

Titre:	Project management's perceptions on software systems evaluation
Title:	
Auteurs:	Germinal Boloix, & Pierre N. Robillard
Authors:	
Date:	1995
Type:	Rapport / Report
Référence:	Boloix, G., & Robillard, P. N. (1995). Project management's perceptions on software systems evaluation. (Rapport technique n° EPM-RT-95-02). https://publications.polymtl.ca/10034/
Citation:	

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie:	https://publications.polymtl.ca/10034/
PolyPublie URL:	

Version:	Version officielle de l'éditeur / Published version
-----------------	---

Conditions d'utilisation:	Tous droits réservés / All rights reserved
Terms of Use:	

Document publié chez l'éditeur officiel

Document issued by the official publisher

Institution:	École Polytechnique de Montréal
---------------------	---------------------------------

Numéro de rapport:	EPM-RT-95-02
Report number:	

URL officiel:	
Official URL:	

Mention légale:	
Legal notice:	

22 MARS 1995

EPM/RT-95/02

Project Management's Perceptions on Software Systems Evaluation

Germinal Boloix
Pierre N. Robillard

Département de génie électrique
et génie informatique

École Polytechnique de Montréal
Janvier 1995

gratuit

Tous droits réservés. On ne peut reproduire ni diffuser aucune partie du présent ouvrage, sous quelque forme que ce soit, sans avoir obtenu au préalable l'autorisation de l'auteur,

OU des auteurs

Dépôt légal, novembre 1993
Bibliothèque nationale du Québec
Bibliothèque nationale du Canada

Pour se procurer une copie de ce document, s'adresser:

Les Éditions de l'École Polytechnique
École Polytechnique de Montréal
Case postale 6079, succ. Centre-ville
Montréal, (Québec) H3C 3A7
Téléphone: (514) 340-4473
Télécopie: (514) 340-3734

Compter 0.10 \$ par page et ajouter 3,00 \$ pour la couverture, les frais de poste et la manutention. Régler en dollars canadiens par chèque ou mandat-poste au nom de l'École Polytechnique de Montréal.

Nous n'honorons que les commandes accompagnées d'un paiement, sauf s'il y a eu entente préalable dans le cas d'établissements d'enseignement, de sociétés ou d'organismes canadiens.

Project Management's Perceptions on Software Systems Evaluation

Germinal Boloix, Pierre N. Robillard

January 1995

École Polytechnique de Montréal

Département de Génie Electrique et de Génie Informatique

C.P. 6079 succ Centre-ville

Montréal, Québec H3C 3A7

Tel. (514) 340-4031, 340-4238 - Fax. (514)340-3240

boloix@rgl.polymtl.ca

pnr@rgl.polymtl.ca

Summary

We report in this paper the perceptions of project managers from a software systems evaluation perspective. Testing software approaches in industry is a fundamental step to consolidate theoretical underpinnings. Project managers recognized the importance of the approach and considered evaluation of information systems a fundamental contribution for improvement-oriented paradigms.

Important software systems evaluation attributes are identified using a hierarchical approach to software systems evaluation. Definitions of software systems attributes to be evaluated are improved after exchanging with project managers. Objectives for each software system attribute to be evaluated are clearly presented, defining the basis for evaluation. The analysis of additional levels of depth in the hierarchy of software systems attributes has been used as a mechanism to validate the choices at higher levels. Finally, additional software system attributes deemed important by practitioners have been incorporated into the scope of software system evaluation.

Keywords: software systems evaluation, project management perceptions, hierarchical evaluation approach, technology transfer

1.0 Introduction

The benefits of operating a system in an organization is an area of concern for any business. Productivity is the fundamental economic measure of a technology's contribution. Delivered computing power in the U.S. economy has increased by more than two orders of magnitude since 1970 yet productivity, especially in the service sector, seems to have stagnated [B93]. There is a lack of metrics for usability and productivity of people and organizations who use computers. There are almost no quantitative behavioral measures of general trends, over the years, on how human and organizational productivity are directly affected by the use of computer systems [GBL91].

In the information systems area, post-implementation evaluation approaches have been suggested to establish the worth of information systems [K90]. Multiple-criteria evaluation approaches, which include subjective and objective evaluations, give equal consideration to both user and system constraints [C82]. The problem of conflicting user points of view has been highlighted in these approaches. User, manager and developer evaluations are considered within a goal-centered view to compare pre-established objectives to actual results.

There are so many aspects to consider during software system evaluations, that simplifying approaches are required. Instead of considering hundreds of unrelated evaluation points, a hierarchical decomposition of software systems attributes is necessary. The advantages of a top down approach include a high-level view of the evaluation problem which facilitates management understanding, the possibility of introducing additional levels of decomposition for those software system attributes requiring in depth evaluation and a structured evaluation approach which allows evaluation interpretation.

We report in this paper the perceptions of project managers from a software systems evaluation perspective. Testing software approaches in industry is a fundamental step to consolidate theoretical underpinnings. Without practitioners participation, software research rest an intellectual exercise which could make little contribution to the software community. Thus, involving industry in research can be considered an important step to facilitate technology transfer among scientists and industry.

The first step in gathering project management perceptions is to present software system evaluation attributes to practitioners. Each software system is characterized into

three dimensions: software systems intrinsic attributes, software systems production process and contribution of the system to the organization. Each of these dimensions is decomposed into three factors and each factor is categorized into three levels of maturity. Once participants understand the evaluation categories, general comments from participants are gathered. These comments include the difficulties to assign evaluation categories, the need of expressing the evaluation objective for each factor, the identification of those descriptions that require more elaboration, and the identification of general comments about the evaluation approach and its usage.

The second step in the process requires participants to analyze each of the software systems factors and determine the important subfactors. For example, for the agent factor, subfactors suggested are team organization, team experience, team motivation and team size. Project managers are requested to establish the order of importance of these subfactors. Participants could suggest additional subfactors deemed important or other factors not included in the evaluation approach.

The third step in the process is to evaluate specific information systems familiar to the participants. This step requires a judgment by participants on the maturity level of each of the software system attributes of information systems. A clear understanding of evaluation categories is required in this step. An evaluation profile is generated presenting the results of the evaluation. Those attributes judged weak during the evaluation may require more in depth evaluations. Comparisons among information systems can be performed using the evaluation profile.

The article is organized as follows. Section 2 describes the approach to gather project management perceptions. Section 3 analyzes software systems evaluation attributes, the ranking of attributes and the feedback's comments from participants. Section 4 presents the evaluation profiles information systems. Finally, Section 5 gives some conclusions and ideas for further research.

2.0 Requesting project management viewpoint

In this Section, the characteristics of the testing environment are presented. The session involved 2 facilitators and 5 project managers during a working session. A document containing the details of the evaluation approach was released to the participants together

with an evaluation profile used to assess each system. Software systems evaluation elements were organized hierarchically by dimensions and factors. Each dimension contained three factors, and each factor was evaluated using three categories (i.e., basic, intermediate and advanced). The package contained a questionnaire organized by software evaluation factors and the statements to evaluate their maturity (i.e., 9 factors and 27 categories). The package also included a suggested decomposition of factors into subfactors.

Project managers participating in the validation had senior experience on software development. They have been recently involved on a major software development project. Participants were allowed to interact with facilitators to clarify the statements describing each category. Facilitators were in charge of presenting software systems characteristics and guiding participants to select the categories for the evaluation of information systems.

Project managers had chosen one or two systems they were familiar with, and were ready to evaluate these systems using the questionnaire and evaluation profile. Participants also had the task of understanding the description of each category and defining a ranking of a few subfactors according to their relative importance. Results of the ranking process are given in a section below.

The session was interactive. An overall presentation of the important elements in software systems was given to the participants before proceeding to generate the evaluation profile of specific systems.

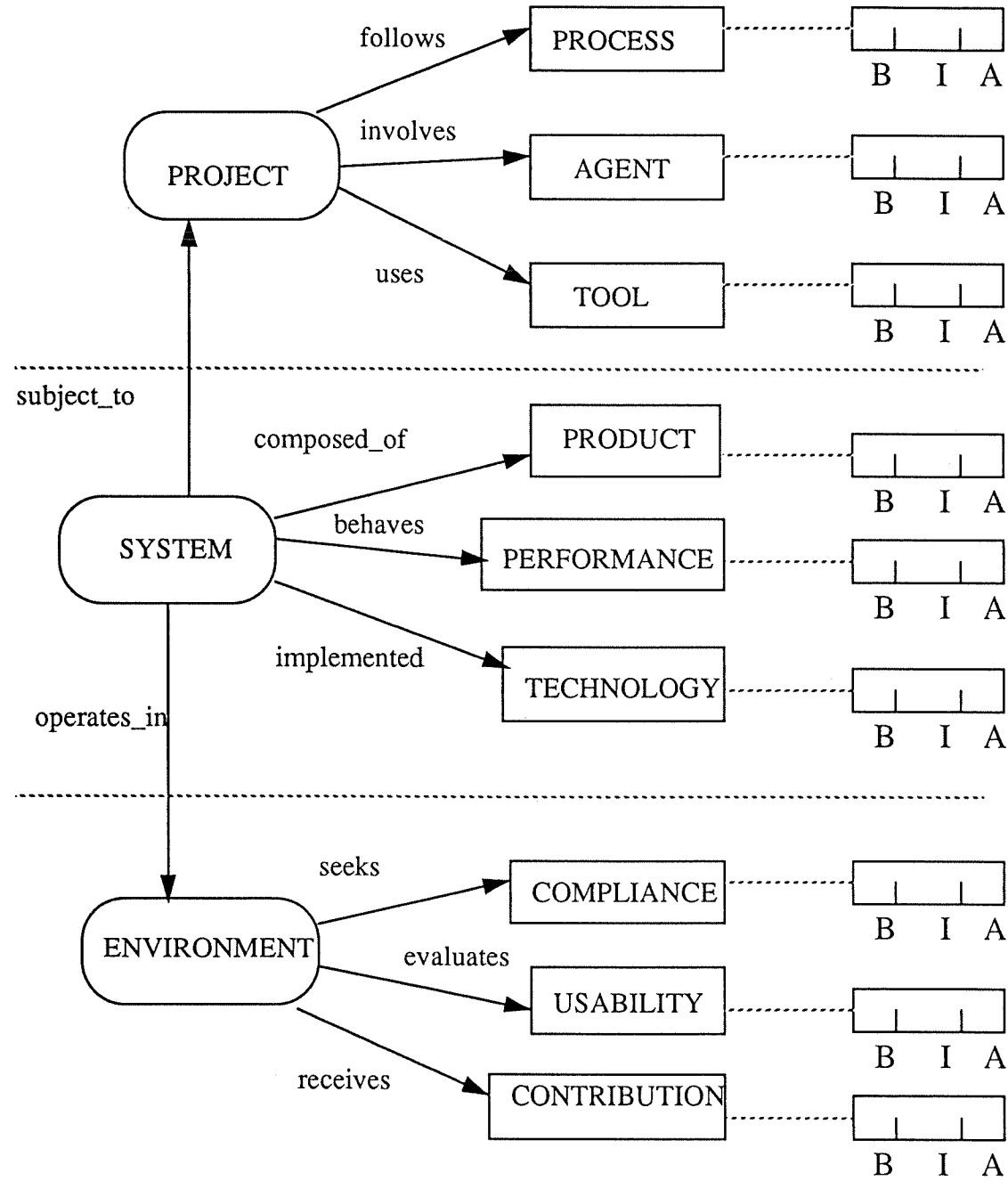
2.1 Understanding the important elements in software systems

We present in a top-down manner the important dimensions to retain in software systems, its decomposition into factors and the categorization of these factors for evaluation purposes. These aspects are organized by level, following a common framework. Each factor highlights the important elements to retain, by suggesting subfactors.

The software system attributes are organized into three dimensions, according to three different viewpoints (i.e., those of the developer, operator and user). The knowledge retained is that of the software product, its production process and its impact on the organization. The three dimensions are interconnected. A system may be *subject_to* several projects during its lifetime: initial development and enhanced versions. A system *operates_in* an organizational environment: users interface with the system and services are provided by the system.

Figure 1 presents an entity-relationship diagram which further decomposes the dimensions into factors. A project *follows* a process, it *involves* some agents and *uses* some tools. The system is *composed_of* products, it *behaves* at some performance level and it is *implemented* in a particular technology. The environment *seeks* compliance with system requirements, it *evaluates* the usability of the system from the user's perspective and it *receives* a contribution or benefit from the operation of the system.

Figure 1. Software Systems Attributes



2.2 Ranking of attributes

2.2.1 Project dimension

Project organizations use to carry out activities in coordinated ways. Project teams differ in how they operate as well as in their underlying observable operation. Projects can be coordinated by a traditional hierarchy of authority, by reliance on individual initiative, by collaborative discussion and negotiation, or by virtue of alignment with a common vision or direction [C93].

The following tables present the ranking of subfactors such as project managers suggested. The rows represent the subfactors and the columns their relative importance. The first five columns represent the relative importance given by each manager interviewed (M1 to M5), whereas the last column (T) is the resulting ranking, obtained by analyzing individual rankings and assigning a final order among the subfactors. In some cases, managers gave the same ranking for each subfactor, meaning they considered the same relative importance of subfactors (e.g., the first column in Table 1 indicates that process model and method are equally important). In other cases, managers considered that all the subfactors were equally important, such as the third and fourth columns in Table 1.

Process factor

The process that led to the development of the system, together with the design decisions and their justification, constitutes the backbone to improved quality and productivity. Process model evaluation establishes whether it is a well defined process, it is well

Table 1: Process

Subfactors	M1	M2	M3	M4	M5	T
Process Model	1	2	1	1	1	1
Method or technique	1	3	1	1	4	3
Measurement facilities	4	4	1	1	5	5
Availability of standards	3	5	1	1	3	4
Schedule constraints	2	1	1	1	2	2

documented and it has been already experienced by the organization producing the software. The characteristics of the method or technique (e.g., Structured Analysis/Structured

Design, Object-oriented approach, Jackson development method). Considering whether the process is being measured for project control and product quality assurance using software metrics. Determination of the availability of standards documenting the process and the characteristics of the deliverables. Considerations regarding planning and scheduling of the project.

According to the results of ranking these subfactors, process model considerations are the most important, followed by schedule constraints. It is interesting to notice that in most cases availability of measurement facilities was not considered important. Some comments given by participants were as follows:

Process should involve the software life cycle activities and the techniques being utilized. The process factor requires an assessment of the process model characteristics (e.g., waterfall, prototype, incremental) using specific development techniques (e.g., Structured Analysis / Structured Design, Object-oriented approach, Jackson Development Method).

Process should capture primarily the management of the process rather than its technical characteristics. It is important to identify the stability of the process and the quality of its documentation.

Aspects regarding project schedule and conformance to plans should be considered. The assessment of the project factor should evaluate the stability and quality of the standard process model being used.

Agents factor

Criteria for agents require the understanding that a team of developers is being evaluated, rather than individuals. Well defined project organizations with rigid command lines and pre-established authority decision points, have to be differentiated from informal organizations; autocratic organizations also differ from democratic styles of project management. Experience of personnel involved during software development may impact attributes of the software (e.g., quality, complexity); it is well recognized that the quality of resources has a major impact on project quality and productivity. Motivation is also an important characteristic to retain, even if difficult to measure objectively.

Analyzing Table 2, it is clear that team experience and motivation were the impor-

Table 2: Agents

Subfactors	M1	M2	M3	M4	M5	T
Team organization	2	4	2	3	1	3
Team size	4	3	4	4	4	4
Team experience	1	1	1	1	3	1
Team motivation	3	2	3	2	2	2

tant criteria to retain, followed by organization and size. This result highlights the well known fact that if the overall experience and motivation of the team is high, its size and organization are secondary factors that can be controlled.

Important comments given by participants included:

Type of agents participating during the project should include managers, technical personnel and users assigned to the project. The assessment of agents concerns all participants that contribute to the project. It is a team evaluation rather than individual evaluations.

Emphasize qualitative aspects regarding agents capacity, rather than quantitatives aspects regarding number of participants in the project. The number of participants in the project can be captured using absolute scales of measurement. However, the assessment of the team requires judgment.

Experience and motivation are the most important considerations for the evaluation of agents. This comment stresses the most important subfactors regarding agents.

Tool factor

The importance of using tools in software projects is well recognized by the software community, though definite conclusions on their impact are lacking. It is hoped that tools would contribute to increased productivity and quality of deliverables. It is convenient to assess the level of automation and the power of tools in the project. CASE tools and project control tools impact project efficiency. A reference framework classifies these tools into *tools* that support only specific tasks in the software process, *workbenches* that

support only one or a few activities, and *environments* that support (a large part of) the software process [F94].

To classify tools, several categories were suggested: basic development tools, advanced development tools, management tools, support tools and measurement tools. Table 3 shows the result of this ranking. Project control tools were important but were considered attached to the process factor. Other general usage tools such as editors or spreadsheets were considered trivial. Measurement tools were considered less important than development tools.

Table 3: Tools

Subfactors	M1	M2	M3	M4	M5	T
Basic Development Tools	1	1	1	1	1	1
Advanced Development Tools	1	3	1	2	1	2
Support Tools	2	2	2	1	2	3
Measurement Tools	3	4	3	3	3	4

The more important comments gathered by practitioners were as follows:

Tools to be considered in this factor should be related to software development or maintenance rather than project management. The tool factor is oriented to evaluate the technical assistance during development or maintenance activities. The impact of project management tools is determined within the process factor.

A tool maturity taxonomy should be suggested: basic toolkit, workbenches and software development environments. This classification follows a CASE tools taxonomy in the literature [F93].

2.2.2 System dimension

Product factor

During the software production cycle, different products are generated and referred. Characteristics of products, including deliverables and the final software, are assessed. Three subfactors were identified for product: size, quality and complexity. Table 4 pre-

sents the results of the ranking. Size could be interpreted in source lines of code, function points or other indirect measurement such as number of modules or routines and pages of documentation. Quality was suggested as conformance to standards rather than a subjective index. Complexity considered intra-module and inter-module complexity.

The ranking of product shows that quality and complexity were more important than size. This result indicates that a product can be managed independently of its size, provided its quality is high and its complexity is low.

Table 4: Product

Subfactors	M1	M2	M3	M4	M5	T
Size	2	3	2	2	2	3
Quality	1	1	1	3	1	1
Complexity	3	2	1	1	3	2

Some comments given by participants included:

The evaluation should be oriented towards tangibles delivered rather than temporary products generated during development or maintenance. The evaluation of this factor requires the analysis of tangibles supplied with the system, rather than work products during development. It is an assessment of deliverables of the project.

Functionality and complexity of the software are the most important subfactors to evaluate. The objective of the product evaluation factor is to assess the internal characteristics of the system, primarily concerning its size and complexity.

It is difficult to measure size and complexity (e.g., function points, number of screens). Unless automated measurement tools are available to quantify this factor, the number of implemented functions gives a good indication of size and complexity.

Use indirect measurements (e.g., development effort, number of people in charge of maintenance) instead of direct measurements on the product. When concrete measures are not available, it is necessary to depend on other attributes related to the product.

Performance factor

Dynamic aspects of performance of the system are considered within this factor. Performance was analyzed in terms of efficiency and reliability. Both were considered important for evaluation purposes. Table 5 presents the results of the ranking.

Table 5: Performance

Subfactors	M1	M2	M3	M4	M5	T
Efficiency	1	2	1	2	1	2
Reliability	1	1	2	1	1	1

Some comments given by participants included:

It is difficult to establish the context of reliability and efficiency. Only some functions within a system require reliability and efficiency concerns. The evaluation of this factor should identify these functions and generate a global assessment for the system.

Technology factor

Target software technology allows classification of systems for purposes of comparisons. Technology required the evaluation of software and hardware which implements the system. Language and operating system characteristics are associated to software. Single user or multi user microcomputers, minicomputers, mainframes and open systems are hardware related considerations.

Table 6 shows the results of the ranking. It was primarily the software: programming languages, database managers and operating systems that were considered more important than hardware. Hardware was secondary even though it was suggested that its characteristics should be known.

Table 6: Technology

Subfactors	M1	M2	M3	M4	M5	T
Hardware	1	2	1	2	2	2
Software	1	1	1	1	1	1

Comments given by participants regarding this factor included:

The level of sophistication of the technology is an important consideration: from traditional to advanced. The implementation technology indicates potential problems during the operation of the system. New technologies are prone to unreliable behavior, whereas old technologies used to be stable.

The complexity of the technology is difficult to establish. Moving from single users to open systems reflects an increasing level of complexity of the technology.

The level of technology innovation impacts development and maintenance. New technologies require time to stabilize.

Beware of giving specific example technologies because the description of categories may become old-fashioned. The descriptions of technology categories should avoid suggesting precise software or hardware technologies.

2.2.3 Environment dimension

Compliance factor

Compliance involves the overall evaluation of the system from the users perspective. An important activity is to verify that users requirements are met by the system. Compliance was considered in terms of conformance to requirements, information accuracy and user satisfaction. Table 7 presents the results of the ranking. Remarkably, user satisfaction was not considered by project managers as important as conformance to requirements or information accuracy. Conformance to requirements, considered the most important subfactor, showed the need to clarify whether it was conformance to specifications or conformance to system needs. The fuzziness of user needs was indicated by management.

Table 7: Compliance

Subfactors	M1	M2	M3	M4	M5	T
Conformance to requirements	1	2	1	1	1	1
Information accuracy	1	1	2	3	3	2
User satisfaction	2	2	3	2	2	3

Some comments given by participants included:

The difference between requirements and specifications has to be stressed. Requirements are users-oriented, whereas specifications are developer-oriented.

Project management considers that requirements are always accomplished, while users may have a different opinion. It is the users perception of accomplishment of their requirements that should be evaluated.

What is important is that the user be capable of doing his job. This is a judgment on the part of the user, it has to validate the accomplishment of the requirements.

Usability factor

Usability involves some degree of subjectivity because it is the user's evaluation of the system. Some researchers in this area have suggested its importance and possible objective quantification of its indicators [N92], [GBL91]. Usability was subdivided into learnability and interface characteristics. Table 8 shows the results of the ranking. Learnability included the time to learn the system and shortcuts available to expert users. The interface was evaluated in terms of user-oriented terminology, non redondancy and quality of error messages.

Table 8: Usability

Subfactors	M1	M2	M3	M4	M5	T
Interface characteristics	2	1	2	2	2	2
Learnability	1	2	1	1	1	1

Comments given by participants were as follows:

Considerations on learning the system are very important. To make a judgment about the usability of the system requires several considerations regarding redundancy, consistance and so on. To facilitate the assessment task, learning considerations are primarily evaluated.

Shortcuts for experienced users are also important. This is an important concern for usability. Once a system is mastered by users, their productivity becomes a concern.

Contribution factor

Contribution of a system to the organization is a very important indicator to evaluate a system, there are however serious difficulties as to how to quantify this factor because, besides economic benefits, there are intangible benefits difficult to quantify [B93], [GW91], [C91]. Table 9 presents the results of the ranking. We tried to identify those aspects that may provide intangible benefits like support to decision making, benefits to customers and support to organizational objectives apart from those that are more tangible in nature such as productivity improvements and tangible benefits to the organization. Even if all the aspects are important, tangible benefits are rated first.

Table 9: Contribution

Subfactors	M1	M2	M3	M4	M5	T
Support to user's decision making	1	3	1	5	3	5
Support to organizational objectives	1	3	1	3	4	3
Users productivity improvements	2	2	1	4	2	4
Benefits to the organisation	2	1	1	2	1	1
Benefits to customers	2	2	1	1	1	2

Some subfactors were considered embedded within other subfactors like for example support to organizational objectives and benefits to the organization. Benefits to the organization and the customers were rated first, indicating that the relationship organization-customers is fundamental to the subsistence of any organization.

Some comments regarding this factor were as follow:

Benefits to the organization are the most important. The orientation of this factor is towards identifying tangible benefits. However, other intangible benefits can be identified.

Support to high level management is also important. This would be a type of intangible benefits to evaluate.

For some systems, the customer of the organization is as important as the user of the system. The impact of the system towards its surroundings has to be evaluated.

When evaluating the contribution factor consider not only the user of the system, but also its customers.

Different systems have different impacts, how should this be considered? A system oriented to customers is more sensitive than one that is internal to an organization. The evaluation of the contribution of the system should take this aspect in consideration.

2.3 General comments from participants

Some final comments, given by participants at the end of the session, were as follows:

Gathering data during evaluations, at lower levels of detail, may provide a baseline for estimation purposes. Detailed data can be available for other purposes such as estimation.

The user organization is an important aspect to be characterized, it makes a difference on the evaluation of the system. Politics in the user organization may impact the success of a project. This aspect should be analyzed during the evaluation of the process.

The level of granularity of the evaluation may require more detail. Additional levels of detail can be introduced. Dynamically, it is possible to request this detail according to evaluation needs following the hierarchy of the framework.

Each factor has to be defined by an evaluation objective. Defining the objectives at the dimension and factor levels improves the context of evaluation.

Establish when the evaluation is applied: before development (i.e., estimation), after development (i.e., installation) or operation (i.e., post-evaluation). The evaluation can be applied at any point during the life cycle of a system. At earlier stages, the evaluation represent the desired characteristics of the system.

What is being evaluated, the system or the project. What happen in the case of a project that affects many systems. The approach is oriented to evaluate software systems. In the case of a project involving several systems, a separate evaluation for each system is recommended.

There are some terms that require clarifying (e.g., user, customer, organization).
Terminology concerns are being addressed by introducing a glossary of terms.

3.0 Profiles of systems

An evaluation of several information systems was carried out. The following table presents a summary of the results. The choice of systems was based on manager's recent familiarity with those systems. Project management was the main role performed by the participants. The next step was for the participants to assign the proper category for each factor.

The levels of maturity for each attribute in the framework are represented by three standard categories. Categories are useful for classifying software knowledge from a high-level perspective. To keep categories simple, only three ratings have been identified: basic (B), intermediate (I) and advanced (A). An intermediate category indicates a nominal rating or a standard in the industry, whereas basic and advanced ratings identify lower and higher ratings than the nominal one.

3.1 PROJECT

The project dimension seeks to characterize project efficiency considerations (i.e., ability to develop a system without waste of time, energy, etc.)

Process

The process factor seeks to evaluate the degree of efficiency and continuity of the process. It is primarily oriented to process management assessment.

Process categories

Basic: The project is characterized as one without a stable environment for producing software. Methodologies are adapted for each project and there is no follow-up of organizational learning from experience. Performance can only be predicted by individual, rather than project, capability.

Intermediate: The project follows a standard process for producing software. The software engineering and software management processes group facilitates software process defini-

tion and improvement efforts. Projects use the organization-wide, standard software process to create their own defined software process which encompasses the unique characteristics of the project. Each project uses a peer review process to enhance product quality.

Advanced: The project sets quantitative quality goals for software products. Productivity and quality are measured for important software process activities. Software processes have been instrumented with well-defined and consistent measures which establish the quantitative foundation for evaluating project processes and products.

Agent

The agent factor seeks to assess the capability of the team of people participating in the project. It should include in the assessment management and technical personnel.

Agent categories

Basic: The team is unprepared to undertake the project. There is some experience with similar applications, system design issues and programming techniques, but they are insufficient to build the required system. An extra effort is required from participants to improve their competence (e.g., attending training sessions, working overtime).

Intermediate: The team has the capability to undertake the project. Team members have varied levels of experience with related applications, system design and programming techniques. There is some level of confidence on their chances of success.

Advanced: The team has already demonstrated successfully its capacity to undertake similar projects. Team members have a consistent mix of experiences with related applications, system design and programming techniques. There is confidence of successful results.

Tool

The tool evaluation factor requires the establishment of the level of sophistication of tool support for development or maintenance activities. It is oriented to identify tools used by technicians rather than managers.

Tool categories

Basic: A basic programming toolkit is available for software production (i.e., compilers, debuggers and testers).

Intermediate: An improved programming toolkit is available for software production. Programming, verification and validation, configuration management, and measurement tools are available during software production. The use of workbenches in a limited basis may be available.

Advanced: Software development environments and CASE tools are available for software production which include software and process metrics. User interface development workbenches are used regularly.

3.2 SYSTEM

The system dimension is oriented to evaluate internal and external product characteristics and the technology implementing the system.

Product

The product factor requires determining an overall assessment of the internal product characteristics: size, quality and complexity.

Product categories

Basic: Software applications which implement few functions with low complexity and low quality requirements.

Intermediate: Software applications which implement several functions with increasing requirements on quality and complexity.

Advanced: Software applications that implement a large number of functions with high levels of complexity and increasing requirements on quality.

Performance

The performance factor concerns the assessment of software external characteristics: reliability and efficiency. It concerns dynamics rather than static characteristics.

Performance categories

Basic: The system is not bound by reliability or efficiency concerns. The effect of a software failure is an easily recoverable loss for users.

Intermediate: The system requires increased support on reliability or efficiency. The effect of a software failure is a situation from which users can recover without extreme penalty.

Advanced: The system has a major concern on reliability or efficiency requirements. The effect of a software failure can be a major financial loss or a massive human inconvenience.

Technology

The technology factor requires the establishment of the level of sophistication of the technology implementing the system.

Technology categories

Basic: Single-user and multi-user microcomputers and minicomputers.

Intermediate: Multi-user mainframe technology.

Advanced: Open-systems computer technology allowing distributed computing.

3.3 ENVIRONMENT

The environment domain evaluates the user's satisfaction with the system.

Compliance

The compliance factor assesses the accomplishment of user's requirements.

Compliance categories

Basic: Basic requirements are partially fulfilled by the system. There is no major impact on the users and their jobs. Improvements to the system are required in the short term.

Intermediate: Basic requirements are fulfilled by the system. There is an impact on the users and their jobs. Some areas of the system might be improved, but it is not considered to be urgent.

Advanced: Requirements are fulfilled by the system. The service provided by the system is appropriate, the information is adequate, current, and timely. There is a major positive impact on the users and their jobs.

Usability

The usability factor is directed to assess user interface characteristics. The impact on learnability is considered an important aspects to evaluate usability.

Usability categories

Basic: Learning the system may require a great deal of time. The system is not providing interfaces in user terms. Redundant information is required from the users.

Intermediate: The system requires some time to learn, but this is considered acceptable by the users. The system provides interfaces in user terms. There are some concerns with the system because error messages are not clear and there are no shortcuts for experienced users.

Advanced: The system requires a short time to learn. The system is user-oriented, efficient to use, prevents errors by the user and clearly signals the seriousness of any errors. Infrequent users have the facility to return to using the system without having to relearn it, and frequent users find shortcuts that improve their efficiency.

Contribution

The evaluation of the contribution of the system to the organization assesses the benefits provided by the system to the organization.

Contribution categories

Basic: There is no major contribution to the organization, the system has automated the same tasks as they were performed manually.

Intermediate: There are some intangible benefits to the organization. Decision-relevant information is processed in a cost-effective way, improving the quality and speed of management decision-making processes.

Advanced: The system provides tangible benefits to the organization. The organization is more cost-effective by using the system. Internal coordination costs have been reduced.

The results of the evaluation process are presented in the following table. Eight information systems were evaluated, corresponding to the eight columns of the table. The rows present the software systems attributes being evaluated.

Information Systems Profiles

	1	2	3	4	5	6	7	8
Process	I	I	I	I	I	I	I	I
Agents	A	I	A	I	I	I	A	A
Tools	I	B	I	I	I	B	B	I
Product	A	I	B	I	A	I	B	B
Performance	A	I	A	B	A	I	A	A
Technology	I	I	I	B	A	I	I	I
Compliance	I	I	I	I	A	I	I	I
Usability	I	A	A	I	A	B	I	I
Contribution	I	A	I	I	I	A	I	A

From these profiles it is possible to find out similarities among information systems. Analyzing the results by column, it is possible to identify systems possessing similar characteristics. Systems 2 and 6 are similar in all their factors except usability (advanced versus basic). Systems 1 and 3 are similar except for product and usability factors.

It also is possible to analyze, by row, each factor in the table to find out similarities. For the process factor, all the systems were considered intermediate. The systems were developed for different organizations by the same consulting firm using the same process modeling approach, which explains why they occupy the same process factor category. For the compliance factor, all the information systems were intermediate, except one

which was advanced (system 5). The rest of the factors were more varied for the rest of the systems.

Analyzing profiles by dimension, it is possible to identify systems sharing the same evaluation pattern, thus possessing similarities. At the project dimension level (process, agents and tools), systems 1, 3 and 8 share the same project evaluation (I, A, I); systems 2 and 6 share (I, I, B); systems 4 and 5 share (I, I, I). At the system dimension level (product, performance and technology), systems 3, 7 and 8 share the same evaluation (B,A,I), while systems 2 and 6 share (I, I, I). At the contribution dimension level (compliance, usability and contribution), systems 1, 4 and 7 share the same evaluation (I, I, I).

This analysis at a high level helps to identify similar systems sharing similar dimension evaluation. From this information, it is possible to determine the convenience to search for further details. For example, for estimation purposes, it would be convenient to select those systems that share similar project and system dimensions, get additional data for each factor and identify whether the new system being estimated shares the same characteristics of the existing systems.

3.4 Final comment regarding the evaluation

Some comments regarding categories and the purpose of the evaluation profile were as follows:

Categories should be clearly described, they should be precise to avoid wrong interpretations. Descriptions of categories must avoid being biased to predefined judgments. Some changes to the descriptions were made to improve understandability.

Context within which each category is evaluated. Missing comparison framework. Guidelines to apply the evaluation and examples of category choices are being documented.

The number of categories for each factor seems sufficient, it would not be convenient to increase this number. We try to keep the approach simple for easy usage.

How to use the evaluation profile. This profile gives an evaluation image of a system. It can be used to compare systems from a high level perspective.

Establish who should evaluate the system (e.g., technical, managerial or users and clients). A concensus process can be introduced to consider several points of view.

An advantage of having a baseline for evaluation is that it gives a uniform evaluation pattern for system comparison purposes. This is certainly an advantage of the evaluation approach.

Evaluating a system using a common baseline is going to improve the practice. Availability of evaluation approaches like the one being suggested here helps software system improvement.

4.0 Conclusions and directions for further research

We have gathered project management perceptions on software systems evaluations. A common software evaluation baseline which integrates in a hierarchy the important dimensions of software, its development environment and the organizational assessment of a system, was used to guide participants. Project managers provided their perceptions on important attributes to be evaluated, this perceptions are used as feedback improve the evaluation process. This experience indicates that it is fundamental to validate an evaluation approach in industry before applying it to a large number of systems.

Project management perceptions of software system attributes differ widely. This experience demonstrates that evaluation approaches have to be well documented before being applied in industry. Defining the importance of software systems attributes requires a concensus among practitioners. Different points of view, such of those of developers, operators and users, have to be integrated in a consistent software evaluation framework.

Project managers recognized the importance of the approach, considering the contribution to evaluating information systems a fundamental aspect for software system improvement. Management understanding is increased by performing information systems evaluations. Hot-spots attributes are identified in a top down fashion, requiring minimum evaluation effort on the part of project managers. Comparisons among information systems can be performed using a simple evaluation profile. Historical data on information systems evaluation can be used to estimate the characteristics of future systems.

Project managers have indicated that currently organizations are not performing software system evaluations. Some approaches to evaluate systems require weeks or month to produce a final result. Instead of hundreds of evaluation attributes, our approach only requires nine factors. By reducing the number of attributes being evaluated and concentrating at high-level attribute evaluation, it is possible to convince high-level managers of the importance of the evaluation stage.

Important software systems evaluation attributes have been identified using the hierarchical approach. The way the evaluation descriptions are presented has been improved after exchanging with project managers. Objectives for each software system evaluation attribute have been defined to clarify the basis for evaluation. The analysis of additional levels of depth in the hierarchy of attributes has been used as a mechanism to validate the choices at higher levels. Finally, additional software system attributes deemed important by practitioners have been incorporated into the scope of software system evaluation.

Some aspects not considered initially in the approach were identified, such as the importance of the user organization bureaucracy which affects project performance. Also, logically related attributes were attached together in the hierarchy. For example, tools related to project management should be associated to the process factor rather than the tools factor.

The time to produce an evaluation was an important consideration, about two hours were considered very convenient. There are so many aspects to consider in current evaluation approaches, that simpler approaches are required. Project managers found the approach much more flexible because additional information could be requested only in case of need. The hierarchical approach facilitates the selection of dimensions considered important by evaluators.

Work is in progress to perform information system evaluations in industry. Guidelines on the evaluation approach are developed to facilitate a self-assessment procedure. Comparison of evaluation results within a company should identify similarities among systems. Applying the evaluation approach to different companies should determine the feasibility of using a common evaluation approach across organizations.

Acknowledgement

The authors would like to thank the contribution of senior managers from a consulting company in Montreal.

References

- [B93] Brynjolfsson, E. 'The Productivity Paradox of Information Technology', Communications of ACM, Vol. 36, No. 12, December 1993, pp. 66-77.
- [C82] Chandler, J.S. 'A Multiple Criteria Approach for Evaluating Information Systems', MIS Quarterly, March 1982, pp. 61-74.
- [C93] Constantine, L.L. 'Work Organization: Paradigms for Project Management and Organization', Communications of the ACM, Vol. 36, No. 10, October 1993, pp. 34-43.
- [F93] Fuggetta, A. 'A Classification of CASE Technology', IEEE Computer, Vol. 26, No. 12, December 1993, pp. 25-38.
- [GBL91] Gould, J.D.; Boies, S.J.; Lewis, C. 'Making Usable, Useful, Productivity. Enhancing Computer Applications', Communications of the ACM, Vol. 34, No. 1, January 1991, pp.74-85.
- [GW91] Gurbaxani, V.; Whang, S. 'The impact of Information Systems on Organizations and Markets', Communications of the ACM, Vol. 34, No. 1, January 1991, pp.59-73.
- [K90] Kumar, K. 'Post Implementation Evaluation of Computer-Based Information System: Current Practices', Communications of the ACM, Vol. 33, No. 2, February 1990, pp. 203-212.
- [N93] Nielsen, J. 'Usability Engineering', Academic Press, Inc. 1993.

ÉCOLE POLYTECHNIQUE DE MONTRÉAL



3 9334 00289921 7