Evaluation methods of standard software with respect to their effectiveness

Citation for published version (APA):

Martin, H. H. (1992). Evaluation methods of standard software with respect to their effectiveness. *International Journal of Production Economics*, 24(3), 249-261.

Document status and date:

Published: 01/01/1992

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Evaluation methods of standard software with respect to their effectiveness

H.H. Martin*

*Faculty of Industrial Engineering and Management Science, Eindhoven University of Technology,
Eindhoven, The Netherlands*

Abstract

“Make or buy” of software has been a key question for management since the emergence of computerized information systems. Until today this question has not been answered satisfactorily. It is still unclear what evaluation methods should be used to support the decision process in the selection of a suitable package unambiguously.

In this article a first step in the development of a framework for evaluating standard software packages for maintenance management is taken. It will be demonstrated that conventional evaluation methods are unsatisfactory; therefore development of better alternatives is required. The importance of information requirements analysis is emphasized, in which the user-specialist, e.g. the maintenance manager, plays the keyrole.

Despite the importance of information requirements analysis there are no clear research findings yet with regard to analysis methods to be applied. The article reports on the on-going research which is focussed at the maintenance function.

1. Introduction

Due to the enormous decline in price of computer equipment as a result of technological advancements during recent years, the computer becomes more and more feasible for many kinds of new users. Especially for smaller firms and departments personal computers (= PCs) become an attractive possibility. Together with the large assortment of standard software packages (= SSWPs) for PCs becoming available, this enlarged the applicability of the computer considerably.

In most cases the PC is operated by the user himself, thus making him independent of centralized processing facilities and of the direct assistance by informatics specialists. As a consequence the user becomes more directly involved in the decision process whether to buy standard software, to adapt standard software or to develop software in-house.

The advantages versus the disadvantages of buying standard software have been discussed by several authors (i.e., Wortmann [1] and Stahlknecht and Nordhauß [2]). In general the authors agree that buying a suitable SSWP is cheaper as compared with in-house development. However, not seldom it is found that SSWPs do not “fit” without modification, requiring a considerable additional effort in costs and time, and usually functional constraints have to be accepted or the existing organization has to be changed in order to make the SSWP applicable at all. Still, sales representatives do spend a lot of effort in trying to convince potential clients of their readily available “optimal” solution. In that respect one may speak of a boom in advertisement promises. In order to be able to judge if SSWPs meet the requirements of a client a measurement instrument is needed, which measures the deviation between predefined required properties, in terms of information requirements analysis, and the properties of the available SSWPs considered.

In this paper a framework is presented within

*IFRIM (International Foundation for Research in Maintenance).

which some of the major aspects of the software evaluation process will be discussed. The framework describes the main evaluation activities and in what sequence they should be carried out. The most frequent evaluation approaches encountered in literature and in practical situations will be subjected to a critical analysis with respect to the requirements derived from the framework. In addition problems frequently encountered in practice will be discussed. Finally an indication will be given concerning future research in relation to information analysis and to its role in the software evaluation process.

2. The evaluation framework

2.1 Nomenclature

In order to prevent ambiguity some definitions relevant to the explanation of the evaluation principle will be presented and discussed in this section, i.e.,

- (1) function
- (2) main criteria definitions
 - (2.1) effectiveness
 - (2.2) efficiency
 - (2.3) flexibility
- (3) the dynamics of information requirements
 - (1) *Function*. The term *function* originates from systems theory, which is based on "thinking in terms of processes or functions". According to Blumenthal [3] an organization is a functional unit, described as an ordered set of subfunctions. A functional unit itself consists of subfunctions on a lower level of decomposition. The set of all subfunctions (or processes) required in the management of maintenance within an organization constitutes the maintenance function. The maintenance function has relations with other organizational functions, such as the production planning and control and procurement.

The interaction of (sub)functions is determined by functions at a higher level of aggregation. Low level functions can be formalized more easily as compared to high level functions, because of their rather concrete operational nature. Usually high level functions imply a high level of abstraction, resulting in a less formal degree of

specification. Functions at higher levels of aggregation set objectives and determine the constraints of functions at lower levels (Anthony's [4] levels of control).

The design and the evaluation of a function requires a thorough knowledge of the (application) area of the function concerned. In our case, general knowledge of maintenance theory in combination with situational aspects is a necessity in defining the functions of the maintenance function.

(2) *Main criteria definitions*. Software is, like any other type of information processing tool, the outcome of several different decisions made in the design stage. Each decision requires a certain type of knowledge in its own right. Therefore an evaluation should be based on sets of criteria with respect to each relevant area of knowledge. In this article, we distinguish effectiveness, efficiency and flexibility criteria.

(2.1) *Effectiveness*. The term *effectiveness* is defined as the degree to which a SSWP is functional, i.e., what a package is capable to do. Whether a SSWP meets the functional requirements of an application area is the first major question in an evaluation.

It goes without saying that the primary interest must be focussed at the achievable functional quality of the actual contribution to the functions supporting a predefined application area.

(2.2) *Efficiency*. The term *efficiency* is defined as the effectiveness traded off against the effort required to apply the SSWP.

Effort should be interpreted in a broad sense. The effort is not only determined in terms of hardware or informatics but also of, e.g. the required skills and education level of the users in addition to the cost of using the SSWP.

In fact, all evaluation aspects which do *not* clearly relate to the effectiveness criterion will be related to efficiency criteria. Due to the diversity of possible relevant efficiency aspects a multitude of specialistic disciplines are involved in the assessment of the efficiency.

(2.3) *Flexibility*. The term *flexibility* is defined as the degree to which the effectiveness or efficiency of a SSWP can be changed. The flexibility criterion is certainly the most complex of the three main criteria. Flexibility in relation to

the effectiveness criterion concerns the degree in which SSWPs are able to cope with e.g.,

- changing information requirements within an organization in the future;
- difference in requirements due to differences in organization types, considering that SSWP is put forward as "standard".

The flexibility of a SSWP can be provided by:

(a) The possibility to add functions (external modules) to the SSWP, or delete functions of the SSWP. Some SSWPs provide some flexibility by activating alternative functions already available in the SSWP (so called "parameterized software") in accordance with his needs. Then, no additional programming effort is needed if minor changes in information requirements are desired.

(b) (Re)design of the SSWP. If nonnegligible changes in information requirements cannot be met by the options readily provided by the SSWP under consideration, redesigning the software is a possibility to be considered. Specific attention of the SSWP developers for the implementation process (in terms of informatics: the way the programming-code is achieved) with respect to the ease of redesigning the software is essential here.

Software as a product is the eventual result of the contributions of several disciplines. E.g. the coding of the program requires programming skills in the field of informatics, the design of the layout of the screen display requires specific knowledge of ergonomics, etc. Therefore it is evident that not only application-dependent, but also application-independent factors relate to the flexibility of SSWPs. Fundamental multidisciplinary studies are required to complete the assessment of flexibility criteria for the evaluation of SSWPs. However, the application-independent factors are not explored in this article.

The effectiveness, efficiency and flexibility are situation-dependent criteria. The effectiveness of a SSWP in a typical maintenance situation is not increased by its possibility to add functions which are not required in that situation.

In view of the scope of this article we will, in the following sections, concentrate on the effectiveness criterion only.

(3) *The dynamics of information require-*

ments. Before entering into the evaluation and selection process, management should be aware of the consequences of using SSWPs. Accepting SSWPs, especially in complex application areas as in maintenance management, requires a long-term commitment of management and users due to the limited flexibility to the SSWPs which were chosen, and inevitably also to the seller of the SSWPs, at least if they can still be approached when the problems turn up.

Apart from the sellers creditability and his experience in the application area, the sellers "upgrading" policy is of major interest. The effectiveness, efficiency and flexibility of a SSWP can be improved by the developers of SSWPs, by making new versions of the SSWP available. Reasons to change a SSWP can be found in e.g. differences in information requirements, originating from important clients, or technological advancements in computer technology. However, there is always a certain risk that future versions presented are trivial if not undesirable or even invalid for particular clients, who in this respect are faced with, and are dependent on, the changes determined by the policy of the developer. This risk may enforce management to have to choose for in-house development only. On the other hand, the user may be faced with a more or less dynamic environment resulting in sometimes unpredictable information requirements. The use of SSWPs becomes less feasible in such situations.

2.2 *The parties involved*

In the decision process of selecting software four interested parties can be distinguished. On the one hand the *developer* and *seller*, on the other hand the *procurer* and *actual user*.

The *developer* is the "engineer" of the SSWP on hand. Because software is a multidisciplinary product, more than one developer with different skills may be involved in the development process. The dominant question here is from what source the functional requirements on which the SSWP is based did originate. Developers often designed software for a specific client. In that case the software is tailored to meet the specific requirements of that client. As both the client and the developers are faced with the high costs of

this development of software, they seek for possibilities to spread these high costs. Not seldom such specific software then is presented as “generalized” and is offered as SSWP in the standard software market in a later stage. Stahlknecht [2] for instance, found that the majority of SSWPs offered for administrative functions in the printing industry were, originally, designed for a single situation.

The seller is interested in maximizing his profits by selling SSWPs. It is clear that, in order to maximize profits, the investment in development should be minimized, whilst the suggestion of general applicability of the SSWP should be maximized. Up to what extent general applicability of SSWPs in a complex application area has been taken into account, from the seller point of view, is dominated by marketing considerations and not by solid research of the varying situations in the application area itself. In the sellers marketing approach it is very important to address both the user and procurer party.

The procurer is the eventual decision-maker. Based on mainly economical considerations (cost-benefit estimates) he decides about the possible investment in SSWPs. In general, the procurer has no knowledge in-depth of the application area on hand.

The user is considered to be the actual “problem-haver”. The user, being strongly involved in his day-to-day activities, must define the essential information requirements expected to improve his performance. In general, it can be stated that the users knowledge of specific situational aspects is not substitutable by external consultants.

2.3 Types of relationships

On the basis of the interests of these four parties, the characteristics of their relationships can be explained. In Table 1 the basic relationships are shown. Each relationship will be discussed according to its reference given in Table 1.

Type A. The procurer-user relationship

The procurer and user reside in the same organization. In general the user, as the primary problem-haver, is the initiator of the SSWP eval-

TABLE 1

The types of relationships of the interested parties

	Developer	Seller	Procurer	User
Developer		C	F	D
Seller			B	E
Procurer				A
User				

uation and selection process. Once the user knows what functionality is needed two basic approaches are possible. They have to be discussed with the procurer, who is responsible for the organizational and financial consequences of the user's proposed functional changes. In-house development, the first approach, potentially provides all required functionality whereas SSWPs, the second approach, at best can result in an acceptable functionality in view of the cost-benefit trade-off.

Type B. The seller-procurer relationship

This relationship usually becomes important at the end of the evaluation and selection process. If the user has arrived at a recommendation of a SSWP with respect to its functional requirements and the “efficiency”-experts support that recommendation, the procurer starts the negotiations with the seller to get on terms regarding pricing and delivery conditions. (One should not think only of cost and delivery date for SSWP, but also of additional services such as consulting, conversion of data files, training, etc.).

Type C. The developer-seller relationship

In most cases the developer of a SSWP and the seller are not identical. Application SSWPs such as maintenance management information systems, are mostly sold by consultants and by software houses established in the country in which they find their potential clients.

The seller is more or less dominant in this relationship. Because of his direct contact with the (potential) clients, he determines the upgrading policy (future versions) concerning functionality. The developer acts in this relationship as an implementor (“efficiency”-expert) only.

Type D. The developer–user relationship

Whether in-house development is preferred or whether a suitable SSWP has been selected for use, in both cases this relationship is of utmost importance. The developer's main goal is to design an information system which satisfies the functionality requirements established by the user. The better the developer understands the needs of the user, the more likely the resulting information system will satisfy the user's needs. Nowadays, the developer is not an expert in business functions and, consequently, relies primarily on his communicative abilities. In case the developer designs information systems for users within the same organization, direct communication can be much better established as compared with external developers. In the optimal situation users would not only define their own functionality, but would also be able to develop the information system themselves.

Type E. The seller–user relationship

The user's primary interest is to know where to find accurate and reliable information on the SSWPs functionality designed by the developer. Usually, it is impossible for the user to get in touch with the (original) developers of a SSWP. The seller party constitutes an intermediate between developer and user, thus adding an additional communication problem.

Type F. The developer–procurer relationship

In most cases the developer and the procurer do not communicate directly. If the developer is not a member of the same organization as the user and the procurer, it is the seller who will establish direct contact with the procurer. However, if the developer is a member of the procurer and the user organization (i.e., in-house development), he has to deal with financial constraints put forward by the procurer. These financial constraints can be a heavy burden for the "in-house" developer, as compared with the external developer, who can spread development costs over all future buyers of his SSWPs.

It is evident that it is vital that in particular the user is aware of the difference in interests of the

parties involved and of the dangers of the resulting unreliability and incompleteness of the information he will receive, as compared with what he needs to arrive at the correct choice for his specific situation.

2.4 Evaluation and selection phases

The assumption of general suitability of SSWPs including situations which they were not specifically designed for, is based on the assumption that different maintenance situations still can be considered to consist of sufficiently identical maintenance functions. However, the large variety of environments in which the maintenance function is embedded makes this assumption more than questionable.

In order to by-pass the problem of differences in situations it is assumed frequently that somehow the primary process (i.e., the production function) is directly related to the way in which the maintenance function is realized. The primary process utilizes certain types of technical systems, which are supposed to be – usually – arranged and handled in a type-of-industry-specific way. Ideally, the realization of the maintenance function, in terms of information requirements, is supposed to be very similar in different firms within the same type of industry. Individual differences are considered to be that small that they can be solved by the inherent flexibility of the SSWP.

On the other hand it is assumed that maintenance functions of different types of industry consequently will lead to different results with respect to the information requirements of the maintenance function.

However, only a detailed decomposition of the maintenance function permits to make conclusions if maintenance functions can be considered to be identical indeed. For the time being, therefore, we assume that each specific maintenance situation is unique. In particular at the lower levels of decomposition nonnegligible differences will show up.

The evaluation and selection process concerning the suitability of a SSWP can be carried out in a number of subsequent steps:

- (1) define the domain of the maintenance situation;

- (2) establish possibly suitable SSWPs;
- (3) determine the information requirements;
- (4) establish and evaluate remaining suitable SSWPs;
- (5) select functionally suitable SSWPs, or develop in-house.

The third step, determination of the information requirements, is certainly the most important. Figure 1 shows the steps (the single bordered blocks) of the evaluation and selection process and their relationships. The double bordered blocks represent the methods or aids to be used in each step.

First, the domain of the maintenance situation is established. In this step the main subfunctions

belonging to the maintenance function are determined. E.g., it can be decided to include the function inventory control of maintenance items in the maintenance function. This type of decision requires a global general functional maintenance model on which to base the considerations for that decision. This general model should be detailed more specifically according to the situation at hand. As a starting point the EUT-maintenance model can be used [5]. The EUT-model, intended to represent any maintenance situation, reflects the subfunctions, and the special areas of knowledge in maintenance at a very high level of abstraction. A sound understanding of these knowledge areas in combination with situ-

Fig. 1. The steps in an evaluation and selection process.

ational circumstances provides the basis for the definition of the global (high level) maintenance functions in a specific case.

Once the combination of subfunctions has been defined a preliminary selection of SSWPs can be made. The purpose of this step is to reject those SSWPs which obviously are unsuitable to support the specified functions. At this point it will become clear if there are no or only partially suitable SSWP alternatives left. If no SSWP meets the demands completely a choice must be made between adaptation and in-house development.

If there are still SSWPs left worth considering, a detailed information requirements analysis is necessary. This detailed analysis of the information requirements must provide, essentially, the effectiveness criteria for the evaluation. In the following section the principles of such an evaluation will be discussed more in depth.

Once the (functional) evaluation has been carried out it has become definitely clear if there are still candidate SSWPs left for selection. Eventually, the final possible selection is to be based on the efficiency criteria. In case the selection step does not result in finding a SSWP considered to be suitable, in-house development is the only option left.

3. The evaluation principle

3.1 Introduction

In the foregoing section it was illustrated that the eventual evaluation step is only part of the total evaluation and selection process of SSWPs. Because of the sequential relationships the correctness of the evaluation depends on the accuracy with which each of these preceding steps was carried out. In this section the evaluation itself will be discussed. Special attention will be paid to the basic steps of an evaluation and to the major constraints of each basic step. Starting point in the evaluation phase is the effectiveness, determined in the detailed information requirements phase.

The major problem in this stage is to decide how the information requirements analysis is to be carried out, both theoretically sound, and practically possible. Despite the short history of Information management science a diversity of

approaches have been published in the literature (e.g. [6]). Some of these advocate a top-down information analysis, comparable to the approach presented in this article, others prefer a bottom-up approach. Some pretend to cover only certain management areas and only one decomposition level within an organization or department. The majority of the information requirements analysis approaches make use of graphical representations of data and of processes. In general, these approaches are geared towards the design of more or less completely new information management systems. So far, it is unclear which approach is to be recommended for evaluation purposes, and whether the same technique is also suitable in the redesign of an existing information system. Research in this area is essential, as by now the only requirement of these approaches is to provide a similar format for both the information requirements and the SSWP properties, in order to allow their comparison.

In order to carry out an evaluation, effectiveness criteria must be defined beforehand. Many criteria can be established, defining the situational relative importance of the functional requirements. E.g., sometimes it is desirable to limit the number of work orders on hand in an operational work order control function in order to prevent losing the overview. In that case a volume restriction for the number of work orders on hand is a criterion.

Of all effectiveness criteria the completeness criterion may require major attention. Basically, the functional completeness criterion facilitates a direct comparison of the required functionality and the functionality provided by a SSWP. The required functionality, derived from the detailed information analysis, should be defined at least with respect to:

- the manual and the automatic functions considered necessary for operation of the information system;
- the relations between the distinctive functions;
- the input and output datatypes handled by each function.

Only the functions requiring automated data processing can be compared with the functions available in SSWPs, as the required manual functions are embedded in conventional proce-

dures which were found to be irrelevant for automated data processing.

As a result, a comparison of functions can give certain types of outcome (i.e., a function is identical to the function available in a SSWP; a function has a different internal operational mode but uses identical input and provides identical output data meeting the requirements as defined, and so on). A clear definition of requirements of all possible outcomes should be agreed upon by the responsible person(s) before for the evaluation is started. Other effectiveness criteria can be defined if considered necessary.

3.2 Evaluation steps

In an evaluation three basic steps can be distinguished:

- (1) *Criteria*. Establishment of the criteria based on the requirements for information.
- (2) *Properties*. Determination of the properties of the SSWP which are relevant in view of the criteria.
- (3) *Comparison*. The assessment of the lacunae in the information provided the SSWP.

(1) *Criteria*. The comparison of the properties of the SSWPs and the requirements that have been established cannot be carried out in all cases. Sometimes it is undesirable, or even impossible, to automate all defined functions. In particular functions at higher levels of aggregation can be defined in a less formal manner only. Therefore unambiguous specification of information requirements at higher levels is subjected to constraints.

(2) *Properties*. The criteria established in step (1) determine how and what to look for in a SSWP. As has been demonstrated earlier, it is of primary importance that the format to present the functional properties of the SSWPs is similar (compatible) to that which has been used for the functional requirements. In assessing the required information on SSWPs one is usually completely dependent on information provided by the sellers and by registered users. Because there are no accepted standards on the way in which the functional properties of SSWPs should be presented to potential clients, there is only a small chance that a client will receive any detailed information about the SSWPs indeed, un-

less he is known beforehand to buy the SSWP anyhow. At best, some advertisement-like sheets can be obtained from the sellers, giving only global promises and global black box illustrations of the SSWP concerned.

(3) *Comparison*. A suitable measuring scale is required in order to be able to compare the values of the potential properties of the SSWP with the criteria. The comparison of the potential of a SSWP with the information requirements normally is carried out by means of a *relative* scale. Using a relative scale means that evaluation criteria are used only to compare a set of available SSWPs. Inevitably, relative scale comparison favors SSWPs which provide more functions, e.g. the SSWP which can store more workorders, etc. (the "more-is-better-syndrome").

However, in order to be sure that the evaluation is correct indeed, it is necessary to define the information requirements and the evaluation criteria on an *absolute* scale. Using an absolute scale requires to establish the exact functional requirements in the first place. Then, these requirements are used as criteria to assess SSWPs.

In conclusion, with a *relative* scale at best the "least worst alternative" turns up, whereas only an *absolute* scale can provide the answer concerning the actual suitability.

4. Evaluation methods

The need for practical evaluation methods has been put forward by many authors over the years. Some methods concern a general approach, other are directed at specific branches or specific departmental functions. In this section the diversity of existing evaluation methods is categorized in five standard approaches. Taking into account the evaluation principles presented in Section 3, some critical remarks will be made about these approaches.

The following approaches will be discussed:

1. The certification method.
2. The checklist.
3. The ranking approach.
4. The questionnaire.
5. The phased approach.

4.2 The certification method

In this approach a committee of acknowledged specialists is formed in order to present each their opinion about a set of SSWPs. They are expected to consider whether in their opinion the SSWPs do have a certain minimum set of functions which they regard to be essential. In case a SSWP supports that minimum set of required functions the SSWP passes the evaluation. Thus, an accepted SSWP is a candidate recommended by the committee.

Usually, a committee is used if the specific expert knowledge considered essential is taken to be not available with one person, because of the diversity of relevant aspects. The essential question then is how, exactly, the committee arrives at its judgment. The criteria which were used remain unclear in most cases (see e.g. [7,8]). As has been demonstrated before, criteria must be based on the results of an information requirements analysis.

An accurate definition of the maintenance situation is a prerequisite for this information analysis. In general, certification appears to lack sufficient specification of the maintenance situation considered. Therefore the opinion of a committee is, in most cases, questionable.

4.3 The checklist

The most frequently encountered approach is the checklist method (see Fig. 2). The checklist

	WORKORDER PLANNING				SHUTDOWN PLANNING				BUDGETING				PROCUREMENT							
PACKAGE X	YES	NO	YES	YES																
PACKAGE Y	YES	YES	NO	NO																
PACKAGE Z	NO	YES	YES	NO																

Fig. 2. The checklist.

consists of a table with the functions which have been distinguished, on one axis, and the SSWPs which support the intended application, on the other axis. A marker is placed in the appropriate box if a SSWP contains the function mentioned (see [9,10]), limiting the answer to "yes" or "no". Thus, theoretically, only the criterion of completeness is covered in this approach.

Again, the authors do not explicitly present the maintenance model which they used to justify the global or detailed functions they took up in the checklist.

Some authors don't make a clear distinction in effectiveness and efficiency criteria, which results in a checklist mixing up functional and efficiency aspects. In that case the maintenance specialist, who is inclined to be preoccupied with effectiveness, rather than efficiency criteria, is confronted with features which can be judged only by informatics specialists, ergonomists, etc. Usually authors fail to specify at what stage in the evaluation and selection process they want to apply the checklist: in the global maintenance domain specification or in the detailed information requirements analysis.

In most cases only main, globally indicated, functions are distinguished, which limits the suitability of the checklist to the preliminary assessment of SSWPs.

4.4 The ranking approach

The ranking approach is similar to the checklist approach in most aspects. The presence of aspects ("properties") in SSWPs, covered in this approach, is balanced against the relative importance of the required properties distinguished (e.g. see [11]).

An example of a frequently used ranking approach is presented in Fig. 3.

In essence, the remarks which were made with respect to the checklist approach apply here as well. In addition to this comment it can be stated that it seems practically impossible to define *objective* criteria for the ranking process, which true ranking would require.

4.5 Questionnaires

Questionnaires are very popular in literature as well (see for example [12,13]). Figure 4 illus-

		PACKAGE X		PACKAGE Y	
		A	B	A	B
P R O P E R T I E S	1	10	80%		
	3		100%		
	2		70%		
	.	.	.		
	.	.	.		
n					
TOTAL SCORE =		315			

LEGEND:

A: relative importance of a property
 B: the degree to which a property is available
 Total score: e.g. the sum of A·B over n properties

Fig. 3. An example of a ranking approach.

trates an example of a part of a questionnaire published and recommended by the COMGE-group [2]. The purpose of a questionnaire is twofold. Firstly, it should assist in assessing all aspects, some of which otherwise might be forgotten, and secondly, questionnaires are used to question the seller of the SSWP, in an effort to discover the completeness and the weak and the strong points of the seller and of his package(s). Typical questions about the sellers cover topics such as seller reliability, financial disposition, etc.. In most cases questionnaires tend to be very general, emphasizing the efficiency aspects. Therefore questionnaires can be useful to collect additional information needed in the final selection stage when the specific functional evaluation has already been carried out.

*	<u>The developer</u> Name, address of developer Nature of the firm - Hardware vendor? - Software house? - Systems developer? etc.
*	<u>The package</u> Name of package Available version (release date) Is a new version currently under development? etc.

Fig. 4. An example of some elements out of a questionnaire [6].

4.6 The phased approach

The phased approach, which is similar to the approach presented in Section 2 (see Fig. 5) consists of a number of sequential steps beginning at the top, going down step by step, without conditional bypasses.

Brownstein and Lerner [5] recommend the phased approach in software evaluation and selection, referred to as "the evaluation process". However, the selection of SSWPs is only one step in the evaluation process. Furthermore they assume that there is always, definitely, at least one suitable (in our view: "least unsuitable") SSWP available for selection. Thus implicitly, the "buy" decision has been made already before the start. The final selection is made by means of the ranking approach. Thus, all weaknesses of the ranking approach are also taken up in the phased approach, in addition to the lack of a true suitability check. Brownstein and Lerner describe their approach in global terms and their primary interest is focussed on the way in which the evaluation is to be organized. Because of the, suggested, general nature of their approach they consider it suitable for the evaluation of SSWPs for any field of application, including the field of maintenance information systems.

4.7 Conclusion

The categorization of evaluation approaches presented above lists only the approaches most

Fig. 5. The phased approach [5].

frequently described in literature. In essence, even these five approaches are not truly methods, if compared with the requirements of an evaluation as described in Section 2. Only the checklist, the ranking approach and the questionnaire can be categorized as, intentionally, true evaluation methods.

Most striking is the fact that no or only very little attention has been paid to the development of sound functional maintenance models. The omission of such a model makes any outcome of an evaluation and selection process questionable. If maintenance models are being used at all, they are presented as very general maintenance models. However, not all functions are described at the same level of decomposition, a drawback which is probably due to the specialistic technical background of the authors. I.e., mechanical engineers appear inclined to pay much attention to the function “make lubrication schedules”, in contrast to a much more complex, but less decomposed, function such as “spare parts inventory control”, which usually is limited to stock registration. Some authors make no clear distinction between *functions* and *data* in their models, thereby confusing the reader.

Generally, maintenance models presented are suitable for the description of maintenance on a high level of aggregation only. However, only a *detailed* information requirements analysis can provide sufficient possibilities to eventually discriminate between the SSWPs under consideration. Finally, the general lack of *absolute* measuring scales leads to unreliable results in respect of the correctness of the results of the evaluation.

5. Executing an evaluation in practice

The foregoing discussion focussed on some fundamental considerations in executing evaluations. In this chapter we will present some practical considerations indicating drawbacks in the evaluation of software. In general most limitations, as appeared in the discussion of existing methods, presented in the foregoing section do apply here as well.

Some problems turning up frequently are:

(a) No clearly structured maintenance model is used. Without a structured high level maintenance model it is virtually impossible to assess which global functions have to be distinguished or need detailing. In the evaluation and selection approach presented in Section 2 it was shown that subsequent stages then will be questionable, resulting in a doubtful outcome of this process.

(b) The information requirements analysis stops at a too high level of aggregation. The stage of information requirements analysis is basically the most vital step in the evaluation and selection process. The selection of an information requirements analysis method, the level of detail and the accuracy of the analysis, determine the quality of the eventual evaluation.

(c) The criteria are based on relative, not on absolute measuring scales. Relative criteria can discriminate only between the *existing* SSWPs which were, somehow, chosen for evaluation. The resulting sacrifice with regard to the functionality cannot be prevented if relative criteria are used. In the end this results in selecting the “least unsuitable SSWP”, which may very well be very poor in respect of the – not assessed – true requirements of the situation at hand.

Frequently, a low level of attention for information requirements analysis tends to accept the use of relative criteria, expecting that it still will to lead to correct results.

(d) No clear distinction is made between effectiveness and efficiency aspects. As has been stated before the software phenomenon involves many aspects involving several disciplines. Theoretical knowledge of maintenance in view of specific situational maintenance aspects is needed to define the evaluation criteria. Other disciplines cover other valid but not maintenance-specific, e.g. efficiency considerations. There-

fore, in order to be complete, the execution of the evaluation and selection process requires a team of all specialists concerned, in which the maintenance specialist has a vital position because of his specific knowledge of the functional requirements.

(e) The information available about the functionality of the SSWPs is global only and insufficiently adequate to allow of judgement of the functionality. In the information requirements analysis stage a certain method and a certain format are used. In order to be able to use the pre-defined criteria, derived from this analysis, it is necessary to describe the properties of the SSWP to be compared in a similar format. Even if software sellers are willing to give information about the functionality of their SSWPs, it is necessary to convert that information into the format required, which may be hampered by ambiguity due to incompatibility.

6. Conclusion

Considering the problems explained in this paper the question arises whether complete in-house development is to be preferred to buying some available SSWP.

To answer that question it is necessary to examine if the general recommendations in support of buying SSWPs are still valid, because most statements on the potential of SSWPs refer to the status quo at the time that the SSWP was designed, which may be up to a decade earlier.

Proponents of the SSWP buying strategy primarily state that in-house developments are much more time-consuming, more costly and more risky, resulting – if positive at all – in more expensive, delayed and usually inferior product.

This strategy may be acceptable for general purpose software for rather standard situations, like wordprocessing packages in which case the user is preferred to adopt his way of working to the requirements of the package, but in cases where individual specific information requirements dominate and where generalizations remain unproven – as can be expected in the area of the maintenance function – very often so called time-consuming and risky in-house developments turn out to be the only feasible alternative. In such complex application areas the principal

question whether to buy SSWPs or to develop dedicated software in-house is not an issue to be determined beforehand on the basis of some global simple statements (see Section 2.1).

As has been demonstrated, only a detailed information requirements analysis can provide the tool to arrive at a correct judgment of the effectiveness of SSWPs.

In complex application areas, such as maintenance, a detailed information requirements analysis represents a considerable “investment” in terms of effort spent by scarce application specialists, i.e., familiar with the maintenance function. If the only purpose of this analysis would be to enable users to define their evaluation criteria, with the potential risk to find no suitable SSWPs in the end, the “investment” could be unjustified. However, in the ideal situation, the information requirements analysis is performed to serve both alternatives: the evaluation of existing SSWPs and, if the result is negative, the in-house development, from scratch if necessary.

So far no references have been found making clear statements about a satisfactory way in which an information requirements analysis could be carried out, i.e., methodology, and in what terms analysis results should be defined, i.e., format definition.

Because of the general nature of the information requirements analysis problems identified in this article, the same type of problems turns up in other application areas, e.g. production planning and control, inventory control, and such like. Fundamental multidisciplinary research is needed in order to identify generally valid principles for the methodological way in which an information requirements analysis could be carried out.

Research in a specific area, such as maintenance, will contribute to the perfection of the fundamental framework, and at the same time will assist in establishing the refinements required in maintenance.

References

- 1 Wortmann, J.C., 1986. Developments of the computer and its applicability. Terotech '85, NVDO-Rep., Den Haag (in Dutch).

- 2 Stahlknecht, P. and Nordhauf, R., 1981. Case Study in the Methodology of Hardware and Software Selection in Small and Medium Sized Firms. Oldenbourg, München (in German).
- 3 Blumenthal, S.C., 1969. Management Information Systems a Framework for Planning and Development. Prentice-Hall, Englewood Cliffs, NJ.
- 4 Anthony, R.N., 1965. Planning and control systems: a framework for analysis. Studies in Management Control. Division of Research, Graduate School of Business Administration, Boston, MA.
- 5 Geraerds, W.M.J., 1990. A general functional maintenance management model. Paper presented at the IFORS Conf., Athens (to be published by Pergamon Press).
- 6 Blank, J. and Krijger, M.J., 1982. Evaluation of Methods and Techniques for the Analysis, Design and Implementation of Information Systems. Academic Service, Den Haag.
- 7 DIN 66230, 1981. Informationsverarbeitung, Program Documentation. Normen ausschuf Informationsverarbeitung (NI) im DIN Deutsches Institut für Normung, e.V. (in German).
- 8 DIN V 66285. Vornorm (preliminary norm), Application Software Testing Principles. Normenausschuf Informationsverarbeitung (NI) im DIN Deutsches Institut für Normung e.V. (in German).
- 9 Armstrong, J., 1985. Planned maintenance and the use of computers. BSRIA, Berkshire.
- 10 Parkes, D., 1982. What's what in the software scene. National Conf. Computers for Maintenance Engineers, London.
- 11 Frank, J., 1977. Standard Software; Criteria and Methods for Evaluation and Selection of Software Products. Verlagsgesellschaft Rudolf Müller, Köln-Braunsfeld (in German).
- 12 COMGE-questionnaire on behalf of an investigation of computer programs, 1981. COMGE (in Dutch).
- 13 Kelly, A., 1980. Maintenance documentation and computers.
- 14 Brownstein, L. and Lerner, B., 1982. Guidelines for Evaluating and Selecting Software Packages, Elsevier, New York.