefits may defy analysis in financial terms. Cost benefit analysis can be a useful tool where costs and benefits can be clearly identified. This stage should culminate in the generation of a short document (the shorter the better!) which could be described as the outline systems specification. This document, which should identify the general objectives of the new system should form a basis for in-house communication, communication with vendors and other systems personnel and act as a reference document for later consultation in that it lays down what the project is intended to achieve. Note that at this stage the emphasis is on the application and not software or hardware. Perceptions and visions will be influenced by prior experience of computer systems, and the extent of detail of the outline specification will vary in accordance with previous experience, but the emphasis must remain with the application.

Evaluation of options

This phase concentrates on the identification of the candidate software packages, their examination and the refinement of the outline systems specification. The central activity of this stage is the collection of information. Both a general awareness of systems available and trends in the marketplace, and more specific knowledge concerning some key packages must be acquired. Attendance at exhibitions, conferences, and seminars offers the opportunity to gain an overview and to examine specific products. Directories, such as those listed in the references at the end of this article list the systems available on the market, and facilitate preliminary comparisons. Accounts of the application of packages are worth tracing in appropriate periodicals, since they offer a different perspective in showing how a package can be exploited and may trigger new ideas. Software packages can be examined at exhibitions, and demonstrations at the purchaser's location may be appropriate once detailed examination of a specific system is required. Evaluation versions of packages can be invaluable for the reasonably experienced computer user, but may prove a little daunting for the new user. Above all else, the range of options, and perhaps more importantly, the problems that are likely to lurk around the next corner, can be gleaned from other users and members of user groups. All reputable software suppliers should be able to recommend a few successful implementations of their system from which one or two can be selected for a visit or discussion.

The refinement of the outline specification will need to take into account local circumstances. These may include hardware constraints and any special aspects of the application, such as downloaded MARC records, or graphic display images alongside text. These local constraints may serve to narrow the choice of packages significantly.

Definition

During this stage attention is re-focused on the application under consideration. With the greater enlightenment gleaned during the information gathering of the previous phase it should be possible to specify a number of aspects of the application in greater detail. Typical issues that might be addressed include:

- databases to be established
- database sizes
- growth rates
- record structures, for both bibliographic and non-bibliographic data
- the information that will be sought from the database
- the form in which information needs to be presented
- the users of the system, their roles and experience in relation to the system
- how the system might be implemented, including personnel involved and project timescales

The outcome of these deliberations should be recorded in a more detailed systems specification which should form a basis for detailed discussions with systems vendors and any contractual agreements. This systems specification is also the detailed reference document.

The systems requirements specification or specification of operational requirements is an important document. It has three basic roles:

1. as a communications document to support discussion amongst those concerned with the development of the system.

2. as a reference document for consultation during implementation, maintenance and review.

3. as a legal document which may form part of a contract with a supplier.

The focus of the system requirements specification is the details of the facilities to be provided by the computerised system, with an indication of those which are mandatory and those that are merely desirable. The size of the system in terms for instance, of the number of records and transactions to be handled is also important. Background and environmental information, together with information on any special constraints should also feature. Lastly, a timetable for the implementation of the system is most important as a basis for monitoring progress.

Selection and design

Having identified what is required of a system, it should be possible to identify which system best meets the requirements for a given application. Quotations or tenders should be obtained from a small number of potential systems suppliers and formal negotiations for purchase should commence. For a small, cheap system purchase agreements and associated contracts may be relatively straightforward, but with a larger investment considerable debate may be important at this stage. An integrated package of hardware and software needs to be selected and the capability of all components tested. Once a suitable configuration has been designed, orders can be placed.

Implementation

After orders have been placed there may be a lull in activity whilst delivery of software, and in some circumstances, hardware is awaited. The opportunity can be taken to review the plans for implementation of the system. Implementation should start with any necessary building works, office re-arrangement, or network installation, followed by hardware installation. Once hardware has been installed, or if hardware is already in place, software installation and testing can proceed. This will start with the establishment of a small trial database, and associated elements of a trial system. Hardware and software, and their capability should be tested and any problems resolved, in consultation with suppliers.

Once the system is operational, databases can be established, and any system design, such as the format of report forms and printed outputs, undertaken. Security arrangements, such as the allocation of passwords and user ID's and write or read access to specific parts of the database need to be put in place. Staff training is essential, and appropriate user familiarisation programmes should be planned. Once these preparatory measures have been completed the system can move into full operation.

Evaluation and maintenance

After a short period of operation initial evaluation needs to be conducted. Is the system meeting the objectives identified at the beginning of the project? If not, where and how can improvements be made? Longer term evaluation needs to be conducted at regular intervals, say, annually, in order to ensure that the system is still meeting its objectives, and, indeed that those objectives are still valid. Software maintenance will often be provided by the software supplier. Upgrades and updated versions of the software will be available for implementation at intervals.

CHOOSING SOFTWARE

The new user of a software package often experiences difficulty in drafting an appropriate specification of the features that might be useful in a software package, and thus experiences difficulty in developing the crucial systems requirements specification. This specification must be unique to each application, but often a general checklist of the functions that are normally available in a specific kind of software package forms a useful basis from which the new user can develop their own specification. Checklists can be further developed by examining the features of the systems in the marketplace, and indeed need to be regularly updated to accommodate developments.

Criteria that might be included in a Checklist can be divided into two categories:

(1) those that are applicable to all software packages.
(2) those that relate to the specific kind of software package.

Here, we will first briefly review those criteria in the first category.

General Criteria for Software Selection

1. Cost

Cost is clearly a consideration, but since, in general, you get what you pay for, cost should not be a primary consideration. Software cost may also be a small component of the costs of the entire system, and better software may significantly reduce operating costs.

2. Lifetime and Life History

Well established packages, preferably with the same supplier throughout their lifetime are to be favoured. In many fields packages have now been available for ten or more years. Although not all of the best packages have done so, it is preferable if the package has enjoyed a solid reputation through a number of releases.

3. Originator

The reputation of the systems house responsible for writing a software package is important. Experience with other packages supplied by the same originator may be used in assessing a new package.

4. Supplier

With specialist software the supplier is often the originator, but with standard business packages there is often an agent acting as supplier. The user may look to the supplier for support and needs to feel confident that this will be forthcoming. The suppliers reputation and history should be considered.

5. Support

Most suppliers or originators offer some support. Good, readable manuals should be the norm. Other support may take the form of on-site training, off-site training, consultancy, assistance in setting up a system and a help desk. Some software packages have associated user groups and user group membership may provide a valuable source of information on the package. Both the quality and the cost of these support elements must be considered.

6. Maintenance

The software package should be appropriately maintained by the supplier. Maintenance involves removing bugs or errors; and, improving the software so that it incorporates new facilities and concepts. Many software suppliers offer maintenance contracts at about 10% of the price of the original package and this entitles users to new releases of the software. Other suppliers offer special discounted rates for upgrades to existing users.

7. Technical Considerations and Compatibility

The software must run under the operating system available in the hardware configuration to be used, and must also be available in a version that is compatible with the hardware. With the move towards UNIX - based systems and extensive use of DOS in microcomputer systems, compatibility is less of a problem than it once was, but still needs to be carefully checked.

8. Ease of Use

The quality of the human-computer interface is important for any software package. Factors to be considered include dialogue design and screen display.

9. Interfaces and Integration

Most software packages should be able to export and import data to and from other packages, of the same kind, such as between two word processing packages and two database packages. Some software will also export data to other kinds of packages as from, for instance, a database package to a word processing package. Other, packages may be part of an integrated suite of software that supports different activities such as word processing, databases, graphics and spreadsheets. It is important to be able to re-use data in a system in different formats so a high level of flexibility should be sought.

Criteria for Specific Software Packages

All of the earlier sections of this paper are applicable to the selection of any software package for any application. Criteria appropriate to specific software packages reflect on the facilities offered by these packages. A sense of Checklists must be developed to accommodate different applications. Since I am not sure which application areas may be of interest to you individually I have chosen to develop a general Checklist for the evaluation of database systems. Database systems may cover a wide variety of different applications. This list may be developed to include features appropriate to special purpose database systems, for example, terminology management or the recording of business transactions. Alternatively, it may be viewed as a model checklist showing the nature of such a checklist and from which it may be possible to develop similar checklists for other applications.

Checklist 1 lists the key features of database software. The basic database structure and its definition are considered first, followed by features associated with data entry. Ease of data entry and appropriate security to protect data integrity are important. All database systems, must consider renewal. In some systems such as text management systems more sophisticated facilities are available for indexing than in other systems. Equally the retrieval facilities may vary in nature and quality between systems. Once retrieval has been effected, data must be output to screen or in printed format. Security of data and software must be carefully controlled, especially in systems where there are many users.

CONCLUSION

The evaluation of software should be approached as a project. A methodology for a strategy for conducting the evaluation of software has been outlined. Criteria have been offered as an aid in the evaluation of software. Both general criteria and criteria applicable to the evaluation of database packages have been given.

CHECKLIST 1 : A CHECKLIST FOR THE EVALUATION OF DATABASE SOFTWARE.

1. DEFINING THE DATABASE

Basic facilities
What parameters are specified at definition?
What parameters can be modified once data has been added?
Display of data structure
Modification of data structure, both before and after database has been added
Relational database facility

2. DATA ENTRY

Preformatted screens
User-definable screens
Full screen editing and word processing facilities
Easy amendment of existing records
Easy deletion of existing records
Immediate retrieval on all keys
Check for duplicate records
User-defined fields
Variable and fixed length fields
Flexible field lengths
Use of windows to offer help and display authority lists
Protected screen areas eg labels
Data validation

3. INDEXING

Constraints on indexing terms eg length
Stoplists
Go-lists
Facilities for handling personal and corporate names
Controlled and natural language indexing
Selection of fields for indexing
Definition of field indexing characteristics
Authority control for names
Thesaurus control for subject terms, showing synonyms, homonyms, broader terms and narrower terms.

4. INFORMATION RETRIEVAL

Boolean search operators
Field limited searching
Proximity searching
Truncation
Range searching
Search history
Save search
Online help facilities
Display of index
Display of thesaurus
Use of related terms in searching
User-friendly interface

5. OUTPUT FACILITIES

Pre-formatted screen display
User-defined screen display
Printed output formats, both preformatted and user-defined
Special print formats and other facilities for SDI
Table creation and statistical analysis facilities

5. SECURITY

Passwords
User aids
Read only access
Access restricted to certain records, and/or certain fields